



中华人民共和国国家标准

GB/T 11457—XXXX
代替 GB/T 11457-2006

软件与系统工程 术语

Software and system engineering - Terminology

(ISO/IEC/IEEE FDIS 24765-, NEQ)

(征求意见稿)

(本草案完成时间：2026.6.16)

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX – XX – XX 发布

XXXX – XX – XX 实施

国家市场监督管理总局 发布
国家标准化管理委员会

目 次

前言 II

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

附录 A（资料性） 缩略语..... 592

参考文献 596

索引 598

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件代替GB/T 11457-2006《信息技术 软件工程术语》，与GB/T 11457-2006相比，主要技术变化如下：

——删除了已拒用术语条目 707 项；

——保留了术语条目 469 项；

注1：词条中用[来源：GB/T 11457-2006，2.1744]来表示。

——依据近 5 年发布的软件与系统工程领域国家标准，修改了原 GB/T 11457-2006 中术语条目 683 项；

注2：词条中在来源标注“有修改”进行表示。

——依据 ISO/IEC/IEEE 24765:2017 以及 IEEE、INCOSE 等国际组织中标准文件等术语来源，根据行业发展现状及需求，增加了术语 4128 项。

本文件参考ISO/IEC/IEEE FDIS 24765-《软件与系统工程 术语》起草，一致性程度为“非等效”。

本文件由全国信息技术标准化技术委员会（SAC/TC28）提出并归口。

本文件起草单位：中国电子技术标准化研究院、中国质量标准出版传媒有限公司、嘉兴市文诺财经大数据技术研究院、广西达译科技有限公司、北京软件产品质量检测检验中心有限公司、重庆市软件评测中心有限公司、北京工业大学、上海计算机软件技术开发中心、安世亚太科技股份有限公司、南京大学、中国航天系统科学与工程研究院、中国电子科技集团公司第二十八研究所、中国电子科技集团公司第十研究所、西南计算机有限责任公司、北京南天软件有限公司、中国航天科工集团第三研究院第三〇四研究所、中国科学院软件研究所、清华大学、北京大学重庆大数据研究院、普元信息技术股份有限公司、上海创景信息科技股份有限公司、成都飞航智云科技有限公司、中国核动力研究设计院、中船凌久高科（武汉）有限公司、同济大学、江苏省软件产品检测中心、深圳奥哲网络科技有限公司、浙江省电子信息产品检验研究院、中国软件评测中心（工业和信息化部软件与集成电路促进中心）、长江职业学院、上海市软件行业协会、国电南京自动化股份有限公司、深圳市金蝶天燕云计算股份有限公司、中国电子科技集团有限公司电子科学研究院、中国工程物理研究院计算机应用研究所、中国电子科技集团公司第五十四研究所、四川大学、北京理工大学、山东建筑大学、北京海鹰科技情报研究所、北京航空航天大学、上海同思廷软件技术有限公司、中兴通讯股份有限公司、中交第一公路勘察设计研究院有限公司、中国电力科学研究院有限公司、浪潮软件科技有限公司、北京赛昇认证服务有限公司、广东省科技基础条件平台中心、中邮信息科技（北京）有限公司、上海市软件评测中心有限公司、北京航天自动控制研究所、工业和信息化部电子第五研究所、电子科技大学、山东山科数字经济研究院有限公司、抚州中科院数据研究院、苏州市软件评测中心有限公司、中国电子科技集团公司第七研究所、北京软件和信息服务交易所有限公司、中国航空工业集团公司沈阳飞机设计研究所、山东正中信息技术股份有限公司、比亚迪汽车工业有限公司、金航数码科技有限责任公司、湖北光谷标准创新科技有限公司、北京凯思昊鹏软件工程技术有限公司、苏州洞察云信息技术有限公司、中国电子科技集团公司第十五研究所、厦门理工学院、山东省科技咨询中心有限公司、北京市科学技术研究院、湖南省佳策测评信息技术服务有限公司、浪潮通用软件有限公司、中海油能源发展装备技术有限公司、北京通和实益电信科学技术研究所有限公司、中国航空工业集团公司雷华电子技术研究所、上海微小卫星工程中心、上海至臻美信息科技

有限公司、太极计算机股份有限公司、四川科技职业学院、山东创恩信息科技股份有限公司、成都赢瑞科技有限公司、北京天圣华信息技术股份有限公司、南京华苏科技有限公司、广州职业技术大学、中国人民大学、贵州电网有限责任公司数智运营中心、江苏数睿数据科技股份有限公司、北京融工场科技有限公司、北京科信深度科技有限公司。

本文件主要起草人：张旻旻、苏伟、彭茜、王公韬、张文渊、邓姿娴、王威、丁晓明、刘潇健、胡芸、段海波、杨桂枝、张盟、周晓明、李宣东、李文鹏、农强、郝琳、叶玲、胡浩、李小霞、赵毅、邓日洁、郭栋、曾陈程、卜磊、毛亚萍、王林章、黄小蓉、李燕平、王伟、刘伟、戴敏、孟庆余、吴俊爽、王之栋、薛令德、张繁、刘张宇、林茂、杨隼、蔡临宁、曾辉、史雅馨、陆德波、杜博闻、赵浩强、吴真炜、邓燕君、季永炜、陶新昕、方鹏、肖筱华、朱辰泽、郭炳荣、胡戎、陶以政、赫畅、陈文、陈亚贤、刘新锋、职亮亮、李超、周震漪、王翔宇、康明、刘智、李哲、黄浩、吴玫青、邓昌义、余达明、张庆鑫、卢加乐、孟艳、郭晓慧、张玉宏、汤恩义、解宏泽、周兴、李梁、颜乐鸣、隋雅倩、艾笑天、于敏、陈纪旻、刘煜、陶福星、顾鑫、闫苗、张元元、张琦、卢俊文、许志国、李琼、李红霞、彭晓迪、高尚建、冯可欣、朱明慧、王国华、高猛、王春娥、于林平、范林东、王明亮、吕英胜、龙广宇、龙国胜、陈大龙、常超、李照川、纪元、向宇珂、张勇、罗鲜、朱文会、韦克炜、朴希望。

本文件及其所替代文件的历次版本发布情况为：

——1989 年首次发布为 GB/T 11457-1989，1995 年第一次修订；

——2006 年第二次修订为 GB/T 11457—2006；

本次为第三次修订。

软件与系统工程 术语

1 范围

本文件界定了软件与系统工程领域基础的术语和定义，并提供了相应的缩略语，见附录A。

本文件适用于软件与系统工程领域的概念理解、信息交流，也适用于软件开发、系统集成、信息技术服务相关标准的制定以及教育培训等活动。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

3.1

A/B 测试 **A/B testing**

拆分运行测试 **split-run testing**

用于测试人员判断两个系统或组件中性能更好的统计测试方法。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.1]

3.2

自主/智能系统创建者 **A/IS creator**

对自主/智能系统（A/IS）进行设计、开发、工程实现、编程或以类似方式进行创造的个人或实体。

[来源：IEEE 7010-2020, 2.1]

3.3

<个人>能力 **ability**

习得或自然的才能或者天赋，使得个人能够执行特定任务。

注：参见系统能力、企业能力、过程能力。

[来源：INCOSE SE Handbook, v5]

3.4

异/反常终止 **abnormal end**

过程在完成前终止。

[来源：ISO/IEC/IEEE FDIS 24765, 3.23]

3.5

异常中止 **abort**

在一过程完成之前被迫终止。

参见：异常终止，异常。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.6

缺失变体 **absent variant**

未在特定时间确定或开发的变体。

[来源：ISO/IEC 26554:2018, 3.1]

3.7

绝对地址 **absolute address**

显式地址 **explicit address**

特定地址 **specific address**

一种地址，它永久地赋予某一设备或存储单元，用以标识该设备或单元而不需要转换或计算。

参见：相对地址、可重定位地址、符号地址、绝对汇编程序、绝对代码、绝对指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.25]

3.8

绝对汇编程序 absolute assembler

产生绝对代码的汇编程序。

参见：重定位汇编程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.26]

3.9

绝对代码 absolute code

特定代码 **specific code**

一种代码，其所有的地址都是绝对地址。

参见：可重定位代码。

[来源：ISO/IEC/IEEE FDIS 24765, 3.27]

3.10

绝对指令 absolute instruction

一种计算机指令，其所有地址都是绝对地址。

参见：直接指令、有效指令、立即指令、间接指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.28]

3.11

绝对装入程序 absolute loader

一种装入程序，它将绝对机器代码读至主存储器，它从由汇编程序或编译程序赋予的初始地址开始。在执行中，代码中无地址调整。

参见：重定位装入程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.29]

3.12

绝对机器代码 absolute machine code

每当使用时，必须将其装入固定存储单元，且不能再定位的机器语言代码。

参见：可重定位机器代码。

[来源：GB/T 11457-2006, 2.14]

3.13

绝对时间 absolute time

在普遍认同的时间参考系中的时刻。

参见相对时间，自然单位。

示例：如协调世界时（UTC）这样给定参考时区的日历日期和具体时刻。

[来源：IEEE 982:2024, 3.1]

3.14

抽象行为 abstract behaviour

相似行为的描述、模型或其他普遍特征。

示例：用例，协议消息类型，序列图消息，接口控制文档。

[来源：IEEE 982:2024, 3.1]

3.15

抽象类 abstract class

不能独立实例化的类。

注：即，实例化是通过子类完成的。每个实例也是集群（总集群）中子类的实例的类被称为关于该集群的抽象类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.32]

3.16

抽象数据类型 abstract data type

只对数据的特性和在其上的操作作出规定而不管数据如何表示以及操作如何实现的一种数据类型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.33]

3.17

抽象设计 abstract design

需要专门化（进一步设计工作）以产生具体设计的通用形式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.34]

3.18

抽象机 abstract machine

过程或机器的一种表示。

[来源：GB/T 11457-2006, 2.16]

3.19

抽象 abstraction

关注与特定目的相关的信息而忽略其余信息的对象视图。

[来源：ISO/IEC 10746-2:2009, 6.3]

3.20

可接受性 acceptability

〈风险管理〉组织愿意承受的风险损失, 包括财务损失或其他形式的损失。

〈质量要求〉当接受或安装旨在执行某些常用功能的产品、系统或服务软件时, 人员响应所达到的满意程度。

注：风险可接受性可以适用于单个风险或风险集合, 例如项目或企业面临的全部风险。不同类别的风险的可接受性可能不同, 并取决于处理成本或其他因素。

[来源：ISO/IEC 25019:2023, 3.2.3]

3.21

可验收准则 acceptability criteria

项目工作产品的文件化特征集, 如果满足, 则构成判断每个产品内容可验收的充分基础, 以支持成功的评审或审计。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.1]

3.22

可验收的 acceptable

满足利益相关者的合理或值得的期望。

[来源：ISO/IEC/IEEE FDIS 24765, 3.39]

3.23

验收 acceptance

依据协议部分或全部履行的结果, 由需方或授权代表采取的行动, 通过该行动, 需方取得产品的所

有权的过程。

[来源：ISO/IEC/IEEE 24748-5-2017, 3.1]

3.24

验收准则 acceptance criteria

系统或组件需要满足才能被用户、客户或其他授权实体验收的标准。

[来源：ISO/IEC 33202:2024, 3.1]

3.25

验收测试驱动的开发 acceptance test-driven development; ATDD

一种开发方法，其中具有不同背景的团队（开发方、测试方和业务分析师）在开发相关功能之前共同编写验收测试。

注：测试驱动的开发侧重于先编写单元测试来驱动开发；ATDD 侧重于先编写验收测试来驱动开发。

[来源：ISO/IEC TR 7052:2023, 3.1]

3.26

验收测试 acceptance testing

确定一系统是否符合其验收准则，由用户、客户或其他授权实体确定是否接受系统或部件的正式测试。

参见：开发测试、运行测试、合格性测试。

3.27

验收的可交付成果 accepted deliverables

项目产出的，且被项目客户或发起人确认为满足既定验收标准的产品、结果或能力。

[来源：项目管理知识体系指南（PMBOK Guide）第五版]

3.28

已接受系统或系统元素 accepted system or system element

系统元素或系统已从供方转移到需方，而且产品或服务对项目可用。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.29

访问 access

获取资源的使用。

[来源：ISO/IEC 2382:2015, X.X]

3.30

访问设施 access facility

一组服务原语，允许桩对象协商抽象和传输语法，这些语法用于操作通过通道传输的数据。

[来源：ISO/IEC 14752:2000, 3.3.1]

3.31

访问方法 access method

用于获取数据使用权、存储读写权，或使用输入输出通道传输数据的技术。

[来源：ISO/IEC 2382-2015]

3.32

访问例程 access routine

一种访问隐藏数据结构的程序例程，这些结构常是全局变量或属于抽象数据类型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.48]

3.33

访问透明度 access transparency

为实现对象之间的互通，屏蔽了数据表示和调用机制差异的一种访问方式。

[来源：ISO/IEC 10746-3:2009，4.4.1.1]

3.34

访问（存取）控制机制 access-control mechanism

为使某一计算机系统或其某一部分允许被获准者和防止未获准者接触、访问而设计的硬件或软件的特性、操作过程或管理过程。

[来源：GB/T 11457-2006，2.21]

3.35

可达性 accessibility

〈通用〉产品、系统、服务、环境和设施能够被具有最广泛用户需求、特征和能力的人群使用，在确定的使用场景中达成既定目标的程度。

[来源：ISO TR 25060:2023，3.1.14]

[来源：ISO/IEC TS 25052-1:2022，3.1.3.1]

3.36

易访问性测试 accessibility testing

用于测量某一测试项可以被用户所使用程度的易用性测试的一种测试。

[来源：GB/T 38634.1-2020，3.1]

3.37

易访问性 accessiblity

在指定的使用周境中，为了达到指定的目标，产品或系统被具有最广泛的特征和能力的个体所使用的程度。

[来源：GB/T 25000.10-2016，4.3.2.4.6]

3.38

事故 accident

意外的事件或系列事件，导致死亡、受伤、疾病、环境破坏或者设备或财产的损坏或损失。

[来源：IEEE 1228-2002，3.1.1]

3.39

可问责性 accountability

产品具备唯一追踪实体行为并将其归因于该实体的能力。

[来源：ISO/IEC 25010:2023，3.6.4]

3.40

准确[性] accuracy

信息质量，即信息正确且与产品一致。

[来源：ISO/IEC/IEEE 26514:2022，3.1.2]

3.41

测量精度 accuracy of measurement

测量结果与被测量真值之间的一致性程度。

注：准确性是一个定性概念。精确度并非“准确性”的同义词。真值是指与给定特定量的定义一致的值，即通过完美测量获得的值。在完美测量不切实际的情况下，约定真值是指赋予特定量的值，有时根据惯例，认为该值具有适合给定目的的不确定度。“约定真值”在同一参考资料中有时也称为赋值值、最佳估计值、约定值或参考

值。准确性可以用相对误差的平均量级来表示。

[来源：ISO/IEC TR 14143-3:2003，3.1]

3.42

确认的系统之系统 acknowledged system of systems; acknowledged SoS

具有明确目标、指定管理者和资源的系统之系统（SoS）。

注：成员系统保留其独立的所有权、目标、资金及开发和维持方法。系统中的更改基于 SoS 和系统之间的合作协议。

[来源：ISO/IEC/IEEE 21841:2019，3.2.1]

3.43

获取资源 acquire resources

获取项目所需的团队成员、设施、设备、材料、用品和其他资源的过程。

[来源：项目管理知识体系指南（PMBOK(R) 指南）——第六版]

3.44

购得系统 acquired system

从供方交付给需方的系统或系统元素（产品或服务），符合采购/采办协议的交付条件。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.45

需方 acquirer

从供方获得或采购某一产品或服务的利益相关方。

注：需方的同义术语，常用的有买方、顾客、所有者、购买者或内部/组织赞助方。

[来源：GB/T 45802-2025, X. X]

3.46

获取 acquisition

获得某一系统、产品或服务的过程。

[来源：GB/T 22032-2021, X. X]

3.47

获取协定 acquisition agreement

项目组织和供应商间关于关系与承诺的谅解协议。该协议可以从正式合同到不太正式的组织间工作订单。正式协议通常包括条款和条件。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.48

获取后勤 acquisition logistics

为在获取过程早期及全过程中充分考量保障性影响，所开展的技术与管理活动，旨在降低保障成本，并为用户提供在现场持续支撑系统运行所需的资源。

[来源：INCOSE 系统工程手册，第五版]

3.49

采购/采办需要 acquisition need

识别遇到需求的组织内无法满足的需求，或供应商可以以更经济的方式满足的需求。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.50

采购/采办付款 acquisition payment

对所购系统的付款或其他补偿。包括汇款和确认。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.51

采购/采办记录 acquisition record

与采购/采办相关的永久、可读形式的数据、信息或知识。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.52

采购/采办回复 acquisition reply

一个或多个候选供应商对供应请求的响应。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.53

采购/采办报告 acquisition report

为利益相关方准备的账户，以传达收购活动的状态、结果和结论。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.54

获取策略 acquisition strategy

基于供应源、采办方法、需求规范类型、合同或协议类型以及相关采办风险的考虑，获取产品和服务的特定方法。

[来源：ISO/IEC/IEEE FDIS 24765，3.61]

3.55

动作 action

措施

用户在执行过程中所完成步骤的组成单元。

[来源：ISO/IEC/IEEE 26514:2022，3.1.3]

3.56

动作条目 action entry

标识某一动作与特定规则的关联关系。

[来源：ISO 5806:1984，3.9]

3.57

措施项 action item

一项已被接受的措施建议。

[来源：GB/T 11457-2006，2.26，有修改]

3.58

关注动作 action of interest

事务中引发关键状态变更的动作。

[来源：ISO/IEC 10746-3:2009，13.7.1.2]

3.59

措施建议 action proposal

更改过程或过程有关项的文档化建议，它将预防未来再出现已被缺陷预防活动标识出的缺陷。

参见：软件过程改进建议。

[来源：GB/T 11457-2006，2.27]

3.60

动作序列 action sequence

决定动作的序列的一种表示式。

[来源: GB/T 11457-2006, 2.28]

3.61

动作签名 **action signature**

对动作的规范说明, 包括动作的名称, 参数的数量、名称和类型, 以及通过动作模板所实例化的对象与该动作间的因果关系指示。

[来源: ISO/IEC 10746-2:2009, 9.14]

3.62

动作状态 **action state**

表示执行某一原子动作 (有代表性的是启用某一操作) 的一种状态。

[来源: GB/T 11457-2006, 2.29]

3.63

动作清单 **action stub**

为解决问题而需执行的所有动作的清单。

[来源: ISO 5806:1984, 3.11]

3.64

激活 **activation**

函数执行的一次具体实例, 将其输入的某个子集转换为其输出的某个子集。

[来源: ISO/IEC/IEEE FDIS 24765, 3.67]

3.65

激活约束 **activation constraint**

函数的一项要求, 作为函数某次激活的先决条件, 特定箭头角色中必须存在非空对象集。

[来源: ISO/IEC/IEEE FDIS 24765, 3.68]

3.66

激活函数 **activation function**

<神经网络> 与神经网络中节点相关联的公式, 用于根据神经元的输入确定节点的输出 (即激活值)。

同义词: 传递函数。

[来源: ISO/IEC TR 29119-11:2020, 3.1.3]

3.67

激活值 **activation value**

神经网络中某一节点的激活函数的输出。

[来源: ISO/IEC TR 29119-11:2020, 3.1.4]

3.68

活跃区域 **active area**

响应用户输入的区域。

示例: 图形上的热点、文本中的超链接、屏幕显示中的按钮。

[来源: ISO/IEC/IEEE 26514:2022, 3.1.4]

3.69

活动类 **active class**

一种其实例是活动对象的类。

参见: 活动对象。

[来源: GB/T 11457-2006, 2.31]

3.70

主动企业对象 active enterprise object

能够充当活动角色的企业对象。

[来源: ISO/IEC 15414:2015, 6.3.1]

3.71

活动文件 active file

尚未超过终止时间的文件。

[来源: GB/T 11457-2006, 2.32]

3.72

主动互联 active interconnection

一种物理交互机制, 允许一个事物的动作引发另一个事物的状态变化或触发其动作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.73]

3.73

活动对象 active object

一种拥有某一线程并能初启控制活动的对象。活动类的一个实例。

参见: 活动类、线程。

[来源: GB/T 11457-2006, 2.33]

3.74

活动冗余 active redundancy

在容错系统中, 采用同时运行的冗余组件来预防故障或支持从故障中恢复的技术方案。

参见: 待机冗余。

[来源: ISO/IEC/IEEE FDIS 24765, 3.73]

3.75

活动文本 active text

屏幕上显示的、响应用户输入的文本。

[来源: ISO/IEC/IEEE FDIS 24765, 3.75]

3.76

活跃空白空间 active white space

文本或图形元素周围的区域, 不包括边距, 用于分隔文本、分隔主题和子主题分组、指示层级和主题关系、突出显示信息或使文本更易于阅读。

[来源: ISO/IEC/IEEE FDIS 24765, 3.76]

3.77

活动 activity

某一过程中结合在一起的各项任务的集合。

[来源: GB/T 22032-2021, 4.1.3]

3.78

活动编码 activity code

由一位或多位数字或字符组成, 用来识别工作的特征, 或者用某种方式对进度活动进行分类, 以便在报告中对活动进行筛选和排序。

[来源: 项目管理知识体系指南 (PMBOK®指南), 第五版]

3.79

活动成本估算 activity cost estimates

进度活动的预计成本，包括执行和完成活动所需的全部资源的成本，包括全部的成本类型和成本元素。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.80

活动图 activity diagrams

通过受控的动作序列展现输入到输出的转换过程。

[来源：INCOSE 系统工程手册 第四版，10.8.2]

3.81

活动持续时间估算 activity duration estimate

对活动持续时间的可能量或结果的定量评估。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.82

活动组 activity group

一组相关活动。

[来源：ISO/IEC/IEEE FDIS 24765，3.73]

3.83

活动标识 activity identifier

为了使项目活动彼此区别，而分配给每项进度活动的简短且唯一的数字或字符标识。在任何一个项目进度网络图中，活动标识通常是唯一的。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.84

活动清单 activity list

计划活动的记录列表，显示活动描述、活动标识符和足够详细的工作描述的范围，以便项目团队成员理解将要执行的工作。

[来源：项目管理知识体系指南（PMBOK®指南）-第七版]

3.85

活动类型 activity type

由执行相同算法所定义的一类活动分类。

[来源：ISO/IEC 14756:1999，4.2]

3.86

作业成本核算 activity-based costing; ABC

按特定生产活动而不是从单一间接费用池中分配间接费用的成本核算方法。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.87

面向活动的 WBS activity-oriented WBS

活动导向型 WBS

以体现待完成工作的动词来标识各项活动与任务的工作分解结构。

注：每个任务名称均包含该任务所要产出的一个或多个工作产品。

[来源：ISO/IEC/IEEE FDIS 24765，3.82]

3.88

参与者 actor

与某活动相关的对象在活动中所扮演的角色。

[来源：ISO/IEC 15414:2015, 6.3.2]

3.89

实际成本 **actual cost, AC**

已完成工作的实际成本 **actual cost of work performed ACWP**

在特定时间段内、就某一活动已完成工作发生的实现成本。

参见：挣值管理，挣值技术。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3.90

实际折旧 **actual depreciation**

资产价值的真实损失，只有在资产被出售时才能确定。

[来源：ISO/IEC/IEEE FDIS 24765，3.85]

3.91

实际持续时间 **actual duration**

以日历单位计算，指进度活动的实际开始日期与给定的规划数据日期（如果该进度活动尚未完成）或实际完成日期（如果该进度活动已经完成）之间的时间长度。

[来源：项目管理指南知识体系（PMBOK®指南），第五版]

3.92

实参 **actual parameter**

在调用子程序时，用来指定数据或要传输给该子程序的程序元素的数值或表达式。

见：序号 263 自变量（argument）。与序号 2133 形参（formal parameter）条相对。

[来源：GB/T 11457-2006 2.39]

3.93

实测结果 **actual results**

作为测试执行结果可获取的测试项的行为、状态集，或相关数据、测试环境的状态集。

示例：向硬件的输出，对数据的更改，报表的生成和展现，通信消息的发送。

[来源：GB/T 38634.1-2020, 3.2]

3.94

临时审查 **ad hoc reviewing**

一种非结构化的独立审查技术。

[来源：ISO/IEC 20246:2017, 3.1]

3.95

适应性 **adaptability**

产品有效且高效地适应或转移到不同硬件、软件或其他操作或使用环境的能力。

[来源：ISO/IEC 25010:2023, 3.8.1]

3.96

配置数据 **adaptation data**

用于使程序适应特定安装现场或其运行环境中特定条件的数据。

[来源：ISO/IEC/IEEE FDIS 24765，3.91]

3.97

配置参数 **adaptation parameter**

被赋予特定值的变量，用于使程序适应给定的安装地点，或适应其运行环境中给定的条件。

示例：变量 `Installation_Site_Latitude`（安装地点纬度）。

[来源：ISO/IEC/IEEE FDIS 24765，3.92]

3.98

适配器 adapter

使得非兼容系统相互接口的软件或设备。

示例：转换电流的设备充电器，将应用连接到不同数据库的对象适配器。

[来源：ISO/IEC/IEEE FDIS 24765，X.X]

3.99

适应性方法 adaptive approach

项目需求具有高度不确定性和易变性，并可能在项目全生存周期中发生变更的（项目）开发方法。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3.100

适应型生存周期 adaptive life cycle

迭代型或增量型项目生存周期。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3.101

适应性维护 adaptive maintenance

对交付后的软件产品以适应已变更或正在变更的环境而进行的修改。

注：适应性维护提供了必要的改进，以适应软件产品必须运行于其中的环境的变更。为了与不断变更的环境保持同步应做出相应的变更。例如，操作系统可能升级并且可以做某些变更，以适应新的运行系统。

[来源：GB/T 20157-2006，4.1]

3.102

适应性策略 adaptive strategy

残障人士用于改善与互联网交互的技术。

示例：在通用浏览器中增加字体大小。

[来源：ISO/IEC 29110-5-1-2:2025，3.4]

3.103

适应性支持 adaptive support

对所关注在役系统进行的安装后修改，以使其在已改变或改变中的运行时环境中可使用。

参见：增量附加性支持、完善性支持。

[来源：IEEE 982:2024 3.1]

3.104

新增源代码语句 added source statements

为特定软件产品新创建的源代码语句的数量。

[来源：ISO/IEC/IEEE FDIS 24765，3.98]

3.105

其他质量规划工具 additional quality planning tools

用来定义质量要求、规划有效的质量管理活动的一组工具。

注：包括但不限于头脑风暴、力场分析、名义小组技术和质量管理控制工具。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.106

增项维护 additive maintenance

在软件产品交付后进行的修改，用于添加功能或特性以增强产品的使用。参见：序号 3576，完善性

维护 (Perfective Maintenance)。

注 1: 在关于可靠性/可用性等语境中, 如果“维护”的定义仅指将系统恢复到先前的运行、功能或性能水平(例如可用性、可恢复性或 MTBF(平均故障间隔时间)的定义、监控或度量), 则增项维护可以被排除在维护定义之外。

注 2: 增项维护类型与“完善性维护”类型相区分, 并被视为一种独立的维护类型, 用于: 提供额外的新功能或特性, 以改进软件在未来的可用性、性能、可维护性或其他软件属性。

注 3: 在软件交付之后, 通过相对大规模的新增或变更来改进软件属性, 并在此过程中为供方与需方就这些新增或变更在维护策略、方法、资源、协议或服务级别等方面进行协商创造机会。这类新增或增强可以通过维护过程来处理, 而更大范围的变更则可能需要启动新的开发工作。

[来源: ISO/IEC/IEEE 14764:2021, 3.1.2]

3.107

附加性支持 additive support

对在役系统进行的安装后修改, 以向系统实例添加功能或特性。

参见: 完善性支持, 适应性支持。

[来源: IEEE 982:2024, 3.1]

3.108

加法加权法 additive weighting

通过为选定的决策属性分配不同的数值(权重), 以提高这些属性在决策中的重要性的方法。

[来源: ISO/IEC/IEEE FDIS 24765, 3.101]

3.109

地址(名词) address

寻址(动词)

处理(动词)

<名词>用于标识特定设备或存储位置的数字、字符或字符组。

<动词>通过标识号、字符或字符组来引用某个设备或存储位置。

<动词>(特指)决定是否以及何时纳入某项已定义的文档主题(直接纳入或引用其他文档); 决定某一项是在测试执行前记录(无论是否使用工具)、测试执行期间记录、测试执行后记录、通过流程处理(不单独记录), 还是予以排除。

[来源: ISO/IEC/IEEE FDIS 24765, 3.102]

3.110

地址字段 address field

地址部分 address part

计算机指令的字段, 此字段包含地址、导出地址必需的信息或操作数值, 与序号 3380 操作字段相对。

[来源: ISO/IEC/IEEE FDIS 24765, 3.103]

3.111

地址格式 address format

在计算机指令中地址字段数和排列。

参见: n 地址指令, n+1 地址指令。

[来源: ISO/IEC/IEEE FDIS 24765, 3.104]

3.112

地址修改 address modification

在地址上执行的任何算术的、逻辑的或句法的操作。

参见：有效地址，索引地址，相对地址，可重定位地址。

[来源：ISO/IEC/IEEE FDIS 24765，3.105]

3.113

地址空间 address space

中央处理单元能寻址的存储单元数。

注：在某些系统中，这可能是与其他程序分开的程序能访问的物理地址集和引用可由其他程序访问的存储单元的虚拟地址集。

[来源：ISO/IEC/IEEE FDIS 24765，3.106]

3.114

寻址异常 addressing exception

当程序计算的地址超出其可用存储空间边界时发生的异常。

参见：数据异常，操作异常，溢出异常，保护异常，下溢异常。

[来源：ISO/IEC/IEEE FDIS 24765，3.107]

3.115

寻址方式 addressing mode

在中央处理器指令集架构中，用于查找操作数位置的方法。

[来源：ISO/IEC/IEEE FDIS 24765，3.108]

3.116

寻址范围 addressing range

由计算机指令系统所指定并使用的地址空间。

注：寻址范围取决于地址线的位数以及所采用的寻址方式。

[来源：ISO/IEC/IEEE FDIS 24765，3.109]

3.117

调整规模 adjusted size

功能规模乘以技术复杂度调整之后的测量值。

注：此测量值不代表实际的功能规模。

[来源：GB/T 42566-2023]

3.118

调整提前量与滞后量 adjusting leads and lags

项目执行过程中使用的，让进度落后的项目活动追赶上计划的一种技术。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]

3.119

采用过程 adoption process

组织广泛推行使用 CASE 工具的系列活动。

[来源：ISO/IEC TR 14471:2007, 2.1.2]

3.120

高级配置文件 advanced profile

针对极小实体(VSE)的配置文件，这些 VSE 希望作为有竞争力的系统或软件开发组织独立维持和发展。

[来源：ISO/IEC 29110-1-2:2024, 3.3]

3.121

对抗性攻击 **adversarial attack**

刻意实施对抗性实例使机器学习模型失效。

注：通常针对神经网络形式的机器学习模型。

[来源：ISO/IEC TR 29119-11:2020, 3.1.6]

3.122

对抗性实例 **adversarial example**

对正常样本施加微小扰动后生成的、输入给机器学习模型的样本，会使模型以高置信度输出错误结果。

注：通常适用于神经网络形式的机器学习模型。

[来源：ISO/IEC TR 29119-11:2020, 3.1.7]

3.123

对抗性测试 **adversarial testing**

基于尝试创建和执行对抗性样本来识别机器学习模型中缺陷的测试方法。

注：通常适用于神经网络形式的机器学习模型。

[来源：ISO/IEC TR 29119-11:2020, 3.1.8]

3.124

不利后果 **adverse consequence**

导致指定级别的后果损失。

注：利益系统处于危险状态，加上系统环境处于最坏情况（相对于不利后果），会导致不利后果。不利后果的概念不仅包括安全方面的伤害，还包括其他损失，例如安全方面的财产损失。

[来源：ISO/IEC 15026-3:2015, 3.1]

3.125

影响 **affect**

改变使用方或其他利益相关方决定获取或使用一个系统的态度。

[来源：ISO/IEC 25019:2023, 3.1.1]

3.126

情感计算 **affective computing**

研究和开发能够识别、解读、处理和模拟人类情感的系统和设备。

[来源：IEEE 7014-2024, 3.1]

3.127

情感关系 **affective relationship**

行为模式中先前输入交互事件和后来输出交互事件之间的功能依赖性。

[来源：IEEE 1175.4-2008, 3.2]

3.128

情感权利 **affective rights**

赋予全人类的规范性权利，这些权利直接关系到交流、思想、信念和情感以及通常就其发展与表达的隐私。

[来源：IEEE 7014-2024, 3.1]

3.129

传入的 **afferent**

在软件系统中，从子模块至超级模块的数据或控制流。

参见：传出 efferent。

[来源：ISO/IEC FDIS 24765, 3.122]

3.130

亲和图 affinity diagram

群体创新技术的一种，用来对大量创意进行分组，以便进一步审查和分析。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.131

相似性分组 affinity grouping

根据事物的相似性将它们归类到相似类别或集合中的过程。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.132

能动性 agency

受系统影响的用户、数据主体和其他利益相关者，其相对系统拥有自主性、自由度、访问和决策能力的性质和程度。

注：能动性也可以指代特定系统影响个体或群体的用户、数据主体或其他利益相关者的能力。

[来源：IEEE 7014-2024, 3.1]

3.133

代理 agent

被一方委托某事（授权、责任、提供服务等）并代表该方行事（行使授权、履行责任、提供服务等）的活动企业对象。

注：代理可以是参与方，也可以是开放分布式处理（ODP）系统或其组件之一。ODP系统环境中的另一个系统也可以是代理。委托可以由一方直接进行，也可以由经该方授权的该方代理人间接进行。

[来源：ISO/IEC 15414:2015, 6.6.8]

3.134

聚集 aggregate

一个或多个系统元素，按照物理的、逻辑的或者二者均含组合的结果。

[来源：ISO/IEC/IEEE 24748-6:2023, 3.1.1]

3.135

聚集[类]型 aggregate [class]

在聚集（整体）与某一构件部分之间，规定整体一部分联系的关联一种特别形式。

[来源：GB/T 11457—2006, 2.52]

3.136

总体责任 aggregate responsibility

最终细化为特定属性和约束的广泛陈述的责任。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.3]

3.137

聚合资源利用率 aggregated resource utilization

服务有效利用资源池中聚合资源以支持多租户的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.1.2]

3.138

聚合 aggregation

两个元素之间的派生关系，这些元素是其他元素的组，表示两个组的分组元素之间的所有个体关系。

[来源：ISO/IEC/IEEE 247651:2024]

3.139

聚集方法 aggregation method

组合一组测量值以创建复合值的方法。

注：聚集方法基于补偿或非补偿模型。

[来源：ISO/IEC 33003:2015, 3.1]

3.140

敏捷 agile

通过实现快速响应反馈，对产品与服务进行开发、交付和维护的方法。

注：只有在工作开始后，才用具体特定的需求详细阐述软件的开发规约。这一精益原则旨在避免工作浪费，并为敏捷团队提供确定工作优先的方法。

[来源：ISO/IEC 33202:2024, 3.2]

3.141

敏捷概念 agile concept

敏捷的基本构建模块。

[来源：ISO/IEC 33202:2024, 3.3]

3.142

敏捷开发 agile development

基于迭代开发、频繁检查和调整以及增量交付的开发方法，其中需求和解决方案通过跨职能团队协作和利益相关方持续反馈演进。

[来源：ISO/IEC/IEEE 26515: 2018, 3.1]

3.143

敏捷环境 agile environment

实施敏捷开发方法和方法的组织或团队。

[来源：ISO/IEC/IEEE 26515:2018, 4.2]

3.144

敏捷成熟度 agile maturity

组织、部门、项目或团队在多大程度上始终如一地应用有助于实现其业务需求的敏捷价值观和原则。

[来源：ISO/IEC TR 24587:2021, 3.2]

3.145

敏捷团队 agile team

采用敏捷方法论，协作开发产品的小型跨职能团队。

注：常见的敏捷团队规模为3至10人。团队通常包括团队负责人、项目经理、用户或用户代表、软件和信息开发人员以及测试人员等角色。

[来源：ISO/IEC TR 24587:2021, 3.3]

3.146

敏捷团队领导 agile team lead

负责确保敏捷团队遵守组织的敏捷原则、实践、价值观和流程的个人。

注：敏捷团队领导是促进者而不是管理者。

[来源：ISO/IEC TR 24587:2021, 3.4]

3.147

协议 agreement

相互认可的条款和条件，并据此使工作关系得以实施。

[来源：ISO/IEC/IEEE 12207:2026, 3.1.5]

3.148

代数语言 algebraic language

一种编程语言，它允许语句的结构重新装配代数表达式。

参见：算法语言。

[来源：GB/T 11457—2006, 2.53, 有修改]

3.149

算法 algorithm

用有限步数求解某问题的一套明确定义的规则的集合：例如，求 $\sin(x)$ 到给定精度的一系列算术运算的完整规格说明。

为执行特定任务的任何运算序列。

[来源：GB/T 11457—2006, 2.54]

3.150

算法分析 algorithm analysis

对一算法的检查，其目的在于确定与其预期用途有关的正确性，确定其运行特性，或为了更充分地理解某一算法以便对其进行修改、简化或改进。

[来源：GB/T 11457, 2.55]

3.151

算法漂移 algorithmic drift

在学习或技术改进过程中，由于系统对相同数据的处理方式发生变化而产生的漂移。

注：该现象会导致系统在输入漂移前的相同数据时，给出不同结果。

[来源：IEEE 7014-2024, 3.1]

3.152

算法语言 algorithmic language

为处理算法而设计的编程语言。例如，ALGOL。

参见：序号 180 代数语言 algebraic language,。

[来源：GB/T 11457, 2.56]

3.153

别名 alias

一种结构的备用名称，比如类、责任、实体或域。

[来源：ISO/IEC/IEEE FDIS 24765, 3.148]

3.154

分配的基线 allocated baseline

经批准的产品、子系统或组件需求，描述从高层级需求分配下来的功能、性能、互操作性和接口需求，以及为证明这些需求已实现所需开展的验证工作，在特定时间点确立并记载于分配构型文档中。

参见：开发配置、功能基线、产品基线、分配配置标识。

[来源：ISO/IEC/IEEE 24748-7:2019, 3.1]

3.155

分配的配置标识 allocated configuration identification

在配置管理中，指针对隶属于某一更高层次配置项的各配置项所制定的、当前已经批准的规范集合，用以指导这些配置项的开发活动。

参见：功能配置标识（functional configuration identification）、产品配置标识（product configuration identification）、分配基线（allocated baseline）。

注：每项规范均定义从上层配置项分配而来的功能特性，规定用于证明其已达成所分配功能特性所需的测试，界定与其他相关配置项之间必需的接口需求，并在存在设计约束时对其进行规定。

[来源：ISO/IEC FDIS 24765, 3.151]

3.156

分配需求 allocated requirement

来自高层级需求, 并被指派至低层级架构元素或设计组件的全部或部分功能、性能要求。

[来源：ISO/IEC FDIS 24765, 3.152]

3.157

分配 allocation

在系统或程序的部件间分发需求、资源或其他实体的过程。

注：分配可以是完全针对硬件、软件或人类，或某种组合，以在进一步功能分解时解决。

[来源：ISO/IEC FDIS 24765, 3.153]

3.158

权利分配 allocation of an entitlement

将给定权利的部分或全部分配给管理自身权利模式库的子公司或其他相关组织单位的过程。

注：权利模式能够记录权利分配。

[来源：ISO/IEC 19770-3:2016, 3.1.1]

3.159

Alpha 测试 alpha testing

产品在可投入商业使用或正式运行之前的第一阶段测试。

注：通常仅由开发软件的组织内部用户执行。

[来源：ISO/IEC/IEEE FDIS 24765, 3.156]

3.160

字母数字混合 alphanumeric

由字母、数字和通常其他字符（如标点符号）组成的数据，以及使用这些数据的过程和功能单元。

3.161 [[来源：ISO/IEC 2382:2015, X.X]

备选流 alternate flow alternate path

描述其备选实现的用例的一部分。

注：它也用于描述错误条件，因为错误可以被认为是一种备选方案。

[来源：ISO/IEC FDIS 24765, 3.159]

3.162

备用键 alternate key

实体中，除主键之外的其他候选键。

注：键样式。

[来源：IEEE 1320.2-1998, 3.1.5]

3.163

备选方案分析 alternative analysis; 24765 中是 “ alternatives analysis”

一种对已识别的可选方案进行评估的技术，用来决定选择哪种方案或使用何种方法来执行项目工作。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.164

备选解决方案类 Alternative solution classes

识别并描述可能（有助于）处理问题或机会的解决方案类。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.165

备选方案生成 alternatives generation

一种用来制定尽可能多的可选方案的技术，目的在于识别执行项目工作的不同方法。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]。

3.166

模糊性 ambiguity

不明确、确定事件原因困难或多选项并存的状态。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3.167

模拟 analog

将项目的部分属性与高度类似的一个或几个已完成项目的数据进行比对，适当调整后获得待估算项目工作量或成本估算值的方法。

[来源：GB/T 36964-2018, 1.72]

3.168

类比估算 analogous estimating

使用相似活动或项目的历史数据，来估算当前活动或项目的持续时间或承办的技术。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.169

模拟计算机 analogue computer

操作模拟另一个系统的行为，并能接收、处理和生成模拟数据的计算机。

示例：算盘。

3.170 [[来源：ISO/IEC 2382:2015, X.X]

类推法 analogy

将本项目的部分属性与高度类似的一个或几个已完成项目的数据进行比对，适当调整后获得待估算项目工作量或成本估算值的方法。

[来源：GB/T 36964-2018 3.10]

3.171

可分析性 analysability

产品可被有效和高效评估的能力，以考察对产品自身一个或多个部件进行预期变更的影响，诊断缺陷或故障原因，或确定需要修改的部件。

参见：可修改性。

注：实施可包括为产品或系统提供机制，以解析其自身故障，并在故障或其他事件发生前后提供报告。

[来源：ISO/IEC 25010:2023, 3.7.3]

3.172

分析 analysis

开发的调查和收集阶段，旨在明确用户类型及其信息需求。

参见：系统分析。

示例：测试。

[来源: ISO/IEC/IEEE 26512:2018, 4.2]

3.173

分析模型 analysis model

组合了一个或多个基本测度和（或）派生测度及其相关判定准则的算法或计算。

[来源: GB/T 25000.40-2018, 4.2]

3.174

分析阶段 analysis phase

软件生存周期中的一个阶段，在此期间将需求阶段所定义的产品需求转化为逻辑上的系统解决方案，定义系统内部的结构、数据流和处理逻辑等，并编制出相应的文档。

3.175

分析情境 Analysis situations

分析的上下文信息。

注 1：包括生存周期阶段、评价驱动因素、成本驱动因素、规模驱动因素、团队特征、项目优先级以及其他表征信息和参数，这些表征信息和参数对于理解分析和表示被分析要素是必要的；还包括来自调用分析的过程的相关信息，与被分析要素相关的任何现有模型与数据，这些数据包括历史数据、当前数据和预测数据。分析的上下文信息可以源自任意生存周期过程。

注 2：来自调用分析的过程的相关信息。与正在分析的元素相关的任何现有模型。与正在分析的元素相关的任何数据，包括历史数据、当前数据和预计数据。可以源自任何生存周期过程。

[来源: INCOSE 系统工程手册 第四版，附录 E]

3.176

分析师 analyst

具备相关技能并经过培训，能够定义问题，分析、开发和表达算法的技术团队中的成员。

示例：系统工程师、业务分析师。

[来源: ISO/IEC/IEEE FDIS 24765, 3.167]

3.177

层次分析法 analytic hierarchy process,; AHP; AHP

在决策中使用矩阵来管理成对关系。

参见：加权法、无量纲缩放、补偿决策技术。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.178

分析技术 analytical techniques

根据可能的项目或环境变量变化及它们与其他变量之间的关系，对潜在后果进行评估、分享和预测的各种技术。

[来源: 项目管理知识体系指南（PMBOK®指南）——第五版]

3.179

祖先类(类的) ancestor (of a class)

类的通用祖先、类的父类或类的父类的祖先。

参见：通用祖先、自反祖先。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.6]

3.180

祖先框 ancestral box

通过一个或多个父/子关系的层次连续序列与特定图表相关的框。

[来源：IEEE 1320.1-1998 (R2004) IEEE 功能建模语言标准 ——IDEF0 语法与语义，2.1.6]

3.181

祖先图表 **ancestral diagram**

包含祖先框的图表。

[来源：IEEE 1320.1-1998 (R2004) IEEE 功能建模语言标准 ——IDEF0 语法与语义，2.1.7]

3.182

定位点 **anchor point**

软件进度安排中的一个里程碑，标志项目生存周期中发生重大阶段转换的节点。

[来源：PMBOK® 指南第五版的软件扩展]

3.183

标注主题列表 **annotated topic list; ATL**

有注释的、信息开发项目中包含的所有主题列表。

注：主题注释中可包含作者、何处使用、文件名和其他数据。

[来源：ISO/IEC/IEEE 26511:2018，3.1.2]

3.184

注解 **annotation**

随附于需求的补充文档。

[来源：ISO/IEC/IEEE FDIS 24765，3.177]

3.185

通告 **announcement**

由客户端对象发起的交互（调用），导致信息从该客户端对象传送到服务器对象，请求该服务器对象执行的功能。

示例：消息系统中消息发送的发起。

[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：体系结构，7.1.3]

3.186

年等值 **annual equivalent; AE(i)**

将现金流表示为在计划期内（按给定利率）的一系列等额年度支付。

同义词：AE(i)。

参见：未来值，现值。

[来源：ISO/IEC/IEEE 24765:2017；ISO/IEC/IEEE FDIS 24765，3.179]

3.187

年当量 **annual equivalent AE(i); AE**

将现金流量表示为计划范围内一系列等额的年度付款（按规定的利率）。参见：未来价值，现在价值。

[来源：ISO/IEC FDIS 24765，3.179]

3.188

年利率 **annual percentage rate; APR**

名义年度利息率。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.189

年金 **annuity**

在计划范围内定期等额支付的一系列金额。

[来源：ISO/IEC FDIS 24765, 3.181]

3.190

反常 anomaly

在系统文档或运行过程中观察到的任何与基于先前已验证的系统、软件或硬件产品或参考文档的预期结果不符的情况。

[来源：IEEE 1012-2024, 3.1]

3.191

先期缓冲 anticipatory buffering

一种缓冲技术，即预先将数据存储于缓冲区中，以备后续对数据的需要。

参见：动态缓冲、简单缓冲。

[来源：ISO/IEC FDIS 24765, 3.184]

3.192

先期调页 anticipatory paging

一种存储分配技术，即预先将页面从辅助存储器传输到主存储器中，以备后续对这些页面的需求。

参见：按需调页。

[来源：ISO/IEC FDIS 24765, 3.185]

3.193

运行可用性 Ao,; operational availability

运行可用性。

[来源：IEEE 982:2024, 3.2]

3.194

非周期任务 aperiodic task asynchronous task

异步任务

按需激活的任务。

[来源：ISO/IEC FDIS 24765, 3.191]

3.195

非周期性任务 aperiodic task

(异步任务) (asynchronous task)

按需激活的任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.191]

3.196

功能域的适用性 applicability to a functional domain

功能规模测量方法中，考虑与功能域中的功能规模测量相关的功能用户需求（FUR）特征的能力。

[来源：ISO/IEC TR 14143-3:2003, 3.2]

3.197

申请人 applicant

已提交申请并进入认证流程的人员。

[来源：ISO/IEC 24773-1:2019, 3.1]

3.198

应用程序 application

支持业务目标的自动化过程和数据的内聚汇集，由一个或多个组件、模块或子系统组成。

同义词：应用系统（application system），自动化信息系统（automated information system）。

示例：应付账款、应收账款、工资单、采购、车间生产、装配线控制、空中搜索雷达、目标跟踪、武器发射、航班安排和乘客预订。

[来源：GB/T 42588-2023]

3.199

应用管理职能 application administration function

由用户执行的职能，包括安装、配置、应用程序备份、维护（打补丁和升级）以及卸载。

[来源：GB/T 25000.51-2016, 4.1.3]

3.200

应用架构 application architecture

约束产品线内特定成员产品的架构性结构和规则（如通用规则和约束条件）。

注：应用架构描述产品线内特定成员产品的高级设计。产品线中的成员产品的应用架构重用（可能经修改后）领域架构的通用部分，并绑定领域架构的可变部分。大多数情况下，成员产品的应用架构需开发特定应用的可变性。

[来源：GB/T 41865-2022, X.X]

3.201

应用领域 application area

一类具有共同的组成或特征的应用，例如，相似的技术或生产方法、相似的客户类型（内部与外部、政府与商业）或者共同的行业部门（公用事业、汽车、航空航天、信息技术等）。

[来源：ISO/IEC FDIS 24765, 3.198]

3.202

应用资产 application asset

特定应用工程过程（如应用实现）的输出。此类资产可用于应用工程的其他生存周期过程，并可根据产品管理决策调整为领域资产。

注1：应用资产包括需求、架构设计、组件和测试。与需要在产品线内支持多种应用的大规模定制领域资产不同，多数应用资产不包含可变性。然而，应用可具有可变性（例如，最终用户可以通过在运行时绑定应用可变性，从而能够批量定制他们所使用的应用）。应用资产也就相应地具备了可变性，但应用资产的可变性仅服务于特定的应用，应用资产就是为这个应用而创建的。所以，应用资产可变性的范围通常比领域资产可变性的范围窄得多。

注2：应用资产并非现货或随时可供使用的物理产品，物理产品（如机械零件、电子元件、线束、光学镜片）根据其各自学科的最佳实践进行存储和管理。应用资产有其自身的生存周期；GB/T 22032 可用于其生存周期管理。

[来源：GB/T 41865-2022, X.X]

3.203

应用需求资产 application assets in requirements

在应用需求工程过程中产生的特定于应用的制品，例如应用需求规格说明和应用需求模型。

[来源：GB/T 45804-2025, X.X]

3.204

应用程序边界 application boundary

应用程序与其用户和/或其他应用程序之间的概念接口。当分析的范围确定后，应用程序边界可以被定义。

[来源：GB/T 42588-2023, 3.1.2]

3.205

应用组件 application component

为某一成员产品选择、复用或新开发的组件。

[来源: ISO/IEC 26553:2018, 3.1]

3.206

应用配置 application configuration

通过绑定可变性并添加应用程序特定可变性而得到的一个应用程序的结果。

[来源: ISO/IEC 26558:2017, 3.1]

3.207

应用设计 application design

生成符合领域架构的单个应用架构的应用工程过程。

[来源: GB/T 41865-2022, X.X]

3.208

应用工程 application engineering

通过重用符合领域架构的领域资产和绑定平台的可变性, 实施和管理应用资产和产品线成员产品的一系列过程组成的生存周期。

注: 传统意义上的应用工程是指开发单个的产品, 不需要领域资产的战略性重用, 也不需要明确的可变性建模和绑定。

[来源: GB/T 41865-2022, X.X]

3.209

应用工程过程 application engineering process

在产品线中开发成员产品的一组过程。

[来源: ISO/IEC 26555:2015, 3.1]

3.210

应用框架 application frameworks

由一组抽象类和具体类以及它们之间的接口组成的子系统设计。

注: 框架通常是许多模式的实例化。

[来源: ISO/IEC/IEEE 24765:2017, 3.177]

3.211

应用程序功能点计数 application function point count

用于衡量应用程序向最终用户所提供功能数量的计数。以功能点这一单位来进一步度量应用的功能规模。

注: 此处所度量的功能, 可以是已经提供给用户的, 也可以是尚待提供的。利用该方法, 还可以确定对已实现的应用程序进行支持所需的工作量。

[来源: ISO/IEC FDIS 24765, 3.208]

3.212

应用程序功能规模 application functional size

用功能点表示应用程序的大小, 由应用程序功能点计数确定。

注: 通过应用程序功能规模, 还可以确定支持已实现应用程序所需的工作量。

[来源: GB/T 42449-2023, 3.3]

3.213

应用[程序]生成器 application generator

一种代码生成器, 它在具体的应用领域为解决一个或多个问题, 产生代码。例如, 工资生成器。

[来源: GB/T 11457-2006, 2.68]

3.214

应用接口 application interface

由成员产品选用、复用或新开发的接口。

[来源：ISO/IEC 26553:2018, 3.3]

3.215

应用管理 application management

负责管理、支持、维护和更新现有应用程序及相关数据结构的全部任务与活动的领域。

注：应用管理包括所有任务、职责和活动，其目的是使应用程序在整个业务过程的生存周期中满足其所有者的需求和要求。

[来源：ISO/IEC 16350-2015, 4.2]

3.216

应用管理组织 application management organization

负责一个或多个应用程序管理的组织单元。

注：应用管理组织可以是与用户组织相关的内部或外部单位。

[来源：ISO/IEC 16350-2015, 4.3]

3.217

应用对象 application object

与应用程序直接相关或构成其一部分的组件。

示例：程序、源代码、数据库、文档、数据结构、测试文件和脚本。

[来源：ISO/IEC 16350-2015, 4.4]

3.218

应用组合 application portfolio

由应用管理组织或该组织内部的某个实体所管理的一组应用程序集合。

注：应用组合的范围可以是该应用管理组织的全部应用组合，也可以是某个客户组织或其部分实体的一个或多个应用程序。

[来源：ISO/IEC 16350-2015, 4.5]

3.219

应用问题 application problem

一种由最终用户提出，并要求通过信息处理来解决的问题。

[来源：GB/T 11457-2006, X.X]

3.220

应用程序接口(API)管理平台 application program interface (API) management platform

作为客户端请求的代理，旨在保护在线服务器的后端不会因过多查询而失效。

注：API 管理平台可限制每个实体每秒或每天的查询次数。通常，API 管理平台包含分析与使用情况报告、API 密钥与授权管理，以及实时更新的文档。

[来源：IEEE 7005:2021, 3.1]

3.221

应用程序接口管理平台 application program interface management platform; API management platform

客户端请求代理，以保护在线服务器的后端免受过多查询造成的瘫痪。

注：API 管理平台可以限制每个实体每秒或每天的查询次数。通常，API 管理平台包括分析和使用情况报告、API 密钥和授权管理以及实时更新的文档。

[来源：IEEE 7005 2021, 3.1]

3.222

应用程序界面 **Application programming interface; API**

用于创建与外部系统或服务的功能或数据交互的软件的一组不同格式的功能、协议、参数和对象。

示例：服务子例程套件的接口、一组返回响应数据的专用 URL，或可向物理设备（例如装配线上的机器人）发出的一组命令。

注：API 可以采用多种形式。一般来说，API 是一组明确定义的组件间通信方法。API 指定了与其他应用程序通信所需的信息和方法。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.6]

3.223

应用实现 **application realization**

应用工程的过程之一，该过程基于应用架构、应用资产和领域资产集合，开发应用资产（其中部分可能来自于领域资产）和成员产品。

[来源：ISO/IEC 26550:2015, 3.5]

3.224

应用需求分析 **application requirements analysis**

理解所有应用特定需求的子过程，通过建模来仔细检查不正确和不一致的应用需求，并分析和协商不能通过领域需求来满足的应用需求。

[来源：GB/T 45804-2025, 3.3]

3.225

应用需求获取 **application requirements elicitation**

识别与应用程序相关的利益相关方，获取应用特定需求并绑定适当的变体的子过程。

[来源：GB/T 45804-2025, 3.2]

3.226

应用需求管理 **application requirements management**

管理应用需求的可追溯性和变更的子过程。

[来源：GB/T 45804-2025, 3.6]

3.227

应用需求规格说明 **application requirements specification**

文档化应用特定需求并将其与领域需求规格说明集成的子过程，领域需求规格说明的变体被绑定。

[来源：GB/T 45804-2025, 3.4]

3.228

应用需求验证和确认 **application requirements verification and validation**

确认应用特定需求的一致性和可行性，并保证绑定的变体满足特定产品的需求的子过程。

[来源：GB/T 45804-2025, 3.5]

3.229

应用软件 **application software**

被设计用于帮助用户执行特定任务或处理特定类型问题的软件，区别于控制计算机本身的软件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.225]

3.230

专用集成电路 **application specific integrated circuit,; ASIC; ASIC**

为特定用途定制的集成电路。

同义词：专用集成电路（application-specific integrated circuit）。

[来源：ISO/IEC/IEEE 24765d:2015, X.X]

3.231

应用测试资产 application test asset

具有复用潜力的特定应用程序的测试资产。

[来源：ISO/IEC 26554:2018, 3.3]

3.232

应用测试 application testing

应用工程的子过程，通过复用领域测试工件来发现应用程序中的缺陷证据。

[来源：ISO/IEC 26554:2018, 3.2]

3.233

应用可变性模型 application variability model

特定应用的可变性模型，包括可变性绑定结果、应用特定修改的可变性和应用特定新增的可变性。

[来源：ISO/IEC 26558:2017, 3.2]

3.234

面向应用的语言 application-oriented language

一种计算机语言，具有用于某种单一应用领域的手段或记号。例如，计算机辅助指令和硬件设计语言。

一种面向问题的语言，其语句包含或汇集了用户职业的术语。

参见：写作语言、规格说明语言、查询语言

[来源：GB/T 11457-2006, 2.69]

3.235

应用特定组件 application-specific component

为特定成员产品开发的组件。

[来源：ISO/IEC 26553:2018, 3.5]

3.236

应用特定需求 application-specific requirements

适用于某个特定应用程序的需求，或未被领域需求所覆盖的需求。

[来源：ISO/IEC 26551:2016, 3.9]

3.237

利用提前量与滞后量 applying leads and lags

用来调整紧前和紧后活动之间的时间量的一种技术。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]

3.238

分配的努力 apportioned effort

一种活动，其中工作量按比例分配给某些可离散的工作，且不可再进一步拆分为可离散的工作。

参见序号 1544：分散的努力。

注：分摊工作量是用于衡量工作绩效的三种挣值管理（EVM）活动类型之一。

[来源：《项目管理知识体系指南》（PMBOK®指南）——第七版]

3.239

评估结果 appraisal findings

评估结果，指出了评估范围内流程改进的最重要问题、难题或机遇。

注：评估发现是从确凿的客观证据中得出的推论。

[来源：ISO/IEC/IEEE FDIS 24765, 3.234]

3.240

评估参与者 appraisal participants

在评估期间参与提供信息的组织单位成员。

[来源：ISO/IEC/IEEE FDIS 24765, 3.235]

3.241

评估小组组长 appraisal team leader

领导评估活动，且已满足评估方法所定义的关于经验、知识和技能资格标准的人员。

[来源：ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.236]

3.242

适用性的可识别性 appropriateness recognizability

用户能够识别产品是否满足其需求的能力。

[来源：ISO/IEC 25010:2023, 3.4.1]

3.243

批准 approval

由授权代表发出的通知，表明某一信息项似乎满足需求且内容完整。

注：这种批准并不意味着在双方关系中供应商满足需求的责任发生了转移。

[来源：ISO/IEC/IEEE 15289:2015, 5.1]

3.244

批准机构 approval authority

负责在系统生存周期内批准各项活动、制品及其他方面的个人（或多人）或组织（或多个组织）。

注：批准机构可以包括多个实体，例如个人或组织。这些实体可以包括具有不同批准级别和/或不同的关注领域。在
两方情况下，批准机构通常由采购方承担。在监管情况下，批准机构可以是第三方，例如政府组织或其代理人。

在其他情况下，例如购买由单方开发的货架产品，批准机构的独立性可能是采购方关心的一个问题。

[来源：ISO/IEC/IEEE 15026-1: 2019, 3.5.2]

3.245

批准的变更请求 approved change request

已通过集成变更控制流程处理并获得批准的变更请求。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]

3.246

批准的变更请求审查 approved change requests review

对已批准的变更请求进行审查，以核实它们是否已按批准的方式得到实施。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]

3.247

批准的修改 approved modification

对一项或多项拟议变更的处理，这些变更授权对任何 SCI 进行更改。

注：拟议变更与已批准的修改之间可能存在多对多关系。一项拟议变更可能导致多个 SCI 的修改（即使仅修改代码
及其测试用例）。一项修改可能源于多项拟议变更，这些拟议变更可能同时获得批准，也可能在修改仍在进行期
间分阶段获得批准。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.240]

3. 248

弧 arc

网的有向边,可以将一个位置连接到一个转换,或者将一个转换连接到一个位置,通常用箭头表示。

[来源: ISO/IEC 15909-1:2019, 2.1.1]

3. 249

弧形标注 arc annotation

可以包含常量、变量和运算符的表达式,用于标注网的弧。

注: 该表达式的值为弧的关联位置类型的多重集。

[来源: ISO/IEC 15909-1:2019, 2.1.1.3]

3. 250

架构师 architect

负责系统架构的个人、团队或组织。

[来源: ISO/IEC/IEEE FDIS 24765, 3.243]

3. 251

架构化 architecting

在所关注实体的整个生存周期中,对一个架构进行构想、定义、表达、记录、沟通,证明其正确实施、维护和改进。

[来源: GB/T 45630-2025, 3.1]

3. 252

架构设计 architectural design

定义硬件和软件组件及其接口的集合,以建立计算机系统开发框架。

参见: 功能设计。

[来源: ISO/IEC/IEEE FDIS 24765, 3.245]

3. 253

架构设计阶段 architectural design phase

生存周期阶段,在此阶段中,系统的总体架构得以开发,从而满足软件需求文档中规定的要求,并详细说明根据这些要求制定的实施计划。

[来源: ISO/IEC/IEEE FDIS 24765, 3.246]

3. 254

架构设计评审 architectural design review

联合采购方-供应商评审,以评估软件设计说明中所述软件架构设计的技术充分性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.247]

3. 255

架构结构 architectural structure

系统设计各组件的物理或逻辑布局及其内部和外部连接。

示例: 面向函数(结构化)设计、面向对象设计和面向数据结构的设计。

[来源: ISO/IEC/IEEE FDIS 24765, 3.248]

3. 256

架构风格 architectural style

通过共享结构和语义属性相关的系统族的特征。

示例: 管道和过滤器、层、基于规则的系统 and 黑板。

[来源: ISO/IEC/IEEE FDIS 24765, 3.249]

3. 257

架构 architecture

实体在其环境中的基本概念或属性，以及该实体及其相关生存周期过程的实现和演化的管控原则。

[来源：GB/T 45630-2025, 3.1]

3. 258

架构集合 architecture collection

由组织拥有并接受该组织整体治理和管理的一组架构。

注：集合中的架构可以相互关联（如产品线中的架构）。集合中的架构可以基于相同的参考架构。

[来源：ISO/IEC/IEEE 42020:2019, 3.4]

3. 259

架构定义 Architecture definition

根据选择标准评估候选方案，并让主动代理发现最佳架构，包括与其他系统的集成。

[来源：INCONE 系统工程手册 第4版, 第9章]

3. 260

架构定义记录 Architecture definition record

与架构定义相关的数据、信息或知识的永久性、可读形式。

[来源：INCONE 系统工程手册 第4版, 附录 E]

3. 261

架构定义策略 Architecture definition strategy

为确定满足需求的选定系统架构，需要考虑的方法、进度安排、资源和具体因素。

[来源：INCONE 系统工程手册 第4版, 附录 E]

3. 262

架构描述元素 architecture description element,; AD element

架构描述的已识别或已命名部分。

注：AD 元素包括：AD 中识别的利益相关方、关注点、利益相关方角度和方面；AD 中使用的 ADL、ADF、对应关系及对应方法；以及 AD 中包含的架构视图、视图组件、架构视点和模型种类。

[来源：GB/T 45630-2025; GB/T 45630-2025, 3.4]

3. 263

架构描述框架 architecture description framework,; ADF; ADF

在特定的应用域或利益相关方群体中建立的描述架构的约定、原则和实践。

参见架构框架。

示例：通用企业参考架构建模框架（GERAM）（ISO 15704:2019 附录 B），开放分布式处理参考模型（RM-ODP），统一架构框架（UAF），以及北约架构框架（NAF）。

注：架构描述框架促进架构视图和模型的组织结构化、描述的一致性、更大的重用潜力以及完整性。

[来源：GB/T 45630-2025, 3.5]

3. 264

架构描述语言 architecture description language,; ADL; ADL

具有语法和语义的表达方式，由一组用于描述架构的表示、约定和相关规则组成。

示例：架构分析和设计语言（AADL），ArchiMate，统一建模语言（UML），系统建模语言（SysML），UAF 概要文件。

[来源：GB/T 45630-2025, X.X]

3. 265

架构描述 architecture description; architectural description; AD

用于表达架构的工作产品。

注 1：工作产品是由过程产生的人工制品。

注 2：AD 是提供给利益相关方的信息的有形表示。AD 认为是一个信息部件。

[来源：GB/T 45630-2025, X. X]

3. 266

架构实体 architecture entity

在架构设计过程中被考虑、描述、讨论、研究或以其他方式处理的事物。

示例：以下是可以通过架构过程处理各类架构实体：企业、组织、解决方案、系统（包括软件系统）、子系统、业务、数据（作为数据元素或数据结构）、应用程序、信息技术（作为一个集合）、使命、产品、服务、软件项、硬件项、产品线、系统族、系统之系统、系统集成、应用程序集合。

注：当提及这些架构实体的架构本身时，通常将实体类别的名称置于“架构”一词之前。例如，当架构工作处理的对象是系统时，使用短语“系统架构”。架构实体的基本概念或属性通常旨在体现在实体的组件、组件之间关系以及该实体与其环境的关系中。

[来源：ISO/IEC/IEEE 42020:2019, 3. 6]

3. 267

架构评估 architecture evaluation; AE

针对指定评估目标对一个或多个架构所作的判断。

示例：在架构评估过程中可以进行各种判断，例如验证架构是否满足利益相关者的需求、评估架构相对于其预期用途的质量、评估架构或架构实体对其利益相关者的价值、确定架构实体是否满足其预期用途、提供有关架构实体的知识和信息以及识别与架构相关的风险和机遇。

[来源：ISO/IEC/IEEE 42030:2019, 3. 4]

3. 268

架构评估框架 architecture evaluation framework

用于以一致且可重复的方式评估架构的约定、原则和实践。

示例：架构权衡分析方法（ATAM）、方法框架和 QUASAR 方法以及备选方案分析（AoA）。

注：该框架本质上可以是通用的，也可以是特定于某个应用领域、待检查的关注点集合或某种方法论。对于具有多层或多级的实体，评估框架可由不同的子架构框架组成。这些可被定义并整合为综合架构框架包的一部分。

[来源：ISO/IEC/IEEE 42030:2019, 3. 5]

3. 269

架构框架 architecture framework

在特定应用领域和利益相关方的团体中，为描述架构所建立的约定、原则和实践。

示例 1：（ISO 15704 中的）通用企业参考架构、方法论（GERAM）是一种架构框架。

示例 2：（ISO/IEC 10746 中的）开放分布式处理参考模型（RM-ODP）是一种架构框架。

[来源：ISO/IEC/IEEE 42010:2011, 3. 4]

3. 270

架构可追溯性 Architecture traceability

架构特征的双向可追溯性。

[来源：INCONE 系统工程手册 第 4 版，附录 E]

3. 271

架构视图 architecture view

组成架构描述的一部分的信息部件。

参见：架构视点（architecture viewpoint）。

注：信息或数据视图处理由信息视点框定的信息相关的关注点。它包含如视图组件、概念数据模型、数据管理模型和数据访问模型，以及将这些组件连接在一起的对应关系。

[来源：GB/T 45630-2025, 3.7]

3.272

架构视点 **architecture viewpoint**

创建、解读和使用架构视图以框定一个或多个关注点的一组约定。

注 1：在本文件中，“框定”关注点意为“塑造、组成、表示”这些关注点。它用于区分由视角框定关注点的阶段和在结果视图中处理这些关注点的阶段。这类似于“框定问题”和“解决问题”之间的区别。

注 2：视点是架构师决定的与架构描述的目的相关的关注点的参考框架。

注 3：架构视点的约定记录在该视点的规格说明中。在一些群体和架构描述框架中，“视图规格说明”和视点是同义词。

注 4：视点的识别通常是该视角所适用域的先验知识、经验和实际运用的结果，表明与解决关注点相关的信息。

[来源：GB/T 45630-2025, 3.8]

3.273

架构驱动现代化 **architecture-driven modernization; ADM**

理解和演进所关注系统的现有软件资产的过程。

[来源：ISO/IEC 19506:2012, 4]。

注：ADM 不排除（在适当情况下）源代码到源代码的迁移，但鼓励用户组织从分析和设计的角度考虑现代化。

3.274

档案页面 **archival page**

作为记录保存且预计不会更改的内容。

注：由于技术升级，某些存档页面除非与活跃页面一同进行升级，否则无法直接呈现。

[来源：ISO/IEC/IEEE 23026:2015, 4.1]

3.275

档案 **archive**

在运行时环境中已不存在，但可用于审计、监管及其他流程检查的系统元素的位置。

[来源：IEEE 2675-2021, 3.1]

3.276

归档 **archiving**

将文档的一个版本存入较少使用的存储区域的过程。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.3]

3.277

自变量，变元，变元的值，自变量的值 **argument**

在调用软件模块时用于指定要传递给该模块的数据或程序元素的常量、变量或表达式。

例如：方程 $E = mc^2$ 中的变量 m 。

[来源：ISO/IEC/IEEE FDIS 24765, 3.265]

3.278

元数 **arity**

参与关系的角色数量。

注：二元关系的元数为 2。n 元关系的元数为 $n \cdots (n > 2)$ ，有时也称为关系的“度”。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 279

安排 `arranging`

事务功能中对属性排序的活动。

[来源：GB/T 42449-2023, 3.5]

3. 280

数组 `array`

由单个名称和一个或多个索引标识的 n 维有序数据项集合，集合中的每个元素都可以单独寻址。

例如：矩阵、表格或向量。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.269]

3. 281

箭头 `arrow`

有向线，由一个或多个连接的箭头段组成，在同一张图中从单个源点（方框或图表边界）指向单个用途点（方框或图表边界）。

2. 以图形方式表示进度活动之间逻辑关系，用于优先图法。

参见：箭头段、边界箭头、内部箭头。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.270]

3. 282

箭头标签 `arrow label`

与箭头段相关的名词或名词短语，表示箭头段的箭头含义。

注：具体来说，箭头标签标识箭头段所表示的对象类型集。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.271]

3. 283

箭头含义 `arrow meaning`

对象类型集中的对象类型，无论这些对象类型如何在对象类型集内进行收集、聚合、分组、捆绑或以其他方式连接。

例如：物理对象、数据元素。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.272]

3. 284

箭头角色 `arrow role`

由箭头段表示的对象类型集合与由箭头段所连接的方框表示的活动之间的关系。

注：箭头有四种角色：输入、控制、输出和机制。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.273]

3. 285

箭头段 `arrow segment`

从方框边、箭头连接点（分支或连接）或图表边界出发，终止于路径上下一个方框边、箭头连接点（分支或连接）或图表边界的有向线。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.274]

3. 286

制品 `artifact`

由某一种软件开发和运维过程所使用的或产生的一种信息的物理件。制品的实例有模型、源文件、文字和二进制可执行文件等。制品构成可部署构件的实现。

[来源：GB/T 11457-2006, 2.76]

3. 287

制品证据 artifact evidence

表征制品是否满足期望的制品可信度的证据。

注：该证据的数据类型分为布尔型、数值型、百分比型和等级型。

[来源：GB/T 37970-2019, 3.1.12]

3. 288

制品可信度 artifact trustworthiness

软件过程产生的制品能够满足期望产品质量的信心度。

[来源：GB/T 37970-2019, 3.1.6]

3. 289

制品证据测量元 artifact-evidence element

用于测量制品证据的测量元。

[来源：在“近年 SC7 发布国家标准术语汇总”中有，但是在 GB/T 37970-2019 中没有找到]

3. 290

人工智能 artificial intelligence; AI

一组方法或自动化的实体共同构建、优化和应用一个模型，使系统能够针对一组预定义任务进行预测、推荐或决策计算。

[来源：IEEE 1012-2024 IEEE, 3]

3. 291

人工智能效应 artificial intelligence effect; AI effect

随着技术进步，先前被标记为人工智能（AI）的系统不再被视为 AI 的情况。

[来源：ISO/IEC TR 29119-11:2020, 3.1.10]

3. 292

人工智能质量元模型 artificial intelligence quality metamodel; AI quality metamodel

旨在确保基于人工智能（AI）的系统的质量的元模型。

[来源：ISO/IEC TR 29119-11:2020, 3.1.11]

3. 293

人工智能系统 artificial intelligence system; AIS; AIS

为给定的一组人为定义目标生成内容、预测结果、建议或决策等输出的工程化系统。

[来源：IEEE 1012-2024, 3]

3. 294

基于人工智能的系统 artificial intelligence-based system; AI-based system

包含一个或多个实现人工智能（AI）的组件的系统。

[来源：ISO/IEC TR 29119-11:2020, 3.1.9]

3. 295

人工语言 artificial language

其规则在使用前被明确定义的语言。

参见：自然语言。

示例：编程语言和数学语言。

[来源：ISO/IEC 2382:2015, X.X]

3. 296

要求 ask

特定活动、由特定时效性函数定义的所需执行时间以及特定任务模式的组合。

[来源：ISO/IEC 14756:1999, 4.19]

3.297

方面 `aspect`

实体特征或本质的一部分。

示例：实体的功能、结构和信息方面。

注：特定方面可用于捕捉感兴趣实体的相关特征，作为对其特征某些部分的一个或多个关注点的细化。方面帮助架构师分析、处理和结构化关注点。一般来说，方面和关注点之间存在多对多的关系。

[来源：GB/T 45630-2025, 3.9]

3.298

汇编 `assemble`

把用汇编语言表示的程序翻译成等效的机器语言程序，。

参见：编译、反汇编、解释。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.287]

3.299

汇编并运行 `assemble-and-go`

一种操作技术，其中计算机程序的组装、链接、加载和执行之间没有停顿。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.288]

3.300

汇编的起点 `assembled origin`

由汇编程序、编译器或链接编辑器分配给计算机程序的初始存储位置的地址。

参见：加载基址、偏移量、起始地址。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.289]

3.301

汇编程序 `assembler`

一种计算机程序，用于将汇编语言编写的程序转换为对应的机器语言。

参见：绝对汇编程序、编译器、交叉汇编程序、解释器、重定位汇编程序。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.290]

3.302

组合 `assembly`

一组单元，其中不同单元上的端口以兼容方式互连，使得一个单元的行为能够通过互连端口发生的交互来影响另一单元的行为。

注：一个组合可被视为在行为上等同于一个单元。该单元的端口正是组合中各单元未在内部互连的那些端口。该单元的行为是组合中各单元共享行为的组合。

[来源：IEEE 1175.4-2008, 3.3]

3.303

汇编代码 `assembly code assembler code`

用汇编程序能识别和处理的形式表示的计算机指令和数据定义。

同义词：汇编程序代码（`assembler code`）。与编译程序代码（`compiler code`）、解释代码（`interpretive code`）和机器代码（`machine code`）相对。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.291]

3.304

汇编语言 assembly language assembler language low-level language second-generation language

汇编语言

低级语言

第二代语言

一种编程语言，它非常接近给定计算机的指令集，允许用符号命名操作和地址且通常是程序指令与计算机指令一一对应，且能提供使用宏指令的便利。

参见：第五代语言、第四代语言、高阶语言、机器语言。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.292]

3.305

汇编语言 assembly language, ; assembler language, ; low-level language, ; second-generation language

一种编程语言，它非常接近给定计算机的指令集，允许用符号命名操作和地址且通常是程序指令与计算机指令一一对应。

参见：第五代语言、第四代语言、高阶语言、机器语言。

同义词：汇编语言、低级语言、第二代语言。

[来源：ISO/IEC/IEEE 24765:2017;GB/T 11457, 2.86]

3.306

断言 assertion

一种逻辑表达式，它规定必须存在的一种程序状态，或规定在程序执行过程中某一特定点上程序变量必须满足的条件集合。类型包括输入断言、循环断言、输出断言。

参见：不变式（invariant）和正确性证明（proof of correctness）。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.293]

3.307

评定的能力 assessed capability

按本技术报告的规定，所完成的一次或多次最近的、相关的过程评定的输出。

3.308

评定 assessment

基于预先定义的准则、标准与需求，对软件产品、过程、项目或从业人员的当前状态、合规性、风险、能力差距等进行系统的度量和分析，并为改进和决策提供依据的过程。

3.309

评估机构 assessment body

执行评估的机构。

注：主体可以是执行评估的组织或组织的一部分。

[来源：ISO/IEC 33001:2015, 3.2.1]

3.310

评估约束 assessment constraint

对评估结果的使用或评估小组在开展评估时自主选择空间所施加的限制。

[来源：ISO/IEC 33001:2015, 3.2.2]

3.311

评估指标 assessment indicator

用于支持评估人员对过程属性进行评级判断的客观证据来源。

[来源：ISO/IEC 33001:2015, 3.3.1]

3.312

评估输入 assessment input

开展流程评估前所需的信息。

[来源：ISO/IEC 33001:2015, 3.2.3]

3.313

评定工具 assessment instrument

在评定中所用的一个或一组工具，它用以帮助评定员对过程性能或能力进行评定、处理评定数据和记录评定结果。

3.314

评估输出 assessment output

评估所取得的实质性成果。

参见：评估记录。

[来源：ISO/IEC 33001:2015, 3.2.4]。

3.315

评估参与者 assessment participant

在评估范围内承担相应职责的人员。

示例：评估发起人、评估员和组织单位成员。

[来源：ISO/IEC 33001:2015, 3.2.5]

3.316

评估目的 assessment purpose

作为评估输入的一部分提供的声明，用于明确开展评估的原因。

[来源：ISO/IEC 33001:201, 3.2.6]

3.317

评估记录 assessment record

对与评估有关的信息进行有序的、文件化的汇集，有助于理解和验证评估生成的过程概况。

[来源：ISO/IEC 33001:2015, 3.2.7]

3.318

评估范围 assessment scope

作为评估输入一部分提供的评估边界定义，包括受评估组织单元的边界、纳入评估的流程、各流程需达到的质量等级，以及流程运行的环境。

[来源：ISO/IEC 33001:2015, 3.2.8]

3.319

评估发起者 assessment sponsor

受评组织单元内部或外部的个人或实体，其提出评估需求，并为开展评估提供资金及其他资源。

[来源：ISO/IEC 33001:2015, 3.2.9]

3.320

评估小组 assessment team

共同执行过程评估的一个或多个个体。

[来源：ISO/IEC 33001:2015, 3.2.10]

3.321

评估员 **assessor**

参与过程属性评级的个人。

[来源：ISO/IEC 33001:2015, 3.2.11]

3.322

资产 **asset**

对组织具有潜在或实际价值的项、事物或实体。

注 1：资产从不同的角度分类，例如有形资产、无形资产；动产、不动产。无形资产分为数字资产和非数字无形资产。

注 2：认知资产是指组织在运营过程中产生的无形资产。数据、信息、知识、智慧和建模资产都属于认知资产。

[来源：GB/T 45615-2025, X.X]

3.323

资产库 **asset base**

由领域工程与应用工程共同产生的可重用资产集合。

[来源：GB/T 41865-2022, 3.6]

3.324

资产寿命 **asset life**

资产由创建到结束之间的时间。

[来源：ISO/IEC 19770-1:2017, 3.2]

3.325

资产管理 **asset management; AM**

组织利用资产实现价值的协作活动。

[来源：ISO/IEC 19770-5:2015, 3.3]

3.326

资产管理计划 **asset management plan**

规定单项资产或一组资产的活动、资源和进度的文件化信息，旨牵头现组织的资产管理目标。

[来源：ISO/IEC 19770-1:2017, 3.4]

3.327

资产管理体系 **asset management system**

资产管理方面用于建立资产管理方针和资产管理目标的管理体系。

[来源：ISO/IEC 19770-1:2017, 3.5]

3.328

资产组合 **asset portfolio**

资产管理体系范围内的资产。

注：组合通常为实现管理控制目的而建立并分配。实物资产的组合可按类别划分，例如厂房、设备、工具、土地等。

软件组合则可按软件发布商或平台（例如 PC、服务器、大型机）进行定义。

[来源：ISO/IEC 19770-1:2017, 3.6]

3.329

资产建议书 **asset proposal**

包括主要资产（所有应用的功能域和高层级通用与可变特征）的制品，这些资产包含在产品线中，并包含量化的成本和收益以及评估结果信息。

[来源：GB/T 45804-2025, 3.8]

3.330

资产保护 **asset protection**

对于服务而言，指该服务通过相关流程，保护用于提供受保服务的物理设施，避免数据丢失、连接中断及必要基础设施与 IT 设备不可用，并在运行过程中保障受保服务安全的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.5.6]

3.331

资产范围界定 **asset scoping**

确定潜在领域资产和评估资产投资回报的过程。

注：资产范围界定过程中产生的信息，以及产品范围界定和领域范围界定过程中产生的信息，可用于确定是否将产品线引入组织。资产范围界定在领域范围界定之后进行。

[来源：ISO/IEC 26550:2015, 3.7; GB/T 41865-2022, 3.7]

3.332

资产系统 **asset system**

一组相互关联或相互作用的资产。

[来源：ISO/IEC 19770-1:2017, 3.7]

3.333

资产类型 **asset type**

具有共同的特征能够区分为一组或一类资产的资产分类。

示例：实物资产、信息资产、无形资产、关键资产、使能资产、线性资产、信息和通信技术（ICT）资产、基础设施资产、动产。

[来源：ISO/IEC 19770-1:2017, 3.8]

3.334

赋值 **assignment**

对于一组变量，将一个（正确类型的）值与每个变量相关联。

参见：绑定。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.319]

3.335

赋值语句 **assignment statement**

为变量赋值的一种计算机程序语句。

参见控制语句、声明、清除、初始化、重置。

示例： $Y = X - 5$ 。

[来源：ISO/IEC/IEEE FDIS 24765, 2025, 3.320]

3.336

辅助技术 **assistive technology**

用于增强、维持或改进个人能力的设备、产品、系统、硬件、软件或服务。

示例：盲文显示器、屏幕阅读器、屏幕放大软件和眼动追踪设备。

注：辅助技术可以包括辅助服务以及评估、推荐和提供所需的专业服务。

[来源：ISO/IEC 29110-5-1-2:2025, 3.6]

3.337

关联 **association**

在 UML 中，演员和用例之间的关系表明演员通过用例 2 与系统交互。独立于支持特定计算交互的协议交换建立的协议对象之间（或协议对象与拦截器之间）的关系（绑定）。

[来源：ISO/IEC 14752:2000, 3.3.2]

3.338

关联类 **association class**

一种兼有关联与类双重性质的模型元素。可以将关联类视为具有类性质的关联，或视为具有关联性的类。

3.339

关联端 **association end**

对某一关联，将其连接到类目上的那个端点。

3.340

关联管理设施 **association management facility**

一组支持管理协议对象之间关联的服务原语。

[来源：ISO/IEC 14752:2000, 3.3.3]

3.341

联想类 **associative class**

为解决多对多关系而引入的类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.325]

3.342

关联实体 **associative entity**

用于表示其他实体之间关系的实体，否则提供足够的机制。

注：当一个关系无法通过其他方式提供足够的机制时，就需要使用关联实体。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.343

关联实体类型 **associative entity type**

包含属性的实体类型，这些属性进一步描述了两个其他实体类型之间的多对多关系。

[来源：GB/T 42449-2023, 3.6]

3.344

关联字面量 **associative literal**

表示实例值的字面量。

注：关联字面量的表达格式为：类名 属性名：属性值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.328]

3.345

假设 **assumption**

在规划过程中，被视为真实、实际或确定，而无需证明或论证的因素。

[来源：《项目管理知识体系指南（PMBOK®指南）》第七版]

3.346

假设和约束分析 **assumption and constraint analysis**

有助于确保假设与约束条件纳入项目计划和文档，并使其保持一致性的评估。

注：已修改，将“确保”改为“有助于确保”。

[来源：项目管理知识体系指南（PMBOK(R) Guide)—第七版]

3.347

假设日志 **assumption log**

用于记录项目过程中所有假设和约束的项目文件。

[来源：项目管理知识体系指南（PMBOK®指南）—第七版]

3. 348

假设分析 assumptions analysis

一种探索假设的准确性并从假设的不准确、不一致或不完整性中识别项目风险的技术。

[来源：项目管理知识体系指南（PMBOK®指南）—第五版]

3. 349

保证论据 assurance argument

将实质性证据和假设联系起来，在特定背景下为主张提供令人信服和有效论据的制品。

注：当且仅当一个论证满足：若所有前提为真，则结论必然为真，该论证才是有效的；当且仅当一个论证既有效，且所有前提均为真，该论证才是可靠的。

[来源：ISO/IEC/IEEE 15026-4:2021, 3.2]

3. 350

保证案例报告 assurance case report

提供保证案例主张及其论据、证据的完整索引，能从索引组合形成所关注保证案例的可审计制品。

注：保证案例报告是对保证案例的介绍。其内容因案例而异。对于简单和/或小型保证案例，报告可能是多余的。对于较大或更复杂的保证案例，报告可以包括系统描述、部分论证及其支持证据，例如更高级别的论证及其支持证据。保证案例报告通常作为保证案例的定期更新，在预先约定的时间点（例如生存周期阶段结束时）发布。它们报告迄今为止已完成的保证活动的结果、对总体进展的评估以及对保证活动绩效的审查。供应商可以在阶段退出/进入的决策关口使用保证案例报告，收购方也可以在项目监控中使用保证案例报告。

[来源：ISO/IEC/IEEE 15026-2:2022, 3.1.2]

3. 351

保证索赔 assurance claim

需核定保证事项的索赔。

[来源：ISO/IEC/IEEE 15026-4:2021, 3.3]

3. 352

保证信息 assurance information

包含关于系统的主张、支持该主张的证据、论证证据如何支持该主张成立，以及上述内容背景信息的信息。

注：当顶层主张隐含要求某项属性持续达成时，保证信息论证中所包含的子主张，可以与所关注系统的生存周期相关。

[来源：ISO/IEC/IEEE 15026-4:2021, 3.4]

3. 353

保证目标 assurance objective

实现保证主张的目的。

注：保证目标决定了保证信息中所需的完整性水平和允许的不确定性。

[来源：ISO/IEC/IEEE 15026-4:2021, 3.5]

3. 354

异步的 asynchronous

两个或多个相互独立、不依赖特定事件（如统一时序）发生的过程。

[来源：ISO/IEC 2382:2015, X.X]

3. 355

异步通信接口适配器 asynchronous communication interface adapter; ACIA

连接异步通信接口的功能单元。

[来源: ISO/IEC/IEEE FDIS 24765, 3.340]

3.356

异步输入输出设备 asynchronous I/O device

I/O 设备在产生一些输入或产生一些输出后产生中断。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.357

异步输入输出设备接口任务 asynchronous I/O device interface task

与 I/O 设备接口并由来自该设备的中断激活的任务。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.358

异步消息通信 asynchronous message communication

松耦合消息通信 loosely coupled message communication

生产者任务向消费者任务发送消息并且不等待响应的通信。

注: 消息队列可以在任务之间建立。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.359

异步消息通信 asynchronous message communication, ; loosely coupled message communication

生产者任务向消费者任务发送消息且不等待响应的通信方式。

注1: 任务之间可以建立消息队列。

同义词: 松耦合消息通信。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.360

原子组件模型 atomic model

对虚拟机系统中不可再分的最小功能模块的形状化描述模型。

[来源: GB/T 36457-2018, 3.12]

3.361

原子类型 atomic type, ; primitive type

数据类型, 其每个成员由单个不可分解的数据项组成。

参见: 复合类型。

同义词: 原语类型。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.362

原子性 atomicity

给定抽象层级上, 该层级下不可再细分的实体。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.363

附加进程 attached process

定义每项资产如何在应用中使用的进程。

[来源: ISO/IEC 26555:2015, 3.2]

3.364

攻击 attack

恶意行为或与系统或其环境的交互可能导致故障或错误（并因此可能导致故障）或不利后果，或产生不利后果。

[来源：ISO/IEC/IEEE 15026-1:2019, 3.4.7]

3.365

属性 attribute

可由人或自动化工具定量或定性辨别的实体特征或特性。

[来源：GB/T 41866.1-2022, X.X]

3.366

质量测度的属性 attribute for quality measure

与软件产品本身、与软件产品的使用或者与其开发过程有关的属性。

注：用于质量测度的属性被用来获得质量测度元素。

[来源：GB/T 25000.40-2018, 4.4]

3.367

属性名称 attribute name

属性值类的角色名称。

[来源：ISO/IEC/IEEE FDIS 24765, 3.351]

3.368

属性抽样 attribute sampling

一种质量测量方法，其过程为：记录所考察的每个单元中是否存在某些特征（属性）；在对每个单元进行检验后，决定接收该批、拒收该批，或对另一个单元进行检验。

[来源：《项目管理知识体系指南（PMBOK® 指南）》第五版]

3.369

属性关系 attributed relationship

具有属性的关系。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.370

属性实体类型 attributive entity type

进一步描述另一个实体类型的一个或多个属性的实体类型。

参见：实体

[来源：GB/T 42449-2023, 3.7]

3.371

受众 audience

具有相同或相似特性与需要的用户类别，这些特征与需求（如用户使用信息的目的、任务、受教育程度、能力、培训情况及经验）决定了目标信息的内容、结构和使用方式。

参见：人物角色。

注：用户的信息可能有不同的受众，例如管理、数据输入、维护、工程、业务专业人员。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.2]

3.372

审核 audit

为评估制品或制品集是否符合规范、标准、合同协定或其他准则而进行的独立检查。

注：审核可以是内部审核（第一方）或外部审核（第二方或第三方），也可以是联合审核或一体化审核（结合两个或多个领域）。审核会明确表明是否满足审核准则。审核范围可包括专业及行业执业规范。“审核”适用于管理体系，而“评定”既适用于合格评定机构，也具有更广泛的适用范围。在可行的情况下，生成支持审核的信息技术（IT）控制证据通常可自动化实现。

[来源：ISO/IEC/IEEE 12207:2026, 3.1.11]

3.373

审核组 audit team; AT

由一名或多名审核员实施审核，必要时由技术专家提供支持。

注：审核组中应指定一名审核员担任审核组长。审核组可包含实习审核员。

[来源：ISO/IEC 29110-12:2024, 3.13]

3.374

可审核性 auditability

对于服务，指其收集并提供与服务运行和使用相关的必要证据信息，以供开展审核的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.8.2]

3.375

被审核方 auditee

被审核的组织。

[来源：ISO/IEC 29110-1-2:2024, 3.11]

3.376

审核员 auditor

进行审核的人。

[来源：ISO/IEC 29110-1-2:2024, 3.12]

3.377

增强现实系统 augmented reality system

由计算机生成的文本、图像、数据或其他媒体补充的物理世界视图。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.4]

3.378

真实性 authenticity

产品证明某一主体或资源的身份与其所声称身份一致的能力。

[来源：ISO/IEC 25010:2023, 3.6.5]

3.379

作者检查 author check

工作产品作者进行的非正式审查。

[来源：ISO/IEC 20246:2017, 3.2]

3.380

创作环境 authoring environment

用于创建、存储和管理内容单元的工具集。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.5]

3.381

写作语言 authoring language

用于开发计算机辅助指令的课件的高级编程语言。

参见：写作系统。

[来源: ISO/IEC/IEEE FDIS 24765, 3.367]

3.382

写作系统 **authoring system**

与写作语言形成一体化的编程系统。

[来源: ISO/IEC/IEEE FDIS 24765, 3.368]

3.383

批准 **authorization**

表示未阻止特定行为的操作。

[来源: ISO/IEC 15414:2015, 6.6.4]

3.384

授权个人 **authorized individual**

组织指定的在责任范围内做出决策、分配资源和接受风险的个人。

[来源: IEEE 7002:2022, 3.1]

3.385

使自动化 **automate**

使过程或设备自动的。

[来源: ISO/IEC 2382:2015, X.X]

3.386

自动设计工具 **automated design tool**

帮助进行软件设计的综合、分析、模拟或文档编制的软件工具。

示例: 仿真器、分析工具、设计表示处理器和文件生成器。

3.387

自动化系统流程 **automated systems process**

辅助流程 **assisted process**

辅助软件流程 **assisted software process**

辅助系统流程 **assisted systems process**

自动化软件流程 **automated process automated software process**

由计算机辅助软件工程 (CASE) 工具完全或部分支持执行的系统或软件过程。

[来源: ISO/IEC 15940:2013, 2.9]

3.388

自动测试生成器 **automated test generator**

一种软件工具, 它以计算机程序和准则作为输入, 产生满足这些准则要求的测试输入数据, 有时还确定预期的结果。

3.389

自动验证系统 **automated verification system**

一种软件工具, 它以计算机程序及其规格说明的表示作为输入 (可能借助人的帮助), 产生该程序的正确与否的证明。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.390

自动验证工具 **automated verification tools**

用于评估软件开发过程中的产品的一类软件工具。这些工具有助于验证正确性、完全性、一致性、可跟踪性、可测试性, 以及检查是否遵守了标准。软件验证工具包括设计分析器、自动验证系统。

3.391

自动的 automatic

在规定条件下，无需人工干预即可运行的过程或设备。

[来源：ISO/IEC 2382:2015, X.X]

3.392

自动化 automation

将过程或设备转换为自动运行，或转换的结果。

[来源：ISO/IEC 2382:2015, X.X]

3.393

自主增长 autonomous growth

在功能细化时对初始未识别但隐含在需求中的功能点展示。

[来源：GB/T 42588-2023, X.X]

3.394

自主系统 autonomous system

能够在无人干预的情况下持续运行的系统。

参见：自主/智能系统。

[来源：ISO/IEC TR 29119-11:2020, 3.1.14]

3.395

自主/智能系统 autonomous/intelligent system; A/IS

半自主或自主的计算机控制系统，通过编程执行某项任务，可在无人干预或有限人工干预下运行，能够通过独立推理进行决策，并成功适应其所处环境。

参见：自主系统。

[来源：IEEE 7010-2020, 2.1]

3.396

自主性 autonomy

系统在无人干预的情况下持续工作的能力。

[来源：ISO/IEC TR 29119-11:2020, 3.1.15]

3.397

基于自主性的改进 autonomy-based improvement

主动自发、自主开展的专业流程改进，基于对工作目标、流程要求、最新技术及产品应用结果的充分理解。

[来源：ISO/IEC 29110-1-2:2024, 3.14]

3.398

辅助类 auxiliary class

以实现二级逻辑式或控制流为典型方式，来支持另一更靠中央或更基础的类的一种衍型类。辅助类与焦点类一起使用具有代表性，且在设计期间，对规定各构件的二级业务逻辑或控制流特别有用。

参见：焦点类（focus class）。

3.399

可用性 availability

产品在需要使用时可操作和可访问的能力。

[来源：ISO/IEC 25010:2023, 3.5.2]

3.400

可用性模型 availability model

用于预测、估计、判定可用性的模型。

3.401

可用交互 available interaction

一个单元允许、由另一个单元控制的交互。

[来源：IEEE 1175.4-2008, 3.4]

3.402

意识要求 awareness requirement

参考其他要求或领域假设及其成功或失效。

[来源：IEEE 7005 2021, 3.1]

3.403

后置辅文 back matter

出现在印刷文档末尾（如索引等）的材料。

[来源：ISO/IEC/IEEE FDIS 24765, 3.387]

3.404

后台 background

在作业调度中，指执行低优先级进程或不需要用户交互的进程的计算环境。参见：前台、后台处理。

[来源：ISO/IEC/IEEE FDIS 24765, 3.389]

3.405

后台处理 background processing

在优先级更高的进程未使用计算机资源时执行低优先级进程，或者执行不需要用户交互的进程。另见前台处理。

[来源：ISO/IEC/IEEE FDIS 24765, 3.390]

3.406

存货 backlog

特性或用户故事的集合，包括功能性和非功能性需求，通常根据价值优先级排序。

[来源：ISO/IEC 33202:2024, 3.9]

3.407

待办清单细化 backlog refinement

逐步细化待办事项列表中的内容，并对其进行（重新）优先级排序，以确定在即将到来的迭代中可以完成的工作。

[来源：项目管理知识体系指南（PMBOK(R) Guide）— 第七版]

3.408

回退 backout

撤销一次提交所产生的影响。

注：通常通过引入新的提交将内容恢复到之前的状态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.393]

3.409

比对测试 back-to-back testing

一种测试方法，其中程序的两个或多个变体使用相同的输入执行，比较输出，并在出现差异时分析错误。

[来源: ISO/IEC TR 19759:2016, 4.2.2.9]

3.410

备份 backup

在发生故障或外部灾害时, 可用于替代或帮助恢复主要项目的系统、组件、文件、程序或人员。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.411

备份及恢复测试 backup and recovery testing

可靠性测试的一种类型。在失效事件中, 在规定的时问、成本、完整性和准确性参数范围内, 测量系统状态可以从备份中恢复的程度。

[来源: GB/T 38634.1-2020, 3.3]

3.412

后备程序员 backup programmer

首席程序员团队的助理领导。参见: 首席程序员。

注: 职责包括为团队开发的软件贡献重要部分, 协助首席程序员审查其他团队成员的工作, 必要时替代首席程序员, 并对所开发的软件有全面的技术理解。

[来源: ISO/IEC/IEEE 24765 FDIS, 2025]]

3.413

巴科斯范式 Backus-Naur Form

用于以文本格式去定义一种语言的语法的形式化元语言。

[来源: GB/T 38634.4-2020, 4.1]

3.414

后向执行 backward execution

一种调试技术, 在该项技术中记录了程序执行的历史, 并于此后在用户控制下, 以向前或向后的方向进行重演。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3951]

3.415

反向传播 backward propagation

一种在人工神经网络中使用的方法, 根据网络输出端的计算误差确定网络连接的权重。

注: 用于训练深度神经网络。

[来源: ISO/IEC TR 29119-11:2020, 3.1.17]

3.416

后向恢复 backward recovery

一种恢复类型, 其中系统、程序、数据库或其他系统资源被恢复到它可执行所需功能的先前状态。参见: 正向恢复。

[来源: ISO/IEC/IEEE FDIS 24765, 3.399 backward recovery]

3.417

包 bag

成员无序但重复项有意义的集合类。

参见: 列表、集合 [来源: ISO/IEC/IEEE FDIS 24765, 3.400]。

3.418

球栅阵列 ball grid array; BGA

底面以栅格形式分布多个连接点的表面贴装集成电路封装。

注：与仅在边缘设有连接器的封装相比，这种封装提供更多连接点。

[来源：ISO/IEC/IEEE 24765c:2014, X.X]

3.419

基地址 base address

作为一个基准点的地址，它用于与一个相对地址相加，以确定要访问的存储单元的地址。参见：索引地址、相对地址、自相关地址。

[来源：ISO/IEC/IEEE FDIS 24765, 3.402]。

3.420

基本选择 base choice

通常根据参数的代表性或典型值选择的输入参数值。

同义词：基本值 (base value)

[来源：ISO/IEC/IEEE 29119-1:2022, 3.11]

3.421

基类 base class

B 实例的模板类 CB 与 A 实例的模板类 CA 的关系，其中模板 A 是模板 B 的增量修改
参见：派生类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.404]

3.422

基础功能组件 base functional component; BFC

功能规模测量方法中为了测量目标而定义和使用的用户功能需求的基本单位。

注 1：功能用户需求可以是“维护客户”，它可能由下列基本功能部件组成：“添加新客户”，“报告客户购买情况”和“更改客户详细信息”；

注 2：“客户详细信息”之类的在研的软件所维护的逻辑相关业务数据的汇集。

[来源：ISO/IEC 14143-1:2007, 3.1]

3.423

基本功能部件 (BFC) 功能服务 base functional component (BFC) functional service; BFC
由 FSM 方法定义和使用的功能用户需求的基本单元，用于测量目的。

示例 1：“维护客户”功能用户需求由以下 BFC 组成：“添加新客户”、“报告”客户购买”和“更改客户详细信息”。逻辑上相关的业务数据的集合由软件维护为“客户详细信息”。[来源：ISO/IEC/IEEE FDIS 24765, 3.409 base functional component。

示例 2：BFC。

示例 3：functional service]。

[来源：ISO/IEC 19761:2011, 2.1]

3.424

基础功能组件类型 base functional component type

已定义的基础功能部件的类别。

[来源：ISO/IEC 14143-1:2007, 3.2]

3.425

基本功能部件类型 (BFC 类型) base functional component type (BFC type); BFC type

已定义的基本功能部件的类别。

[来源：ISO/IEC 14143-1:2007, 3.2]

[来源：GB/T 42452-2023]

3.426

基本测度 base measure

按照某个属性及其量化方法定义的测度。

注：一个基本度量在功能上独立于其他测度。

[来源：ISO/IEC/IEEE 15939:2017, 3.3]

3.427

基本实践 base practice; BP

为实现某个特定过程目的的持续活动。

[来源：ISO/IEC 33001:2015, 3.3.2]

3.428

基本标准 base standard

经批准的国际标准或国际电信联盟（ITU-T）的电信标准化部门推荐的标准。

[来源：ISO/IEC TR 29110-1-2:2024, 3.17]

3.429

基本值 base value

在“基本选择测试”中输入参数的取值，通常依据参数的代表值或典型值进行选择。也称为基本选择。

[来源：GB/T 38634.4-2020, 4.3]

3.430

基线 baseline

经正式核准的技术状态项的版本，与介质无关，是在技术状态项的生存周期中的特定时间节点确定并固化的。

[来源：GB/T 22032-2021, 4.1.10; GB/T 45802-2025]

3.431

基线配置管理 baseline configuration management

3. 正式评审和同意并作为其后开发工作依据的基线的建立。一些软件工作产品，如软件设计和代码，在预定点上具有建立的基线，并且对其实施严格的更改控制过程。

参见：2.127 条 基线管理（base management）。

[来源：GB/T 11457-2006, X.X]。

3.432

基线数据 baseline data

在过程开始时收集的数据，用于与随后收集的数据进行比较。

[来源：IEEE 7010-2020, 2.1]

3.433

基线设计 baseline design

对系统开发都关注的所有利益相关方达成一致的系统设计。

[来源：ISO/IEC/IEEE FDIS 24765, 3.412]

3.434

基线文件 baseline document

定义了已置于配置管理下的工作产品的系统或软件文档。

示例：设计规范、需求规格说明书、系统规范。

[来源：ISO/IEC/IEEE FDIS 24765, 3.413]

3.435

基线功能点计数 baseline function point count

在某个时间点上获取的应用程序功能点计数。可以基于它对之后的变更进行测量。

[来源: GB/T 42566-2023]

3.436

基线管理 baseline management

在配置管理中,通过技术和行政指导的应用来指定文件及其变更,以在配置项生存周期的特定时间点正式确定并建立基线。

[来源: ISO/IEC/IEEE FDIS 24765, 3.415]

3.437

评标基准价 baseline price for bid evaluation

在评标中设定为价格评分最高的价格。

[GB/T 36964-2018, X.X]。

3.438

基本假设 basic assumption

特定领域内保证案例所共享的一系列声明中的假设。

注: 对应于形式逻辑系统中的公理。

[来源: ISO/IEC/IEEE 15026-2:2022, 3.1.3]

3.439

基本工程对象 basic engineering object

需要分布式基础设施支持的工程对象。

[来源: ISO/IEC/IEEE FDIS 24765, 3.450]

3.440

基本流程 basic flow

描述其最常见实现的用例部分。

注1: 基本流程是在假设不存在错误或替代方案的情况下编写的。

同义词: 基本路径 (basic path), 理想场景 (happy day scenario)。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.441

基本流程 basic flow basic path happy day scenario

基本路径

快乐的一天场景

用例的一部分,用于描述其最常见的实现方式。

注: 基本流程是在假设不存在错误或备选方案的情况下编写的。

[来源: ISO/IEC/IEEE FDIS 24765, 3.418]

3.442

基本互通设施 basic interworking facility

与模拟计算操作的计算信号直接对应的一组服务原语。

[来源: ISO/IEC/IEEE FDIS 24765, 3.419]

3.443

基本成熟度等级 basic maturity level

组织过程成熟度等级的最低成就水平。

[来源: ISO/IEC 33001:2015, 3.3.3]

3.444

基本过程集 basic process set

一套流程, 确保达到基本成熟度等级。

[来源: ISO/IEC 33001:2015, 3.3.4]

3.445

基本概况 basic profile

针对由单个工作团队开发单个应用程序的 VSE 的配置文件。

[来源: ISO/IEC 29110-1-2:2024, 3.18]

3.446

基本符号 basic symbol

当例如过程或数据媒介等的确切性质或形式未知, 或无需描绘实际媒介时, 所使用的符号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.423]

3.447

估算的基础 basis of estimates

支持文件, 概述了用于建立项目估算的细节, 例如假设、约束条件、详细程度、范围和置信水平。

[项目管理知识体系指南 (PMBOK® 指南)——第七版]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.424]

3.448

基组 basis set

用于创建多重集的对象集合。

[来源: ISO/IEC 15909-1:2019, 2.1.4]]

3.449

批处理的 batch

与一种系统或操作模式有关, 在这种模式下, 输入信息会一次性全部收集和处理, 而不是在到达时进行处理, 并且一项作业一旦启动, 就会在没有额外输入或用户交互的情况下完成。

参见: 会话的、交互的、联机的、实时的。[来源: ISO/IEC/IEEE FDIS 24765, 3.427 batch]。

(原 11457 有对应翻译, 大致内容也差不多, 是否按推荐定义待后期确定)。

3.450

浴盆曲线 bathtub curve

在系统或部件中失效数作为时间函数的图形。

注: 由图表的通常形状得名: 失效的减量期 (早期失效期), 跟着是相对稳定期 (稳定失效期), 跟着是失效增量期 (磨损失效期)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.428 bathtub curve]

3.451

贝叶斯规则 Bayes' rule

将条件概率 $P(A | B)$ 与逆条件概率 $P(B | A)$ 相关联的统计公式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.429 Bayes' rule]

3.452

开始-结束块 begin-end block

由 begin 和 end 分隔符括起来的设计或程序语句序列。其特征是具有单一的入口和单一的出口。

[来源: GB/T 11457-2006, X.X]。

3.453

行为 behavior

实例规范的方面，由其可执行的状态改变操作决定。

[来源：ISO/IEC/IEEE FDIS 24765, 3.434]

3.454

行为模型 behavior model

从时间域上定义系统的行为，用于描述系统的运行过程中对外部事件触发和仿真逻辑时间的响应过程。

注：区别于面向对象方法对行为的描述，这里的行为模型主要描述了随着逻辑时间推进和外部仿真事件的触发，系统中各组件的调度和仿真运行过程。

[来源：GB/T 36457-2018, X.X]

3.455

行为兼容性 behavioral compatibility

两个对象的相同行为，使得一个对象可以根据一组标准替换另一个对象，而环境无法根据这组标准察觉到对象行为的差异。

[来源：ISO/IEC 10746-2:2009, 9.4]

3.456

行为特征 behavioral feature

模型元素的一种动态特征，例如一个操作或方法。

3.457

行为模型方面 behavioral model aspect

着重系统中各实例的行为（包括方法、协作和状态）一种模型面貌。

3.458

行为驱动开发 behaviour-driven development; BDD

一种开发方法，其中具有不同背景的团队（开发人员、测试人员和业务分析师）在开发相关功能之前共同描述预期功能的行为。

[来源：ISO/IEC TR 7052:2023, 3.5]。

3.459

基准 benchmark

能据以进行比对的参考点。

注：在 GB/T XXXXX（所有部分）中，比较的对象是 IT 项目绩效。

[来源：GB/T 41866.1-2022, X.X]

3.460

基准测试套件 benchmark suite

用于比较备选方案性能的测试集合。

[来源：ISO/IEC TR 29119-11:2020, 3.1.18]

3.461

基准度量 benchmarking

将各关注对象互相比对或分别与某一基准比较，以评价其特性的活动。

注：在 GB/T XXXXX（所有部分）中，关注对象是 IT 项目的绩效，而特性是 IT 项目的某一方面，例如生产率。

[来源：GB/T 41866.1-2022, X.X]

3.462

基准度量分析方 benchmarking analyst

实施基准度量活动的个人或组织。

[来源：GB/T 41866.1-2022, X.X]

3.463

基准度量经验库 benchmarking experience base

包含对信息产品和基准度量活动的评价以及在基准度量和分析期间获得经验的信息存储器。

注：ISO/IEC/IEEE 15939:2007 将“信息产品”定义为“针对信息需求的一项或多项指标及与其关联的解释”。例如模板、图形、统计算法和解释指南。

[来源：GB/T 41866.1-2022, X.X]

3.464

基准度量方法 benchmarking method

2. 对一种或多种属性，在与某一参考属性按所规定尺度比对过程中，描述其通用步骤的逻辑序列。

[来源：GB/T 41866.1-2022, X.X]

3.465

基准度量报告 benchmarking report

有关基准度量实例结果的文档。

注：此文档通常由各种格式（例如文本描述、数值、统计图表和表格）组成，并通过各种媒体（例如电子文档、电子数据集、打印的文档以及嵌入特定计算机软件的数据）进行交换。

[来源：GB/T 41866.1-2022, X.X]

3.466

基准度量数据仓库 benchmarking repository

指定用于基准度量、组织有序的持久性数据存储器。

[来源：GB/T 41866.1-2022, X.X]

3.467

基准对比用户 benchmarking user

利用基准度量结果的个人或组织。

[来源：GB/T 41866.1-2022, X.X;

~~[来源：ISO/IEC 29155-1:2017, 2.5]]~~

3.468

收益 benefit

系统或过程自愿或非自愿产生的积极结果。

注：收益对应于一个或多个潜在的期望值。

[来源：IEEE 7000:2021, 3.1]

3.469

效益成本分析 benefit cost analysis

在非营利决策分析中，根据人口净收益与赞助商净成本之比评价备选方案的可取性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.448]

3.470

Beta 测试 beta test

产品在小规模生产环境中进行的第二阶段测试。

注：通常在客户现场进行。

[来源: ISO/IEC/IEEE FDIS 24765, 3.451]

3.471

BFC 类型 BFC Type

定义的 BFC 类别。

示例: “外部输入”、“外部输出”和“逻辑事务”, 以及如“内部逻辑文件”等的数据存储。

[来源: ISO/IEC 14143-1:2007, 3.2]

3.472

偏差 bias

在机器学习 (ML) 中, 衡量 ML 模型提供的预测值与期望的公平预测值之间的距离。

[来源: ISO/IEC TR 29119-11:2020, 3.1.19]

3.473

投标价 bid price

在招标过程中, 各标人送交的承包价格。

[来源: GB/T 36964-2018, X.X]。

3.474

投标者 bidder

已提交建议, 并作为签订一个或多个产品设计、开发和(或)制造合同候选者的个人、合伙组织、公司或协会。[来源: GB/T 11457-2006, X.X]。

3.475

双向可追溯性 bidirectional traceability

两个或多个逻辑实体之间的关联, 这种关联可以双向辨别 (指向一个实体和从该实体指向另一个实体)。

参见: 需求可追溯性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.459]

3.476

总揽测试 big-bang testing

一类集成测试, 它把软件元素、硬件元素或两者组合在一起, 作为整个系统进行测试而不是分阶段测试。

3.477

物料清单 bill of materials; BOM

记录制造产品所需物理组件、子装配件和零部件的正式层级化表格。

[来源: ISO/IEC/IEEE 24765:2017, 3.378]

3.478

物料清单 (BOM) bill of materials (BOM); BOM

对制造产品所需的物理组件、子组件和部件进行文件记录的正式分层制表。

[来源: ISO/IEC/IEEE FDIS 24765, 3.461]。

3.479

功能清单 bill-of-features

产品线中成员产品的规范, 根据为该成员产品选择的功能目录中的特定功能呈现。

[来源: ISO/IEC 26580:2021, 3.1]

3.480

功能清单组合 bill-of-features portfolio

产品线中成员产品的规范，根据为该成员产品选择的功能目录中的特定功能呈现。

[来源：ISO/IEC 26580:2021, 3.2]

3.481

二元关联 binary association

两个类的之间的一种关联。N 元关联的一个特例。

[来源：GB/T 11457-2006, X.X]

3.482

二进制数字[位] binary digit; BIT

用于表示二进制数字系统，0 或 1 两个数字之一的数字。

参见：字节 (byte) 和 字 (word)。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.483

[捆]绑 bind

为标识符赋值。参见：动态绑定、静态绑定。

示例：在计算机程序中为参数赋值或将绝对地址赋给符号地址。

[来源：ISO/IEC/IEEE FDIS 24765, 3.465]

3.484

[捆]绑者 binder

通道中的工程对象，在交互的基本工程对象之间维持着分布式绑定

[来源：ISO/IEC 10746-3:2009, 8.1.10]

3.485

绑定 binding

根据领域可变性模型和应用可变性模型，对领域资产和应用资产的相应变体进行实例化的任务。

注：执行绑定是应用绑定定义，使用领域和应用可变性模型从领域和应用资产中生成新的应用。

[来源：GB/T 41865-2022, X.X]

3.486

绑定行为 binding behavior

在两个或多个接口之间建立行为，从而在它们的支持对象之间建立行为。

[来源：ISO/IEC/IEEE FDIS 24765, 3.468]

3.487

绑定端点标识符 binding endpoint identifier

在胶囊的命名上下文中，以实现交互为目的，基本工程对象用于选择其参与的绑定之一的标识符。

示例：内存地址（用于表示工程接口的数据结构）。

注：无论涉及的绑定是本地绑定还是分布式绑定，均可使用相同形式的绑定端点标识符。

[来源：ISO/IEC/IEEE FDIS 24765, 3.469]

3.488

绑定对象 binding object

支持一组与其他计算对象之间绑定的计算对象。

[来源：ISO/IEC/IEEE FDIS 24765, 3.470]

3.489

绑定前提条件 binding precondition

成功执行某个绑定行为所需的一组条件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.471]

3.490

绑定时间 binding time

可变性解析的时刻。

注: 绑定时间的选择独立于可变性建模。它是从需求到运行时所做决策的结果。对灵活性的需求和工具的支持允许后期绑定时间, 甚至使用可变绑定时间。

[来源: ISO/IEC 26555:2015, 3.3]

3.491

绑定时间决策 binding time decision

根据时间可变性和空间可变性之间的功能区别, 选择平台中定义的可变性。

[来源: ISO/IEC 26557:2016, 3.5]

3.492

可变性的绑定时间 binding time of variability

确定可变性值的阶段。

[来源: ISO/IEC 26553:2018, 3.7]

3.493

位 bit

在二进制记数系统中使用的数字 0 或 1。

[来源: ISO/IEC 2382:2015, Information technology --Vocabulary]

2. 内置测试。

[来源: ISO/IEC/IEEE 24748-8:2019, 3.2]

[来源: 同义词: BIT]

3.494

位操控 bit steering

微程序技术, 其中微指令中一个字段的含义取决于微指令中另一个字段的值。

参见: 剩余控制, 二级编码。

同义词: 立即控制 (immediate control)

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.495

位控制 bit steering immediate control

即时控制

一种微程序设计技术, 其中, 微指令中某个字段的含义, 取决于该微指令中另一字段的值。

参见: 二级编码 (two-level encoding)。

[来源: GB/T 11457, 2.720]

3.496

黑盒 black box

系统或组件, 其输入、输出和一般功能已知, 但其内容或实现未知或无关紧要。

[来源: ISO/IEC/IEEE 24765:2017, 3.390]

3.497

黑盒测试 black-box testing

忽略系统或组件的内部机制只集中于响应所选择的输入和执行条件产生的输出的一种测试。

3. 498

块【名】 block

组块【动】

〈名〉作为一个单元对待的一组连续的存储位置、计算机程序语句、记录、字、字符或位。

〈动〉组成一个组。

参见：块结构语言、定界符。

[来源：ISO/IEC/IEEE FDIS 24765, 3.490]

3. 499

块分配 block allocation

一种存储分配技术，其中程序或数据被划分为固定长度的块，称为页面；主存储器被划分为相同长度的块，称为页帧，页面根据需要在主辅存储器之间传输。

参见：连续分配、页面、页面中断、页帧、页面交换、页表、零页、分页程序、工作集。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3221]

3. 500

框图 block diagram

表示某一系统、计算机或设备的图，在这种图中，用适当标注的几何图形表示主要部件，以表示各部件的功能及其功能关系。

参见：盒图 (box diagram)、气泡图 (bubble chart)、流程图 (flowchart)、图 (graph)、输入-处理-输出图 (input-process-output chart)、结构图 (structure chart)。

[来源：ISO/IEC/IEEE 24765:2017, X.X]。

3. 501

阻塞性缺陷 blocking bug

阻碍在特定任务或功能上继续工作，直至其被解决的软件缺陷。

[来源：IEEE 982:2024, 3.1]

3. 502

分块因子 blocking factor

块中记录、字、字符或位的数量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.494]

3. 503

块结构语言 block-structured language

一种设计或编程语言，其中定义了语句序列，调用块，通常带有开始和结束定界符，且一个块中定义的变量或标签在该块之外无法识别。

示例：Ada、ALGOL、PL/I。

参见：结构化编程语言

[来源：ISO/IEC/IEEE FDIS 24765, 3.492]

3. 504

正文元数据 body metadata

HTML 文档主体中的元素，可为用户或管理员提供管理或导航功能。

[来源：ISO/IEC/IEEE 24765:2024, X.X]

3. 505

知识体系 body of knowledge; BOK

一般公认的、对理解特定主题至关重要的知识项或领域的集合。

[来源：ISO/IEC 24773-1:2019, 3.11]

3.506

布尔量 Boolean

一种枚举类型，其值只有“真”或“假”两种。

3.507

布尔表达式 Boolean expression

评价布尔量“真”或“假”的表达式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.501]

3.508

布尔签名 Boolean signature

其中一种类型是 Bool 的签名，对应于任何相关代数中的载体 Boolean，而其中一个常数是 true sub Bool，对应于代数中的值 true。

[来源：ISO/IEC/IEEE FDIS 24765, 3.502]

3.509

引导 boot

通过清除内存和重新加载操作系统来初始化计算机系统。

注：派生自 bootstrap。

[来源：ISO/IEC/IEEE FDIS 24765, 3.503]

3.510

引导模式 boot mode

计算机开机时程序运行的初始化模式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.504]

3.511

引导程序 bootstrap

一种永久驻留在计算机中或易于加载到计算机中的小型计算机程序，其执行会将更大的程序（如操作系统或其加载程序）加载到内存中。

[来源：ISO/IEC/IEEE FDIS 24765, 3.505]

3.512

引导加载程序 bootstrap loader

用于加载引导程序的小型计算机程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.506]

3.513

自底向上 bottom-up

用于说明一种活动，这种活动从层次结构的最低层组件开始并逐步向更高层推进。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.514

自下而上设计 bottom-up design

将系统的低级部分组合成整体设计的设计方法。2. 设计系统的过程，通过识别低级组件，分别设计每个组件，然后设计一个结构将低级组件集成到越来越大的子系统中，直到设计完成。

3.515

边界 boundary

在研软件与其用户之间的概念上的界面。

注：从外部用户视角观察，使用 FiSMA 1.1 方法测量的软件边界在概念上分隔了软件和其运行环境。软件边界为测量分析人员提供了可靠的分隔标识，用于明确区分哪部分是受测软件包含的内容，哪部分是受测软件的运行环境。

[来源：ISO/IEC 14143-1:2007, 3.3]

3.516

边界箭头 boundary arrow

一端（源或使用）未连接到图中任何框的箭头。

参见：内部箭头。

[来源：ISO/IEC/IEEE FDIS 24765, 3.510]

3.517

边界值 boundary value

对应于为系统或组件规定的最小或最大的输入、内部、输出值的数据值。

参见：压力测试。

[来源：ISO/IEC/IEEE FDIS 24765, 3.511]

3.518

边界值分析 boundary value analysis

基于规范的测试设计技术，通过测试等价类划分的边界来实现。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.12]

3.519

方框 box

包含方框名称、方框编号和可能的方框详细信息参考，并在图中表示某个函数的矩形。

[来源：ISO/IEC/IEEE FDIS 24765, 3.513]

3.520

方框细节参考 box detail reference

包含方框编号的正方形外框，表示方框由子图分解或细化。

[来源：ISO/IEC/IEEE FDIS 24765, 3.514]

3.521

框图 box diagram

一种控制流程图，它由矩形构成，又可分解以表示顺序步，if-then-else 条件，重复和 case 条件。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.522

方框图 box diagram

Chapin 图 Chapin chart

Nassi-Shneiderman 图 Nassi-Shneiderman chart

控制流程图由一个矩形组成，该矩形被细分以显示顺序步骤、if-then-else 条件、重复和 case 条件。

参见：框图、气泡图、流程图、图表、输入-过程-输出图、程序结构图、结构图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.515]

3.523

方框 ICOM 代码 box ICOM code

ICOM 代码，它将隧道边界箭头映射到附着在某个祖先框上的箭头。

[来源：ISO/IEC/IEEE FDIS 24765, 3.516]

3. 524

方框名 box name

置于方框内的，命名建模函数的动词或动词短语。参见：函数名。

注：方框将其所表示函数的函数名作为方框名。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 517]

3. 525

方框编号 box number

单个数字（0、2、...、9），置于方框右下角，以在图表中唯一标识该方框。

注：在 A-0 和 A-1 上下文图中，唯一可以编号为 0 的框是代表 A0 函数的框。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 518]

3. 526

头脑风暴 brainstorming

一种数据收集和创造性技术，可以通过团队成员或主题专家群体来识别风险、想法或问题解决方案。

[来源：ISO/IEC/IEEE 24765h:2019]

3. 527

分支 branch

一种计算机程序结构，在此结构中，可选择两个或多个程序语句的可替换的集之一执行。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3. 528

分支条件组合测试 branch condition combination testing

基于结构的测试用例设计技术，通过对决策中条件的布尔值组合进行测试来实现。

[来源：ISO/IEC/IEEE 29119-4:2021, 3. 4]

3. 529

分支条件测试 branch condition testing

基于结构的测试用例设计技术，通过对决策条件及其结果进行布尔值取值来执行测试。

[来源：ISO/IEC/IEEE 29119-4:2021, 3. 5]

3. 530

分支测试 branch testing

基于测试项控制流中分支执行的结构化测试用例设计技术。

参见：路径测试、语句测试。

[来源：ISO/IEC/IEEE 29119-4: 2021, 3. 6]

3. 531

分支 branching

一种开发方法，其中复制一组部件，以便可以并行修改这些部件，并可选择在后续时间进行同步。

[来源：ISO/IEC/IEEE 26531:2023, 3. 1. 2]

3. 532

分支点 branchpoint

计算机程序中选择两个或多个程序语句的可替换的集之一的执行点。

[来源：11457, 定义 2. 159]

3. 533

面包屑路径 breadcrumb trail

一种导航辅助工具，显示从主页或其他页面到当前页面的一系列链接。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.3]

3.534

目标优化分析 break-even analysis

分析两个或更多个目标函数，以找出它们具有相同值的位置（如果有的话）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.527]

3.535

断点 breakpoint

在计算机程序中，可暂停执行以手动或自动监控程序性能或结果的点。

注：类型包括代码断点（code breakpoint）、数据断点（data breakpoint）、动态断点（dynamic breakpoint）、跋断点（epilog breakpoint）、可编程断点（programmable breakpoint）、序断点（prolog breakpoint）和静态断点（static breakpoint）。当程序中的一点和在此点引起执行暂停的事件都设置时，认为设置了一断点；当程序执行暂停时，认为断点启动。

3.536

棕地系统工程 brownfield system engineering

在现有或遗留系统或系统元素存在的情况下，开发未来系统或系统元素。

参见：绿地系统工程。

注：棕地方法通常用于扩展、改进或替换正在使用的系统，或复用不受预期变更影响的系统元素。新系统架构需考虑现有系统元素和功能，这可能会对整体系统定义施加约束。

[来源：INCONE 系统工程手册，第五版.]

3.537

浏览器 browser

允许用户检索和阅读超文本、查看超文本节点（网页）内容、在网页间导航以及与内容交互（如更改显示内容的视觉外观）的应用程序。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.4]

3.538

气泡图 bubble chart

一种数据流程、数据结构或其他图，其中，实体用圆圈（circle）表示，关系用圆圈间的连接表示。

3.539

同伴检查 buddy check

由作者的同事独立进行的非正式评审。

参见：同行评审（peer review）。

[来源：ISO/IEC 20246:2017, 3.3]

3.540

预算 budget

项目或任何工作分解结构组成部分或任何进度活动的批准估算。

参见：估算。

注：通常也用于指代工作量，以及（或代替）资金。

[来源：ISO/IEC/IEEE FDIS 24765, 3.534]

3.541

完工预算 budget at completion; BAC

为待执行工作所建立的所有预算的总和。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.542

完工预算 **budget at completion (BAC)**

为待执行工作所确立的所有预算的总和。

[来源: ISO/IEC/IEEE FDIS 24765, 3.535]

3.543

预算价 **budget price**

项目立项时批复的预算额度。

[来源: GB/T 36964-2018, 3.24]

3.544

预算编制 **budgeting**

根据项目项目成本估算的结果确定预期项目费用使用的过程。

[来源: GB/T 36964-2018, 3.23]

3.545

缓冲区 **buffer**

用于临时存储数据的设备、程序或存储区域,以补偿数据流率、事件发生时间或数据传输或使用中涉及的设备或过程所能处理的数据量的差异。

[来源: ISO/IEC/IEEE FDIS 24765, 3.536]

3.546

缺陷 **bug**

系统或系统表示中的缺陷,若执行或激活则可能导致错误。

注:若遇到故障,则可能导致失效。当规范不正确时,可能会出现故障。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1796]

3.547

缺陷植入 **bug seeding**

故意地把已知故障加至已经有故障的计算机程序中,以监控检测和删除的速度并估计程序中余下的故障数。

参见: 原生错误。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1640]

3.548

构建 **build**

从源版本或基线生成(归档)可执行且可测试系统的过程。

注:在软件领域,此指处理源文件以生成目标文件;在硬件领域,此指组装物理实体。构建过程需按正确顺序编译并链接各版本。构建工具可集成至配置管理工具中。

[来源: IEEE 828-2012, 2.1]

3.549

构建过程 **build process**

将项目代码库转化为可用应用程序的过程。

[来源: ISO/IEC 19506:2012, 4]

3.550

基本构建块 **building block**

较高一级程序或模块使用的一个单元或模块。

3. 551

内置随机存取存储器 built-in random access memory; Built-in RAM

嵌入微控制器单元（MCU）芯片的随机存取存储器。

[来源：ISO/IEC/IEEE 24765e:2015；ISO/IEC/IEEE FDIS 24765, 3.538]

[来源：同义词：built-in RAM]

3. 552

内置随机存取存储器（RAM） built-in random access memory (RAM)

微控制器单元（MCU）芯片中嵌入的随机存取存储器（RAM）。

近义词：内置 RAM。

[来源：IEEE Computer Society, Software and Systems Engineering Vocabulary, p51]

3. 553

内置只读存储器 built-in read only memory built-in ROM

微控制器单元（MCU）芯片中嵌入的只读存储器（ROM）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.539]

3. 554

燃尽图 burndown

一个指标，用于表示已完成的工作量，以及对剩余待完成工作或完成产品开发迭代周期所需剩余工作量的估计。

参见：燃起图。

注：工作量的衡量包括产品开发迭代过程中为交付故事点、故事、特性、功能、功能点、用户故事、用例或需求而完成的所有工作。

[来源：ISO/IEC/IEEE FDIS 24765, 3.542]

3. 555

燃尽图 burndown chart, burn chart

表示项目剩余工作量的图表。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.6]

3. 556

燃尽率 burndown rate

每个工作单元（周或迭代）完成的软件故事点、特性、功能、用户故事、用例或需求的数量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.544]

3. 557

老化期 burn-in period

在系统或组件的生存周期中，随着问题的检测和修复，硬件失效发生率逐渐降低的一段时期。

参见：定失效率周期（constant-failure period），磨损失效周期（wearout-failure period），浴盆曲线（bathtub curve）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1512]

3. 558

燃起图 burnup

一个指标，用于表示已完成的故事点、功能、函数、用户故事、用例或需求的数量，以及完成产品开发迭代周期剩余的或剩余所需的工作量。工作量的衡量包括产品开发迭代过程中为交付故事点、故事、特性、功能、功能点、用户故事、用例或需求而完成的所有工作。

参见：燃尽图。

[来源: ISO/IEC/IEEE FDIS 24765, 3.545]

3.559

链路 bus

计算机或系统中的数据通信路径。

[来源: ISO/IEC/IEEE FDIS 24765, 3.546]

3.560

商业案例 business case

拟议项目的价值主张,可能包括财务及非财务收益。

[来源: ISO/IEC/IEEE FDIS 24765, 3.547]

3.561

业务关键产品 business critical product

对业务或组织运营至关重要的产品,其持续故障将对业务造成重大影响,例如收入损失或声誉损失。

[来源: ISO/IEC 29110-1-2:2024, 3.21]

3.562

业务影响分析 business impact analysis

分析中断对组织随时间推移所产生影响的过程。

注:其结果是对业务连续性需求的陈述和论证证明。

[来源: ISO/IEC TR 7052:2023, 3.8]

3.563

业务信息管理 business information management

负责所有旨在支持最终用户使用应用程序,并充当 IT 组织客户的任务与活动的领域。

注:业务信息管理是指在维护信息供应和信息系统功能的过程中,作为应用管理和 IT 基础设施管理组织的客户组织或客户方的业务。它是信息供应的需求方。信息系统可包含非自动化元素,例如表单与用户指南,这类元素通常由业务信息管理组织负责维护。

[来源: ISO/IEC 16350-2015, 4.8]

3.564

商业模式画布 business model canvas

描述价值主张、基础设施、客户和财务状况的单页可视化摘要。

注:常用于精益创业场景。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.565

业务目标 business objective

由高级管理层设计的战略,旨在帮助确保组织的持续存在,并提高其盈利能力、市场份额以及影响组织成功的其他因素。

[来源: ISO/IEC/IEEE FDIS 24765, 3.552]

3.566

业务流程 business process

为实现组织既定目标而执行的部分有序企业活动集,旨在达成某种预期的最终结果。

[来源: ISO/IEC/IEEE FDIS 24765, 3.553]

3.567

业务需求 Business requirements

以业务术语描述需要交付或完成的内容的需求。

[来源：ISO/IEC/IEEE 41062:2024, 3.1.5]

3.568

业务需求规范 Business Requirements Specification

业务或使命需求（业务或使命问题或机遇的定义、各种概念以及各种解决方案的要求条件）及其与外部环境的关系的构成集合。

[来源：GB/T 45802-2025, 3.14]

3.569

业务价值 business value

企业制定的战略优先级事项，涉及收入、成本、风险、安全、隐私、伦理与合规等方面。

[来源：IEEE 2675-2021, 3.1]

3.570

占用 busy

关于系统或组件处于运行、服务和使用中。

参见：关机（down）、空闲（idle）和开机（up）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.557]

3.571

占用时段 busy time

在计算机性能工程中，系统或组件处于运行、服务和使用期间的时段。

参见：停机时间、空闲时间、设置时间、开机时间。

注：运行中、使用中和使用中。

[来源：ISO/IEC/IEEE FDIS 24765, 3.558]

3.572

字节 byte

由若干二进制位组成的串，作为一个单元处理，通常代表一个字符或字符的一部分。

注：通常指由 8 个二进制位组成的集合。

[来源：ISO/IEC 2382:2015, X.X]

3.573

缓存 cache

用于处理器内的数据存储的，具有很高运行速度的随机存取存储器（RAM）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.561]

3.574

节奏 cadence

贯穿项目的开展活动的节奏。

[来源：ISO/IEC/IEEE FDIS 24765, 3.563]

3.575

日历单位 calendar unit

在项目进度安排中所使用的最小时间单位。

注：日历单位通常以小时、天或周为单位，但也可以季度、月、班次或甚至以分钟为单位。

[来源：ISO/IEC/IEEE FDIS 24765, 3.567]

3.576

调用 call

使控制从一个软件模块转移至另一个软件模块，且通常可指定需在模块间传递的参数的计算机指

令。

注：调用可涉及服务、信息或咨询请求；中断或错误报告（事件）；变更请求；任务分配，例如启动计划外生产运行的指令，以及投诉。

[来源：ISO/IEC 16350-2015, 4.9]

3.577

按名调用 call by name

一种传递参数的方法，其中调用模块向被调用模块提供一个表示待传递参数的符号表达式，服务例程对该表达式进行评价，并将结果值提供给被调用模块。参见：引址调用、按值调用。

注：由于每次评价表达式时，其对应的形式参数都会在被调用模块中使用，因此在调用模块的执行过程中，参数的值可发生变化。

[来源：ISO/IEC/IEEE FDIS 24765, 3.569]

3.578

引址调用 call by reference

一种参数传递方法：调用模块向被调用模块提供待传递参数的地址。参见：按名调用、按值调用。

注：采用该方法时，被调用模块能够修改调用模块所存储的参数值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.570]

3.579

引址调用 call by reference

按址调用 call by address

按单元调用 call by location

一种传递参数的方法，其中调用模块向被调用模块提供待传递参数的地址。

参见：按名调用（call by name）、按值调用（call by value）。

注：使用此方法，被调用模块具有变更调用模块所存储参数的值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.570]

3.580

按值调用 call by value

一种传递参数的方法，其中调用模块向被调用模块提供待传递参数的实际值。

参见：按名调用、引址调用。

注：使用这种方法，被调用模块无法变更调用模块存储的参数值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.571]

3.581

调用图 call graph

用于标识系统或计算机程序中的模块，并展示模块间调用关系的图。

参见：结构图、控制流图、数据流图、数据结构图、状态图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.572]

3.582

调用图 call graph ,

调用树 call tree

排列图 tier chart

标识在系调或计算机程序中的模块并显示它们的调用关系的图。

参见：结构图、控制流图、数据流图、数据结构图和状态图。

注：结果不需要与在结构图中显示的相同。

[来源：ISO/IEC/IEEE FDIS 24765, 3.572]

3.583

调用表 **call list**

在软件模块调用中所用的自变量的有序列表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.573]

3.584

调用参考 **call reference**

调用箭头所附的页面参考。

[来源：ISO/IEC/IEEE FDIS 24765, 3.574]

3.585

调用踪迹 **call trace**

子例程 (subroutine trace) 踪迹。

3.586

调用图表 **called diagram**

由调用框调用并由调用箭头所附的页面参考标识的分解图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.575]

3.587

调用框 **calling box**

通过分解图进行详细说明的框，分解图不是该框的子图。

注：调用箭头附在调用框的底部。

[来源：ISO/IEC/IEEE FDIS 24765, 3.576]

3.588

调用顺序 **calling sequence**

计算机指令序列，以及可能执行调用另一个模块所需的数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3.577]

3.589

候选功能规模测量方法 **candidate FSM method**

为符合评估要求而提交的已成文的软件规模测量方法。

[来源：ISO/IEC 14143- 2:2011, 3.1]

3.590

候选键 **candidate key**

实体的属性或属性组合，没有两个实例就其值达成一致。

注：[key style]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.577]

3.591

能力维 **capability dimension**

一组过程属性，它包括过程参考模型的能力方面和过程能力。

注：将属性条理化成能力等级，包括一个有序的过程能力度量。

[来源：11457, 2.182]

3.592

能力成熟度模型 **capability maturity model**

包含一个或多个学科有效过程的基本元素的模型，描述了从临时的、不成熟的过程到具有质量和效

率提升的规范的、成熟的过程的演化改进路径。

[来源：ISO/IEC/IEEE FDIS 24765, 3.584]

3.593

能力需求 capability requirement

规定系统在指定条件下所需满足的状态或行为的需求。

参见：可靠性需求（dependability requirement）。

[来源：IEEE 982:2024,, 3.1]

3.594

交付过程 capable process

能够满足特定产品质量、服务质量和过程性能目标的过程。

参见：稳定过程、标准过程、统计管理过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.586]

3.595

容量 capacity

产品满足产品参数最大限值需求的能力。

[来源：ISO/IEC 25010:2023, 3.2.3]

3.596

容量受限位置 capacity place

最大容量不超过规定值的特殊位置。

[来源：ISO/IEC 15909-3:2021, 3.2]

3.597

容量测试 capacity testing

性能效率测试的一种类型。用于评价增加负载（用户、交易、数据存储等）时，测试项能力水平和所需维持性能的折中点。

[来源：GB/T 38634.1-2020, 3.5]

3.598

资本支出 capital expenditure

企业为购置有形基础设施或设施项（如家具、电脑等）所进行的支出。

注：不包括购买消耗品或纳入待售成品的项。

[来源：ISO/IEC/IEEE FDIS 24765, 3.590]

3.599

胶囊 capsule

为封装处理和存储而将工程对象形成成为单个单元的配置。

示例：虚拟机、进程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.591]

3.600

容器 capsule manager

管理胶囊中的工程对象的工程对象。

[来源：ISO/IEC/IEEE FDIS 24765, 3.592]

3.601

势 cardinality

对通过关系与主体实体相关的实体实例数量的约束。

参见：势约束。

注：对于关系的每个方向，势都可被约束。

[来源：ISO/IEC/IEEE FDIS 24765, 3.595]

3.602

势约束 cardinality constraint

限制关系中可以相互关联的实例数量的约束。

参见：势。

[来源：ISO/IEC/IEEE FDIS 24765, 3.596]

3.603

载体 carrier

多分类代数集。

[来源：ISO/IEC/IEEE FDIS 24765, 3.597]

3.604

用例 case

对系统的行为需求及其与用户交互的描述。

[来源：ISO/IEC/IEEE 26515:2018, 3.15, 有修改]

3.605

情况语句 case multiple exclusive selective construct

单一入口、单一出口的多路分支，其定义控制表达式，对控制表达式的每个值规定要执行的处理，并在所有情况下返回控制至立即跟随在整体结构之后的语句。

参见：转至、跳转、“若-则-否则”语句、多路包含选择结构（multiple inclusive selective construct）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.598]

3.606

转换 cast

将一种类型的对象视为另一种类型的对象。

参见：强制（转换）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.601]

3.607

灾难性失效 catastrophic failure

关键软件的失效。

[来源：GB/T 11457—2006, 2.187]

3.608

分类 categorization

将目标系统划分到某一类别中的特定方式。

[来源：ISO/IEC TR 12182:2015, 3.5]

3.609

结构视图 categorization scheme

对软件相关视图和类别进行的有序组合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.604]

3.610

分类空间 categorization space

具有一个或多个分类轴作为各自维度的系统与软件通用集合，借此可表达涉众对分类的关注点。

[来源：ISO/IEC TR 12182:2015, 3.6]

3.611

类别实体 category entity

其实例代表另一个实体（通用实体）的子类型或子分类的实体。

参见：子类（subclass）、子类型（subtype）。

[来源：IEEE Computer Society, Software and Systems Engineering Vocabulary, p58]。

3.612

原因分析 causal analysis

缺陷分析，以确定其原因。

[来源：ISO/IEC/IEEE FDIS 24765, 3.608]

3.613

因果图 cause and effect diagram

一种有助于将不良影响追溯到其根本原因的分解技术可视化表达。

[来源：ISO/IEC/IEEE FDIS 24765, 3.609]。

3.614

因果图法 cause-effect graphing

基于因果图中决策规则执行的基于规格说明的测试用例设计技术。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.9]

3.615

警告 caution

一种危险情况，若不加以避免，可能导致轻微或中度伤害。

参见：危险、警告、注意。

[来源：ISO/IEC/IEEE FDIS 24765, 3.612]

3.616

CDIF 明文编码 CDIF clear text encoding

CDIF 传输文件的明文文件编码。

[来源：ISO/IEC 15474-1:2002, 4.2]。

3.617

CDIF 导出工具 CDIF exporter

创建 CDIF 传输文件的工具。

[来源：ISO/IEC 15474-1:2002, 4.2]。

3.618

CDIF 系列标准 CDIF family of standards

一组标准，当一起使用时，为建模工具之间的信息交换提供标准定义。

[来源：ISO/IEC 15474-1:2002, 4.2]。

3.619

CDIF 图形符号 CDIF graphical notation

一组管理图表中 CDIF 建模概念表示的规则。

[来源：ISO/IEC 15474-1:2002, 4.2]。

3.620

CDIF 标识符 CDIF identifier

传输模型部分中唯一标识对象的属性。

[来源：ISO/IEC 15474-1:2002，4.2]。

3.621

CDIF 导入工具 **CDIF importer**

读取 CDIF 传输文件并使用它来创建或修改模型的工具。

[来源：ISO/IEC 15474-1:2002，4.2]

3.622

CDIF 元标识符 **CDIF metaidentifier**

在传输的元模型部分唯一标识元对象的元属性。

[来源：ISO/IEC 15474-1:2002，4.2]。

3.623

CDIF 元模型 **CDIF meta-model**

用于定义元模型的一组概念和符号的描述。

注：具体来说，CDIF 元模型元模型定义了一个实体-关系-属性模型，用于构建和定义元模型和 CDIF 元元模型本身。

[来源：ISO/IEC 15474-1:2002，4.2]

3.624

CDIF 语义元模型 **CDIF semantic metamodel**

用于定义模型的一组概念和符号的描述。

注：CDIF 语义元模型定义了一个实体-关系-属性模型，用于构建和定义系统开发中使用的模型。

[来源：ISO/IEC 15474-1:2002，4.2]

3.625

CDIF 传输 **CDIF transfer**

特定语法、该语法的特定编码和元模型的组合。

注：即传输格式和内容的完整定义。

[来源：ISO/IEC 15474-1:2002，4.2]

3.626

CDIF 传输文件 **CDIF transfer file**

传输符合 ISO/IEC 15475 的文件。

[来源：ISO/IEC 15474-1:2002，4.1]

3.627

CDIF 传输格式 **CDIF transfer format**

特定语法和该语法的特定编码的组合，它们共同提供传输格式的完整定义。

[来源：ISO/IEC 15474-1:2002，4.2]

3.628

CDIF 传输语法和编码 **CDIF transfer syntax and encoding**

CDIF 支持的标准车辆格式。

注：SYNTAX.1 和 ENCODING.1 的组合形成初始 CDIF 传输语法和编码。

[来源：ISO/IEC 15474-1:2002，4.2]

3.629

中央处理器 **central processing unit; CPU**

由一个或多个处理器及其内部存储器组成的功能单元。

[来源：ISO/IEC 2382:2015, X. X]

3. 630

中央处理器 (CPU) **central processing unit (CPU); CPU**

由一个或多个处理器及其内部存储器组成的功能单元。

[来源：ISO/IEC 2382:2015, X. X]

3. 631

动作链 / 任务链 **chain**

以定义的顺序提交给 SUT 的一项或多项任务

2. 活动中的动作顺序，其中，对于每对相邻的动作，第一个动作的发生对于第二个动作的发生是必要的。

[来源：ISO/IEC 10746-2:2009, 13.1.1]。

3. 632

链式 **chain type**

由任务类型序列定义的链分类。

注：模拟用户仅提交链指定的链类型到 SUT。

[来源：ISO/IEC 14756:1999, 4.4]

3. 633

变更 **change**

对现有应用程序的修改，包括添加、更改和删除。

参见：增强。

[来源：ISO/IEC 20968:2002, 10]

3. 634

变更控制 **change control**

识别、记录、批准或拒绝与项目相关的文件、可交付成果或基线的修改的过程。

参见：配置控制、版本控制。

[来源：项目管理知识体系指南 (PMBOK® 指南) 第五版]

3. 635

变更控制委员会 **change control board (CCB); CCB**

一个正式特许的小组，负责审查、评估、批准、延迟或拒绝项目的变更，并记录和传达这些决定。

参见：配置控制板。

[来源：项目管理知识体系指南 (PMBOK® 指南) 第五版]

3. 636

变更控制计划 **change control plan**

项目管理计划的组成部分，用于建立变更控制委员会，记录其权限范围，并描述变更控制系统的实施方式。

[来源：项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3. 637

变更控制程序 **change control procedure**

为识别、记录、审查和授权更改正在开发的软件或文档产品而采取的行动。

注：该程序确保变更的有效性得到确认，对其他项目的影响得到检查，并且那些与开发有关的人被告知变更。

[来源：ISO/IEC 26514:2008, 4.8]

3. 638

变更转储 change dump

对自某一指定时间或事件起内容已发生变化的那些存储位置进行选择性的转储。参见动态转储、内存转储、事后转储、选择性转储、快照转储、静态转储。

[来源: ISO/IEC/IEEE FDIS 24765, 3.664]

3. 639

变更管理 change management

明智运用相关方法, 对产品或服务实施一项变更或提出一项变更建议。

参见: 配置管理。

[来源: ISO/IEC/IEEE 24765:2017, X. X]

3. 640

变更包 change package

通过已更改并获批准, 将被移交至生产环境的对象集合。

[来源: ISO/IEC 16350-2015, 4.10]

3. 641

变更项目功能点计数 change project function point count

一种计量工作产出量的计数方式, 该工作产出来源于对现有应用系统的修改, 包括在项目完成时所交付的用户功能的新增、变更或删除。

[来源: ISO/IEC 20968:2002, 10]

3. 642

变更记录 change record

包含受影响的配置项, 以及已批准的变更对这些配置项产生何种影响的详细信息的记录。

[来源: ISO/IEC/IEEE FDIS 24765, 3.669]

3. 643

变更请求 change request

提交修改配置项申请的正式程序。

参见: 修改请求、变更申请。

[来源: ISO/IEC TR 18018:2010, 3.5]

3. 644

变更集 change set

可因发布而发生变更的对象集合。

[来源: ISO/IEC 16350-2015, 4.11]

3. 645

转换系统 changeover system

用于促进从运行系统向其继任系统过渡的临时信息处理系统。

[来源: ISO/IEC 2382:2015, X. X]

3. 646

过渡系统 change-over system

一种临时的信息处理系统, 它用于为由某一运转系统到其后继系统的转换提供便利。

[来源: GB/T 11457-2006, 2.197]

3. 647

信道容量 channel capacity

在给定信道上单位时间内能够传输的最大信息量；通常以比特每秒（bit/s）或波特（baud）为单位计量。

参见：存储容量 / 内存容量、存储空间容量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.674]

3.648

查平图 Chapin chart

一种控制流程图，它由矩形构成，又可分解以表示顺序步，if-then-else 条件，重复和 case 条件。

[来源：GB/T 11457-2006, 2.199]

3.649

字符 character

用于表示信息的字母、数字或其他符号。

[来源：ISO/IEC 2382:2015, X.X]

3.650

字符集 character set

编码中用来表示终结符的字符集合。

注：所使用的字符集，对 CDIF 传输中文本和字符串元属性的编码具有重要影响。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.651

字符类型 character type

数据类型，其成员可以采用指定字符的值，并且可以通过字符运算符进行操作，例如连接、枚举类型、整数类型、逻辑类型、实型。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.652

特性 characteristic

见：2.392 条 数据特性(data characteristic)、2.1475 条 软件特性(software characteristic)。

参见：数据特性、软件特性。

[来源：11457-2006, 2.202]

[来源：可能对生产力产生重大影响的软件开发的内在因素。]

3.653

特性实体 characteristic entity

为另一个元对象提供附加属性的元实体。

注：特征实体的其他通用名称是属性实体和依赖实体。特征元实体的每个实例在逻辑上仅与另一个元对象的一个实例相关，因此进口商可以将特征元实体的元属性与“拥有”元对象的元属性合并，其中拥有元对象是与特征元实体以 1:1 的基数相关的对象。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.654

用户功能需求的特性 characteristic of FUR

用户功能需求的一个显著属性，对于识别特定功能用户需求集合所属的功能领域非常重要。

[来源：ISO/IEC TR 14143-5:2004, 3.1]

3.655

检查单分析 checklist analysis

一种借助清单来系统地审查资料，以验证其准确性与完整性的技术。

[来源：《项目管理知识体系指南》PMBOK® 指南 第五版]

3. 656

基于检查单的评审 checklist-based reviewing

基于问题列表或要求属性引导的评审技术。

[来源：ISO/IEC 20246:2017, 3.4]

3. 657

检查, 检出 checkout

校验

在运行环境或运维支持环境中开展的测试, 用于确保软件产品在安装后能够按需求正常运行。

[来源：ISO/IEC/IEEE FDIS 24765, 3.684]

3. 658

检查点 checkpoint

1 在计算机程序中检查或记录程序状态、状态或结果的点。

2 从工程对象的状态和结构派生的对象模板, 可用于实例化另一个与原始对象在检查点时的状态一致的工程对象。

[来源：ISO/IEC 10746-3:2009, 8.1.20]

3. 659

检查表 checksheets

可用于收集数据时作为检查表的计数表。

[来源：《项目管理知识体系指南 (PMBOK® 指南)》— 第七版]

3. 660

主程序员 chief programmer

首席程序员团队负责人或高级程序员, 其职责包括完成分配给团队的软件关键部分、协调团队活动、评审其他团队成员的工作, 并对正在开发的软件具备全面的技术理解。参见备份程序员、首席程序员团队。

[来源：ISO/IEC/IEEE FDIS 24765, 3.687]

3. 661

主程序员组 chief programmer team

由主程序员、后备程序员、秘书/资料员, 以及根据需要增设的程序员和专家组成的软件开发团队, 该团队采用旨在加强团队沟通并最优利用每位成员技能的流程。参见: 后备程序员、主程序员、无私编程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.688]

3. 662

子 child

在某一泛化联系中, 对另一元素, 即父的专化。

参见: 子类和子类型。与 2.1107 条, 父 (parent) 相对。

[来源: 11457-2006, 2.207]

3. 663

子框 child box

位于子图中的图框。

[来源：ISO/IEC/IEEE FDIS 24765, 3.689]

3.664

子图 child diagram

通过恰好一个子/父关系与特定框相关的分解图。

[来源: ISO/IEC/IEEE FDIS 24765, 3.690]

3.665

子实体 child entity

特定关系中的实体,其实例可以与另一个实体(父实体)的零个或一个实例相关。

[来源: ISO/IEC/IEEE FDIS 24765, 3.691]

3.666

子标签 child tag

与另一个标签有附属关系的标签。

[来源: IEEE Computer Society—Software and Systems

[来源: Engineering Vocabulary]]

3.667

类对象二元体 clabject

同时既是类又是对象的双重实体。

注: 由于其双重特性,类对象兼具类切面对象切面,可随时以任意一种身份运行。幂型的实例通常被视为类对象,因为它们既是作为幂型的类型实例的对象,同时也是作为分区类型子类的类。

[来源: ISO/IEC 24744:2014, 3.13]

3.668

安全断言 claim

关于明确定义的属性(称为声明属性)的取值限制,以及在声明适用期限内、规定条件下该属性的取值不确定性落在这些限制范围内的限制的真假陈述。

注: 声明可潜在包含以下内容: 目标系统的属性; 与声明相关的属性取值限制,例如其范围; 满足该限制的属性值的不确定性限制; 声明适用期限的限制; 与期限相关的不确定性; 与声明相关的条件限制; 以及与条件相关的不确定性。

[来源: ISO/IEC/IEEE 15026-1:2019, 3.1.2]

3.669

类[别] class

面向对象程序中包含功能和数据组合的静态编程实体 3. 在<X>中,满足特定类型的所有<X>的集合。参见: 类型。

注: 类用于表示“其知识或行为相关的实体”的概念。

[来源: ISO/IEC 10746-2:2009, 9.10]

3.670

类图 class diagram

一种图,它显示了由一组声明性(静态)模型元素(如类、类型等)以及它们的内容和彼此之间的关系所形成的集合。

[来源: 11457-2006, 2.212]

3.671

类层次结构 class hierarchy

类的层次结构,其中子类是其超类的特化。

注: 类从其超类继承属性和关系,并且可以定义自己的附加属性和关系。

[来源: ISO/IEC/IEEE FDIS 24765, 3.701]

3.672

类 **class**

用于将数据和操作的方法封装在一起,表示问题域的抽象概念。

[来源:GB/T 45803-2025]

3.673

分类 **classification**

机器学习功能,用于预测给定输入的输出类别。

[来源:ISO/IEC TR 29119-11:2020, 3.1.20]

3.674

分类轴 **classification axis**

根据特定视角对系统和软件进行分类的映射总范围。

[来源: ISO/IEC TR 12182:2015, 3.7]

3.675

分类树 **classification tree**

程序输入数据的层次树模型,其中输入由不同的分类(相关测试方面)和类(输入值)表示。

[来源: ISO/IEC/IEEE 29119-4:2021, 3.10]

3.676

分类树法 **classification tree method**

基于分类树中类的执行的基于规约的测试用例设计技术。

[来源: ISO/IEC/IEEE 29119-4:2021, 3.11]

3.677

分类器 **classifier**

用于分类任务的机器学习模型。

[来源: ISO/IEC TR 29119-11:2020, 3.1.20]

3.678

类级属性 **class-level attribute**

从类本身映射到值类的实例。

[来源: ISO/IEC/IEEE FDIS 24765, 3.702]

3.679

类级操作 **class-level operation**

从类本身与各输入参数类型实例的(笛卡儿)积,映射到其他(输出)参数类型实例的(笛卡儿)积。

[来源: ISO/IEC/IEEE FDIS 24765, 3.703]

3.680

类级职责 **class-level responsibility**

代表类整体的某些方面知识、行为或规则的责任,参见:实例级责任。

示例:注册选民总数是注册选民类的类级属性;整个类只能有一个注册选民总数的值。

[来源:ISO/IEC/IEEE FDIS 24765, 3.1.26]。

3.681

清除 **clear**

将变量、寄存器或其他存储位置设置为零、空白或其他空值。

参见:初始化, 重置。

[来源: ISO/IEC/IEEE FDIS 24765, 3.710]

3.682

明文文件编码 `clear text file encoding`

基于首先使用一些特定字符集定义人类可读表示然后为该集定义编码的数据表示技术类别。

[来源: ISO/IEC 15474-1:2002, 4.1]

3.683

客户端 `client`

在对象上调用操作或从服务器请求服务的实体、代码或进程。

[来源: ISO/IEC 29110-1-2:2024, 3.27]

3.684

客户端对象 `client object`

请求由另一个对象执行服务的对象。

[来源: ISO/IEC 10746-2:2009, 13.4.5]

3.685

时钟脉冲发生器 `clock pulse generator; CPG`

用于产生均匀定时信号的电子单元。

示例: 集成在微控制器 (MCU) 中。

[来源: ISO/IEC/IEEE FDIS 24765; 3.715]

3.686

克隆 `cloning`

从集群检查点实例化集群。

[来源: ISO/IEC 10746-3:2009, 8.1.24]

3.687

闭循环 `closed loop`

没有出口且其执行只能通过来自循环所在的计算机程序或过程外部的干预来中断的循环。参见: UNTIL, WHILE。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.688

闭式子例程 `closed subroutine`

存储在一个给定位置的子程序, 而不是在每次调用它时将其复制到计算机程序的每个位置。

[来源: ISO/IEC/IEEE FDIS 24765, 3.718]

3.689

闭项 `closed term; ground term`

由常量和运算符组成且不含变量的项。

[来源: ISO/IEC/IEEE FDIS 24765, 3.719]

3.690

闭项/基项 `closed term/ground term`

包含常量和运算符但不含变量的项。

[来源: ISO/IEC/IEEE FDIS 24765, 3.719]

3.691

收尾过程组 `closing process group`

为正式完成或关闭项目、阶段或合同而进行的过程。

[来源：《项目管理知识体系指南（PMBOK® 指南）》— 第七版]

3. 692

云计算 cloud computing

一种支持通过网络访问可扩展且弹性的共享物理或虚拟资源池的范式，支持自助式预置和按需管理。

示例：资源示例包括服务器、操作系统、网络、软件、应用程序和存储设备。

[来源：ISO/IEC TS 25052-1:2022, 3.3.1]

3. 693

云互操作性 cloud interoperability

云服务按照规定方法交换信息以获得可预测结果，与云服务客户（CSC）的系统交互，或其他云服务交互的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.2.1]

3. 694

云服务 cloud service

通过定义明确的接口调用的、一项或多项通过云计算提供的能力。

[来源：ISO/IEC TS 25052-1:2022, 3.3.2]

3. 695

云服务客户 cloud service customer; CSC

为使用云服务而建立业务关系的当事方。

注：业务关系不一定包含财务协议。。

[来源：ISO/IEC TS 25052-1:2022, 3.3.2]

3. 696

云服务提供商 cloud service provider; CSP

提供云服务的一方。

[来源：ISO/IEC TS 25052-1:2022, 3.3.4]

3. 697

集群 cluster

为停用、检查点、重新激活、恢复和迁移目的而形成单个单元的基本工程对象的配置。

示例：一段包含对象的虚拟内存。

[来源：ISO/IEC 10746-3:2009, 8.1.2]

3. 698

集群检查点 cluster checkpoint

包含集群中基本工程对象检查点的集群模板。

[来源：ISO/IEC 10746-3:2009, 8.1.22]

3. 699

集群管理器 cluster manager

管理集群中基本工程对象的工程对象。

[来源：ISO/IEC 10746-3:2009, 8.1.3]

3. 700

集群模板 cluster template

对象配置的对象模板以及实例化这些对象和建立初始绑定所需的任何活动。

[来源:ISO/IEC 10746-3:2009, 8.1.19]

3.701

聚类 clustering

对一组对象进行分组,使得同一组,即一个集群,中的对象比其他集群中的对象彼此更相似。

[来源:ISO/IEC TR 29119-11:2020, 3.1.22]。

3.702

配置管理服务 configuration management service; CM service

由配置管理工具完成的工作。

[来源:ISO/IEC TR 18018:2010, 3.8]

3.703

配置管理工具 configuration management tool; CM tool

在配置管理活动中通过提供自动化支持能够协助软件工程师的软件产品。

[来源:ISO/IEC TR 18018:2010, 3.9]。

3.704

代码 code

在软件工程中,用编程语言或由汇编程序、编译程序或其他转换程序的正式输出表示的计算机指令和数据定义。

2. 用编程语言表示计算机程序。

3. 赋予具体含义的字符或位模式。

参见:源代码(source code)、目标代码(object code)、机器代码(machine code)、微代码(microcode)。

[来源:ISO/IEC/IEEE FDIS 24765, X.X]

3.705

代码审核 code audit

由某人、某小组、或借助某种工具对源代码进行的独立的审查,其目的是验证其是否符合软件设计文件和程序设计标准。还可能对正确性和有效性进行估计。参见:审核(audit)、静态分析(static analysis)、审查(inspection)和走查(walk-through)。

[来源:GB/T 11457-2006, X.X]

3.706

代码断点 code breakpoint control breakpoint

给定的计算机指令执行时启动的断点。

参见:数据断点(data breakpoint)、动态断点(dynamic breakpoint)、跋断点(epilog breakpoint)、可编程断点(programmable breakpoint)、序断点(prolog breakpoint)和静态断点(static breakpoint)。

[来源:GB/T 11457-2006, X.X]

3.707

代码数据 code data

可替换数据提供了描述性属性可能拥有的有效值列表。

示例:状态、状态代码、状态名称、支付类型、支付类型代码、支付描述。

注:通常,代码数据的属性包括代码、描述或其他“标准”属性,用于描述该代码。当代码被用于业务数据中时,需要有一种转换机制,将代码转换为用户更容易识别的形式。为了满足用户的非功能性需求,例如可用性、可读性、可维护性,开发人员经常创建一个或多个包含代码数据的表格。

[来源: ISO/IEC/IEEE 32430:2025, 3.1.4]

3.708

代码冻结 code freeze

不允许对代码进行非关键更改的时间段。

[来源: ISO/IEC/IEEE FDIS 24765, 3.746]

3.709

代码生成程序 code generator

一种软件工具, 它接受计算机程序的需求或设计作为输入并产生实现需求或设计的源代码。

参见: 应用程序。

[来源: ISO/IEC/IEEE FDIS 24765, 3.747]

3.710

代码审查 code inspection

一种静态分析技术, 它依靠对开发产品的可视的检查, 以检测错误、对开发标准的违反和其他问题。

类型包括: 代码审查、设计审查。

参见: 审查。

[来源: GB/T 11457-2006, X.X]

3.711

代码评审 code review

由一名或多名开发人员逐行审查源代码, 以确认其质量的活动。

参见: 设计评审 (design review)、形式合格评审 (formal qualification review)、需求评审 (requirements review)、测试就绪评审 (test readiness review)。

[来源: ISO/IEC TR 7052:2023, 3.9]

3.712

代码踪迹 code trace

参考: 执行踪迹 (execution trace)。

在计算机程序执行期间执行的指令序列的记录。常取当程序执行时遇到的代码标记的清单的形式。同义词: 代码踪迹 (code trace) 和控制流踪迹 (control-flow trace)。参见: 回顾的踪迹 (retrospective trace)、子例程踪迹 (subroutine trace)、符号踪迹 (symbolic trace) 和变量踪迹 (variable trace)。

[来源: GB/T 11457-2006, X.X]

3.713

代码优化 code tuning

为优化性能而对程序源代码所做的更改, 通常是为了提高速度或减少内存使用。

[来源: ISO/IEC/IEEE FDIS 24765, 3.750]

3.714

代码走查 code walk-through

参见: 走查 (walk-through)。

一种静态分析技术或评审过程, 在此过程中, 设计者或程序员引导开发组的成员通读已书写的设计或编码, 其地成员负责提出问题并对有关技术、风格、可能的错误、是否违背开发标准等方面进行评论。

[来源: GB/T 11457-2006, X.X]

3.715

编码 coding

在软件工程中, 用编程语言表示计算机程序的过程。

参见：软件开发过程。

[来源：GB/T 11457—2006, 2.227, 有修改]

3.716

强制转换 **coerce**

通过使用不同的对象，将一种类型的对象视为另一种类型的对象，参见类型转换（cast）。

[来源：ISO/IEC/IEEE

[来源：24765n:2025, 3.1.28]]

3.717

共存性 **co-existence; coexistence**

产品在与其它产品共享共同环境和资源的同时，能够高效地执行其所需功能，且不对任何其他产品产生不利影响。

[来源：ISO/IEC 25010:2023, 3.3.1]

3.718

认知走查 **cognitive walkthrough**

可用性评估方法，其中一个或多个评估者（或用户）逐步完成使用场景，从用户的角度识别与使用交互式系统成功完成该场景相关的可用性问题。

[来源：ISO TR 25060:2023, 3.4.3]

3.719

内聚度（内聚性） **cohesion**

在软件设计中，衡量模块内元素间关联强度的指标。

参见：耦合。

注：类型包括偶然性、通信性、功能性、逻辑性、过程性、顺序性和时间性。

[来源：ISO/IEC TR 19759:2016, X.X]

3.720

内聚模块强度 **cohesion module strength**

单个程序模块所执行的诸任务在功能上的互相关联的程度 2. 在软件设计中，衡量模块内元素间关联强度的指标 [来源：ISO/IEC TR 19759:2016 软件工程—软件工程知识体系指南（SWEBOK）] 参见：耦合。

注：类型包括偶然性、通信性、功能性、逻辑性、过程性、顺序性和时间性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.754]

3.721

一致内聚度 **coincidental cohesion**

在软件模块所执行的任务之间不存在功能关系的内聚类型。

参见：通信内聚（communicational cohesion）、功能内聚（functional cohesion）、逻辑内聚（logical cohesion）、过程内聚（procedural cohesion）和顺序内聚（sequential cohesion）、时间内聚（temporal cohesion）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.756]

3.722

协作 **collaboration**

为了实现某一目标，类和实例之间进行协作请求交换。

[来源：ISO/IEC/IEEE FDIS 24765, 3.757]

3.723

协作图 collaboration diagram

一种图，它表示围绕某一模型的结构用类目与关联或者实例与链接所组织的交互。与时序图不同的是，协作图展现各实例间的联系。时序图与协作图所表达的信息相似，但展现的方式不同。

参见：时序图（sequence diagram）。

[来源：GB/T 11457-2006, X.X]

3.724

协作式系统之系统 collaborative system of systems; collaborative SoS

系统之系统（SoS）中，组成部分系统或多或少自愿地互动以履行商定的中心目标。

注1：组成系统集体决定如何提供或拒绝服务，从而提供执行和维护一致性的手段。

同义词：协作式 SoS。

[来源：ISO/IEC/IEEE 21841:2019, 3.2.2]

3.725

分支合并 collapse

通过将一个分支与另一个分支集成来终止其开发。

[来源：ISO/IEC/IEEE FDIS 24765, 3.759]

3.726

集合基数 collection cardinality

对于集合值属性，指定其值（即集合）对于每个实例可以或必须有多少成员的规范。

[来源：ISO/IEC/IEEE FDIS 24765, 3.760]

3.727

集合类 collection class

一个类，其中每个实例都是其他类的实例的集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.31]

3.728

集合值 collection-valued

复杂值。

参见：标量。

注：具有组成部分。

3.729

集合值类 collection-valued class

每个实例都是值集合的类。

参见：标量值类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.34]

3.730

集合值属性 collection-valued property

集合值属性。映射到集合类的属性。

同义词：集合属性。

参见：标量值属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.35]

3.731

集合属性 collection-valued property collection property

映射到集合类的属性。

参见：标量值属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.764]

3.732

组合强度 combinatorial strength

用于表示测试参数取值组合的复杂度。

[来源：GB/T 38639-2020, 2.2]

3.733

组合测试 combinatorial testing

基于参数-值（P-V）对组合的规范的测试设计技术。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.18]

3.734

组合测试方法 combinatorial testing method

基于多参数组合进行软件测试的一种方法。

注：在组合测试中，被测试应用被抽象为一个黑盒，它有多个输入参数，其行为受输入参数之间的组合测试影响。

[来源：GB/T 38639-2020, 2.1]

3.735

舒适性 comfort

用户生理上感到舒适的程度。

[来源：GB/T 25000.10-2016, 4.2.2.3.4]

3.736

命令 command

可以输入到计算机系统中以启动操作或影响计算机程序执行的表达式。示例：用于启动计算机会话的“logon”命令。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.11]

3.737

命令语言 command language

用于向计算机系统表达命令的语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.767]

3.738

命令驱动 command-driven

用于说明一系统或操作方式，在其中，用户通过命令指挥系统。

参见：菜单驱动（menu-driven）。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.739

注释 comment

嵌入在计算机程序、作业控制语句或数据集中的信息，为人类读者提供澄清，但不影响机器解释。

[来源：ISO/IEC/IEEE FDIS 24765, 3.769]

3.740

商业现货 commercial-off-the-shelf; COTS

无需进行开发活动即可购买和使用的软件或硬件产品。

参见：（商业现货）软件产品。

注：COTS 软件产品包含面向用户的信息，例如产品描述、网站信息、安装说明和使用说明，以及包含在计算机可读介质（磁盘、光盘只读存储器或可下载文件）上的软件。COTS 还可包括作为独立制成品开发和维护的产品说明、用户信息及软件，但此类产品不适用常规商业收费和许可条款。

[来源：IEEE 1012-2024, 3; ISO/IEC/IEEE 90003:2018, 3.4]

3.741

商业现货软件产品 `commercial-off-the-shelf software product`

由市场驱动的需要定义的、商业上可获得的、并且其适用性已经得到范围广泛的商业用户证实的软件。

[来源：GB/T 25000.40-2018, 4.6]

3.742

调试 `commissioning`

交付准备投入使用的产品之前或相关的程序。

注：调试可能包括最终验收测试、所支持产品相关文件的移交或指导人员。

[来源：IEC/IEEE 82079-1:2019, 3.2]

3.743

提交 `commit`

将开发人员对源代码私有视图所做的更改集成到可通过版本控制系统的存储库访问的分支中。

[来源：ISO /IEC/IEEE FDIS 24765, 3.772]

3.744

提交信息 `commit message`

伴随提交的解释性消息。

注：通常包含对更改及其基本原理的简要描述；贡献者、审阅者或批准者的姓名；对从中获得更改的第三方软件的引用；将其整合到其他分支机构的时间表；以及对与变更关联的问题标识符的引用。

[来源：ISO /IEC/IEEE FDIS 24765, 3.773]

3.745

提交混乱 `commit war`

由不同意特定元素编码方式的开发人员引入的一系列相互冲突和相互逆转的提交。

注：有时以恶意退出开始。

[来源：ISO /IEC/IEEE FDIS 24765, 3.775]

3.746

提交窗口 `commit window`

允许特定分支提交的时间段。

[来源：ISO /IEC/IEEE FDIS 24765, 3.776]

3.747

承诺 `commitment`

导致行为中的一个或多个参与者有义务遵守规则或履行合同的行为。

注：参与承诺行为的企业对象可以是各方或代表一方或多方行事的代理人。在代理人承诺行为的情况下，委托人有义务。

[来源：ISO/IEC 15414:2015, 6.6.2]

3.748

提交者 `committer`

有提交权限的开发者。

[来源: ISO /IEC/IEEE FDIS 24765, 3. 778]

3. 749

共同祖先约束 **common ancestor constraint**

涉及同一祖先类的两个或多个关系路径的约束，并声明后代实例必须通过每条路径与同一祖先实例相关，或者它通过每条路径与不同的祖先实例相关。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 1. 36]

3. 750

(缺陷的) 共同原因 **common cause (of a defect)**

随机原因 **random cause**

过程变异的来源，由于过程各组件的正常和预期交互作用而存在。

参见: (缺陷的) 特殊原因 (special case (of a defect))。

注: 在控制图上，它表现为随机过程变异的一部分，即来自过程的正常或不异常的变异，由控制限内的随机点模式表示。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 780]

3. 751

通用配置项 **common configuration item**

配置中对所有产品通用的实体。

[来源: ISO/IEC 26563:2022, 3. 2]

3. 752

公共耦合 **common coupling**

一种耦合类型，其中两个软件模块访问共同的数据。

参见: 内容耦合、控制耦合、数据耦合、混合耦合、病态耦合。

[来源: ISO/IEC/IEEE FDIS 24765, X. X]

3. 753

公共数据 **common data**

可以被两个或多个非嵌套的计算机程序模块访问的数据，而无需在模块之间显式传递为参数。

参见: 局部数据。

[来源: ISO/IEC/IEEE FDIS 24765, X. X]

3. 754

共同特征 **common features**

指示一个关键过程域的实施和制度化是否有效、可重复和持久的属性。

3. 755

公共存储 **common storage**

在软件系统中，主存储器中可被两个或更多模块访问的部分。

参见: 全局数据。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 783]

3. 756

通用性 **commonality**

产品线所有应用共有的功能性和非功能性特性的集合。

[来源: GB/T 41865-2022, 3. 9]

3. 757

通用性指数 **commonality index**

用于衡量功能、子系统、组件或产品线通用性的指标。

注：通用性指标的计算公式可以是度量或度量组合（功能）。通用性指标的计算公式反映了功能、子系统、组件或产品线的相对规模、工作量、复杂性或重要性，并以权重形式体现。功能点是用于计算建立核心资产或产品线开发所需工作量的一种成本估算方法。通用性指标通常与功能点结合使用。

[来源：ISO/IEC 26564:2022, 3.2]

3.758

公共环境耦合 common-environment coupling

这是一种耦合类型，在这种耦合中，两个软件模块访问相同的共享数据。

参见：内容耦合，控制耦合，数据耦合，混合耦合，病态耦合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.782]

3.759

沟通限制 communication constraints

对沟通内容、时机、受众或送达沟通的个人的限制，这些限制通常源自特定的法律法规、技术或组织政策。

[来源：ISO/IEC/IEEE FDIS 24765, 3.786]

3.760

通信接口 communication interface

协议对象的接口，可以在交互参考点绑定到拦截器对象或另一个协议对象的接口

[来源：ISO/IEC 10746-3:2009, 8.1.14]。

3.761

沟通管理 communication management

支持 ODP 系统内对象间通信的对象管理。

[来源：ISO/IEC 10746-2:2009, 14.2]。

3.762

通信内聚 communicational cohesion

一种内聚类型，其中软件模块执行的任务使用相同的输入数据或有助于产生相同的输出数据。

参见：偶然内聚、功能内聚、逻辑内聚、过程内聚、顺序内聚、时间内聚。

[来源：ISO/IEC/IEEE FDIS 24765, 3.790]

3.763

通信领域 communications domain

一组有能力进行互操作的协议对象。

[来源：ISO/IEC 10746-3:2009, 8.1.13]

3.764

沟通规划 communications planning

确定如何满足利益相关方的信息和沟通需求的过程：谁需要什么信息，何时需要，以及如何提供给他们。

[来源：ISO/IEC/IEEE FDIS 24765, 3.791]

3.765

社区 community

为了达成某一目标而形成的一组对象的配置。

注：该目标通过一份合同来表达，合同详细规定了如何达成此目标。

[来源：ISO/IEC 10746-3:2009, 5.1.1]

3.766

社区对象 community object

代表社区的复合企业对象。

[来源: ISO/IEC 15414:2015, 6.2.2]

3.767

只读光盘存储器 (CD-ROM) compact disc read only memory; CD-ROM

可读取但不可擦除或重写的光盘。

[来源: ISO/IEC/IEEE FDIS 24765, 3.795]

3.768

紧缩 compaction

在微程序设计中, 将微程序转换为功能等效、但执行速度更快或长度更短的微程序的过程。参见: 局部紧缩, 全局紧缩。

[来源: ISO/IEC/IEEE FDIS 24765, 3.796]

3.769

比较器 comparator

比较两个计算机程序、文件或数据集以识别共性或差异的软件工具 条目注释 1: 比较的典型对象是源代码、目标代码、数据库文件或测试结果的相似版本。

[来源: ISO/IEC/IEEE FDIS 24765, 3.797]

3.770

兼容性 compatibility

产品、系统或组件与其他产品、系统或组件交换信息或执行其所需功能的程度, 同时共享相同的硬件或软件环境。

[来源: ISO/IEC 25010:2023, 3.3]

3.771

兼容性测试 compatibility testing

在与其他独立产品共享环境(共存)时, 以及在需要与其他系统或组件进行信息交流(互操作)时, 测量测试项满足要求的程度。

[来源: GB/T 38634.1-2020, 3.6]

3.772

兼容的评估模型 compatible assessment model

用于执行评估的可操作的模型, 其满足已定义的、与参考模型一致的要求(模型的目的、范围、要素和标志、对参考模型的映射及结果解释)。

3.773

补偿型决策技术 compensatory decision technique

一种多属性决策技术, 允许部分属性上的更优表现, 补偿一个或多个其他属性上的较低表现; 采用权衡取舍的方式。

参见: 非补偿型决策技术、加权求和法、层次分析法、无量纲化缩放。

[来源: ISO/IEC/IEEE FDIS 24765, 3.801]

3.774

补偿型模型 compensatory model

多准则决策模型, 其中复合度量由单独加权的项组成, 其中具有高值的标准(也称为属性项)可以按比例补偿每个权重的低值[ISO/IEC 33003:2015 信息技术 — 过程评估 — 过程测量框架的要求,

3.2]。

注：补偿型模型表明改进更重要的措施（具有更高权重的措施）更有可能增加或提高整体综合价值而不是提高不太重要的价值。该模型假设无论标准的测量水平如何，标准的权重（影响水平）都保持不变。

[来源：ISO/IEC/IEEE FDIS 24765, 3.802]

3.775

胜任特征 competency

个体成功履行特定工作角色或职业职能并实现绩效区分所必需的、可观察且可测量的知识、技能、能力、行为及其他深层特征。

注1：胜任特征通常随具体工作角色或职业职能不同而要求不同的熟练程度。

注2：胜任特征可体现为能够显著区分卓越绩效与一般绩效的个体特征，其内涵可涉及动机、特质、自我形象、态度或价值观等深层因素。

注3：参见胜任（能）力（competence, ISO/IEC/IEEE FDIS 24765, 3.803）。胜任能力通常指个体达到预期结果的总体能力或能力水平，侧重能力的整体性与达成性；胜任特征则更强调在特定工作情境下可观察、可测量并能区分绩效差异的具体特征或能力要素，通常用于描述岗位或角色相关的细分能力。

[来源：GB/T 42560-2023, X.X; 有修改]

3.776

主管评估员 competent assessor

已证明具备进行评估以及监控和验证过程评估合规性的能力的评估员。

[来源：ISO/IEC 33001:2015, 3.2.12]

3.777

编译 compile

将以高级语言表达的计算机程序翻译成它的机器语言等价物。

参见：汇编、反编译、解释。

[来源：ISO/IEC/IEEE FDIS 24765, 3.808]

3.778

编译时间 compile time

在某一软件模块的编译期间所出现的某一事物。

参见：建模时间、运行时间

3.779

编译并执行 compile-and-go

计算机程序的编译、链接、载入、执行之间不间断的操作技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.809]

3.780

编译程序 compiler

将高级语言编写的源程序转换为等价的机器语言的一种计算机程序。

参见汇编程序、解释程序、交叉编译程序、增量编译程序、根编译程序。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.810)]

3.781

编译程序代码 compiler code

用编译程序能识别和处理的形式表示的计算机指令和数据定义。参见 汇编代码(assembly code)、解释代码(interpretive code)和机器代码(machine code)。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.811)]

3.782

编译程序的编译程序 `compiler compiler`

见：序号 752 编译程序生成器（`compiler generator`）。

[来源：GB/T 11457-2006, X.X]

用于构建编译器部分或全部的转换器或解释程序。[来源：ISO/IEC/IEEE FDIS 24765, 3.813]—[编译器生成程序；编译器编译程序；元编译器]。

3.783

编译指令源语句 `compiler directive source statement`

用于定义宏或标签、指示编译程序插入外部源语句（例如包含语句）、指示条件编译或未由其他任何类型属性所描述的语句的源语句。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.784

编译程序的生成程序 `compiler generator`

用来构造编译程序的一部分或全部的翻译程序或解释程序。同义词：编译程序的编译程序（`compiler compiler`）和元编译程序（`metacompiler`）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.813]

3.785

完整的 ICOM 编码 `complete ICOM code`

一种图特征引用，使用点标记法将 ICOM 码与参照图连接起来。

[来源：IEEE 1320.1-1998, 2.1.29]

[来源：ISO/IEC/IEEE FDIS 24765, 3.816]

3.786

完整过程 `complete procedure`

以进入程序开始并以退出程序为结束的整个过程包含的所有活动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.817]

3.787

完备判定表 `complete table`

对所有条件项组合都存在一条可符合的规则判定表。

注：实际上，扩展条目判定表包括限定条目，因此属于混合条目判定表。任何扩展或混合条目判定表均可转换为限定条目表。

[来源：ISO 5806:1984, 3.17]

3.788

完备性 `completeness`

IT 服务支持用户指定的所有目标、目的和数据的程度。

[来源：ISO/IEC TS 25011:2017, 3.2.1.1]

3.789

完代码 `completion code`

通过批处理程序传递给作业流处理器的代码，用以影响输入流中后续步骤执行的代码。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.820)]

3.790

完成准则 `completion criteria`

测试活动完成的条件。

[来源：GB/T 38634.2-2020，4.1.2]

3.791

完成时间定理 completion time theorem

实时调度定理。

注：对于一组独立的周期性任务，如果所有任务同时启动时每个任务都满足其第一个截止期限，则对于任何启动时间间的组合都会满足截止期限。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.792

复杂产品 complex products

客户需求复杂、产品构成复杂、产品技术复杂、制造流程复杂、生产管理复杂的一类产品。

[来源：GB/T 36457-2018，X.X]

3.793

复杂可编程逻辑器件 complex programmable logic device; CPLD

具有完全可编程的与/或门阵列的硬件组件。

[来源：ISO/IEC/IEEE FDIS 24765，3.823]

3.794

复杂可编程逻辑器件(CPLD) complex programmable logic device (CPLD); CPLD

具有完全可编程的与/或门阵列的硬件组件。

[来源：ISO/IEC/IEEE FDIS 24765，3.823]

3.795

复杂性 complexity

系统或部件设计或实现难于理解和验证的程度，由下述因素确定，如：接口的数量和错综程度，条件转移的数量和错综程度，嵌套的深度，数据结构类型，以及其他一些系统特性。与 2.1452 条 简单性 (simplicity) 相对。

[来源：GB/T 11457-2006，2.260]

3.796

复杂度矩阵 complexity matrix

用来给功能类型分配权重的数据表。

注：复杂度矩阵基于数据元素类型，及其引用的记录类型（对数据功能的复杂度）或者逻辑文件（对事务功能的复杂度）的数量，来分配权重。

[来源：GB/T 42588-2023，X.X]

3.797

依从性 compliance

产品或系统遵循相关标准、约定或法规以及类似规定的程度。

[来源：GB/T 37970-2019，X.X]

3.798

部件，构件 component

构成系统的部分之一（系统或程序的基本部分）。部件可以是硬件或软件且可以进一步划分为其他部件。

注1：术语“模块 (module)”、“部件 (component)”和“单元 (unit)”常常可以互换使用或在不同的方法中，定义作为另一个的子元素，取决于上下文。这些术语的关系尚未标准化。

[来源：GB/T 11457-2006，X.X]

3.799

组件内容管理系统 component content management system; CCMS

支持从编写、审阅到发布的整个信息开发生存周期的内容管理系统，包括模块化内容的重用。

注：如果模块化内容基于 XML，则可供管理的各个 XML 元素由 XML 模式或 DTD 定义。无需指定多种标记语言。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.4]

3.800

构件图 component diagram

一种图示各构件之间的组织与依赖关系的图。

[来源：GB/T 11457-2006, X.X]

3.801

组件实现 component implementation

实现组件的活动，包括单元测试。

[来源：ISO/IEC 26553:2018, 3.8]

3.802

组件集成测试 component integration test

对一组相关组件的测试活动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.829]

3.803

构件标准 component standard

描述数据或程序的部件特征的标准。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.830)]

3.804

部件测试 component testing

对独立的硬件或软件部件的测试。

同义词：模块测试 (module testing)。

参见：集成测试 (integration testing)、接口测试 (interface testing)、系统测试 (system testing) 和单元测试 (unit testing)。

[来源：IEEE 1012-2024, 3.1]

3.805

复合组件模型 composed model

由若干原子组件模型组成。

注：它包含结构和行为的形式化描述模型。

[来源：GB/T 36457-2018, X.X]

3.806

组合[类] composite [class]

通过一个组合关系与一个或多个类相关联的类。参见：序号 789 组合 (composition)。

[来源：GB/T 11457-2006, X.X]

3.807

组合聚合 composite aggregation

同义语：序号 781 组合 (composite)。

[来源：GB/T 11457-2006, X.X]

3.808

组合键 **composite key**

由两个或更多属性组成的键。

注：[键风格]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.832]

3.809

复合关键词 **composite keyword**

由两个或多个其他关键词组成的关键词。

[来源：ISO/IEC/IEEE 29119-5:2024, 3.2]

3.810

组合测度 **composite measure**

根据构造规格说明（反映型或构成型）定义的、从构造体的多项测度进行一系列操作派生的变量，该规格代表了所关注构造体的潜变量与其测度之间建立关联的方式。

[来源：ISO/IEC 33003:2015, 3.4]

3.811

复合对象 **composite object**

以组合形式表达的对象。

[来源：ISO/IEC 10746-2:2009, 9.2]

3.812

组合状态 **composite state**

一种由并发（正交）子状态或顺序（互斥）子状态所组成的状态。参见：序号 5064 条 子状态（substate）。

[来源：GB/T 11457-2006, X.X]

3.813

复合任务 **composite task**

包含嵌套对象的任务。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.836)]

3.814

复合（组合）类型 **composite type**

一种数据类型，其每个成员都由多个数据项组成。参见：原子类型（atomic type）。

示例：一个名为 PAIRS 的数据类型，其成员是有序对 (x, y)。

[来源：(ISO/IEC/IEEE FDIS 24765, 3.837)]

3.815

组合 **composition**

两个或更多对象的结合，产生一个处于不同抽象层次的新对象。

两个或更多行为的结合，产生一个新行为。

参见：分解

[来源：ISO/IEC 10746-2:2009, 9.1]

3.816

计算性数据使用 **computation data use; c use**

在任意类型的语句中引用变量的值。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.7]

3.817

计算组件 computational component

计算视点中指定的一种组件，具有控制接口及已声明的操作接口集合：(a) 扮演服务器角色的操作接口 (facets)，(b) 扮演客户端角色的操作接口 (receptacles)，(c) 发起携带类型化事件通知的宣告的操作接口 (事件源)，(d) 消费携带类型化事件通知的宣告的操作接口 (事件接收器)，(e) 支持属性访问器和修改器的操作接口 (属性)。

[来源：ISO/IEC 10746-3:2009, 7.1.16]

3.818

计算容器 computational container

用于承载一组已声明的计算组件类型的容器，有一个管理接口，可通过该接口请求去创建一个指定类型的计算组件并将其加入容器的内容中、从容器中删除一个计算组件、列出当前包含的计算组件以及列出它所提供的计算工厂。

[来源：ISO/IEC 10746-3:2009, 7.1.17]

3.819

计算工厂 computational factory

返回指向其所创建的计算对象的接口引用的工厂。

[来源：ISO/IEC 10746-3:2009, 7.1.15]

3.820

计算接口模板 computational interface template

适用于信号接口、流接口或操作接口的接口模板。

注：计算接口模板包括适当的信号、流或操作接口签名、行为规范和环境契约规约。

[来源：ISO/IEC 10746-3:2009, 7.1.10]

3.821

计算对象模板 computational object template

包含一组可实例化的计算接口模板、行为规约以及环境契约规约的对象模板。

[来源：ISO/IEC 10746-3:2009, 7.1.9]

3.822

计算视点 computational viewpoint

ODP 系统及其环境的一个视点，通过将系统功能分解为一系列在接口处进行交互的对象，实现系统的分布。

[来源：ISO/IEC 10746-3:2009, 4.1.1.3]

3.823

计算机 computer

能够执行大量计算任务的功能单元，包括大量算术运算和逻辑运算，且无需人工干预。

参见：计算设备 (computing device)。

注1：计算机可以由一个独立单元或多个互连单元组成。

[来源：ISO/IEC 2382:2015, X.X]

3.824

计算机中心 computer centre

信息处理中心 data processing centre

包含人员、硬件和软件，旨在提供信息处理服务的设施。

[来源：ISO/IEC 2382:2015, X.X]

3.825

计算机犯罪 computer crime

通过使用、篡改或破坏计算机硬件、软件或数据而实施的犯罪。

[来源: ISO/IEC 2382:2015, X.X]

3.826

计算机数据 computer data

计算机设备和计算机设备之间或计算机设备内部通信用的数据。这种数据可以是外部的（计算机可读形式），也可以是驻留在计算机设备内的，可以是模拟信号，也可以是数字信号。

[来源: GB/T 11457-2006, X.X]

3.827

计算机分代, computer generation

计算机代次

计算机历史分类的类别，主要基于其制造中使用的技术。

示例：基于继电器或真空管的第一代，基于晶体管的第二代，基于集成电路的第三代。

[来源: ISO/IEC 2382:2015, X.X]

3.828

计算机图形学 computer graphics

借助计算机构建、处理、存储和显示图像的方法和技术。

[来源: ISO/IEC 2382:2015, X.X]

3.829

计算机指令 computer instruction

编程语言中的语句，指定计算机要执行的操作以及相关操作数的地址或值。宽泛地说，指计算机程序中的任何可执行语句。

参见指令格式 (instruction format)，指令集 (instruction set)。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.830

计算机语言 computer language

旨在使人类能够与计算机交流的语言 cf. 设计语言、查询语言、编程语言。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.831

计算机网络 computer network

以数据通信为目的的数据处理节点及其互连。

[来源: ISO/IEC 2382:2015, X.X]

3.832

计算机性能评估 computer performance evaluation

测量计算机系统性能并研究提高性能的方法的工程学科。参见系统配置文件、吞吐量、利用率、工作负载模型。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.833

计算机程序 computer program

使计算机硬件能够执行计算或控制功能的计算机指令和数据定义的组合。

符合特定程序设计语言规则并由特定功能、任务或问题解决方案所需的声明和语句或指令组成的

句法单元。

参见：软件

[来源：ISO/IEC 2382:2015, X.X]

3.834

计算机程序摘要 computer program abstract

为潜在用户提供足够信息以确定程序是否适合他们的需求和资源的计算机程序的简要描述。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.835

计算机程序注解 computer program annotation

嵌入至计算机程序、命令语言、作业控制语句或数据之间的说明信息，其目的在给读者提供澄清性材料，并不影响机器的解释工作。

加到或散置在源语言语句当中的描述、附注或解释，在目标语言中这些是无效的。

[来源：GB/T 11457-2006, 2.278]

3.836

计算机程序认证 computer program certification

一个系统、部件或计算机程序符合其规定的需求，对操作使用是可接受的一种书面保证。例如，一计算机系统是安全的允许在定义的环境中操作的书面的认可。

为使系统获准投入运行性使用，对系统遵循规定的需求是可接受的所做的正式演示。

验证系统或部件遵循规定的需求，且其操作使用是可接受的过程。

3.837

计算机程序部件 computer program component(CPC)；CPC

计算机软件配置项的功能的或逻辑的显著的部分，典型地是两个或多个软件单元的集合。

3.838

计算机程序配置标识 computer program configuration identification

配置管理的元素，它由为系统选择配置项并在技术文档中记录它们的功能和物理特征组成。

参见：分配的配置标识、功能的配置标识、产品配置标识、基线

3.839

计算机程序配置项 computer program configuration item(CPCD)；CPCD

为配置管理设计的软件的集合，它在配置管理过程中，作为单个实体对待。

参见：硬件配置项、配置项

3.840

计算机程序开发计划 computer program development plan

为开发某一软件产品而做的项目计划。

参见：计算机程序开发计划

3.841

计算机程序确认 computer program validation

在开发过程期间或结束时对系统或部件进行评价，以确定它是否满足特定的需求的过程。

注1：在设计和开发中，确认关系到检查产品是否符合用户要求的过程、。

注2：确认一般是在规定的操作条件下对最终产品进行的。在早期阶段，这样做是必要的、。

注3：“确认过的”一词用来表示相应的状况、。

注4：如果有几种不同的预期用途，可进行多种确认。

参见：验证

3.842

计算机程序验证 computer program verification

评价系统或部件，以确定软件开发周期中的一个给定阶段的产品是否满足在阶段的开始确立的需求的过程。

程序正确性的形式证明。

注1：在设计和开发中，验证是指对某项指定活动的结果进行检查的过程，以确定该活动是否符合明确的需求。

注2：“验证过的”一词用来表示相应的状况。

参见：确认、正确性的证明

3.843

计算机资源 computer resource

用于特定用途的计算机设备、程序、文档、服务、设施、耗材和人员。

参见：计算机资源分配。

[来源：ISO/IEC/IEEE FDIS 24765, 3.859]

3.844

计算机资源分配 computer resource allocation

将计算机资源分配给当前和等待的作业。参见动态资源分配、存储分配。

示例：将主内存、输入/输出设备和辅助存储分配给计算机系统中并发执行的作业。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.845

计算机科学 computer science

与计算机信息处理有关的科学技术分支。

[来源：ISO/IEC 2382:2015, X.X]

3.846

计算机安全 computer security

通过采取适当措施，保护数据和资源免受意外或恶意行为的损害。

注：这些行为若未经授权，可能是修改、破坏、访问、泄露或获取。

[来源：ISO/IEC 2382:2015, X.X]

3.847

计算机软件组件 computer software component; CSC

计算机软件配置项中功能或逻辑上不同的部分，通常是两个或多个软件单元的集合。

参见：计算机软件配置项、软件配置项、软件项。

[来源：ISO/IEC/IEEE FDIS 24765, 3.861]

3.848

计算机软件部件 computer software component (CSC); CSC

计算机软件配置项的功能或逻辑上可区分的部分，通常是两个或多个软件单元的集合。

参见计算机软件配置项，软件配置项，软件项。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.849

计算机软件配置项 computer software configuration item, CSCI

软件配置项 software configuration item, SWCI

指定用于配置管理并在配置管理过程中被视为单个实体的软件集合。

参见：计算机软件组件，硬件配置项，配置项，软件配置项，软件项。

[来源: ISO/IEC/IEEE FDIS 24765, 3.862]

3.850

计算机系统 computer system computing system

计算系统

由一台或多台计算机和相关软件组成的一种功能装置（系统）。

3.851

计算机字 computer word

在给定计算机中，作为一个单元存储、寻址、发送和操作的位或字符的序列。同义词：2.292 条 计算机字（computer word）。

一计算机存储器的元素，它能容纳如 a 中的位或字符序列。

在某些语言中具有意义作为一实体考虑的位或字符序列。例如，在计算机语言中的保留字。

参见：二进制数字（位）、字节

3.852

计算机辅助的 computer-aided; CA

指计算机承担部分工作的技术或过程。

[来源: ISO/IEC 2382:2015, X.X]

3.853

计算机辅助设计 computer-aided design; CAD

利用计算机设计设备或系统，将其显示在计算机显示器或打印机上，模拟其运行，并提供其性能统计数据。

[来源: ISO/IEC/IEEE 26514:2022, 3.1.10]

3.854

计算机辅助软件工程 computer-aided software engineering; CASE

利用计算机辅助软件工程过程。

参见：集成开发环境。

示例：将软件工具应用于软件设计、需求跟踪、代码生成、测试、文档生成和其他软件工程活动。

[来源: ISO/IEC 15940:2013, 2.2]

3.855

计算机辅助软件工程需求 computer-aided software engineering needs; CASE needs

计算机辅助软件工程工具特性所满足的组织需求。

[来源: ISO/IEC TR 14471:2007, 2.1.3]。

3.856

计算机辅助工具 computer-aided software engineering tool; CASE tool

可通过为软件和系统工程生存周期活动提供自动化支持来协助软件和系统工程师的软件产品。

注：CASE 工具可以仅在选定的功能领域或在广泛的功能领域提供支持。

[来源: ISO/IEC/IEEE FDIS 24765, 3.600]

3.857

基于计算机的软件系统 computer-based software system; CBSS

运行于计算机上的软件系统。

注：CBSS 可以是用户在其终端或等效的机器用户界面上看到的数据处理系统。它包括硬件以及实现用户所需数据处理功能所需的所有软件（系统软件和应用软件）。

[来源: ISO/IEC 14756:1999, 4.5]

3.858

计算机化 computerization

借助计算机实现的自动化。

[来源: ISO/IEC 2382:2015, X.X]

3.859

计算中心 computing centre

通过操作计算机和辅助硬件以及通过工作人员提供的服务来为各类用户提供计算机服务的设施。

[来源: ISO/IEC/IEEE FDIS 24765, 3.870]

3.860

计算设备 computing device

可以执行大量计算的功能单元, 包括大量算术运算和逻辑运算, 无论是否有人工干预。

参见: 计算机 (computer)。

注 1: 计算设备可以由独立的单元或若干互联的单元组成。它也可以是提供一组特定功能的设备, 如电话或个人信息管理器, 或者提供更通用功能的设备如笔记本电脑或台式计算机。

[来源: ISO/IEC 19770-5:2015, 3.6]。

3.861

计算数据使用 computation data use

在任何类型语句中, 变量值的使用。

3.862

概念分析 concept analysis

通过应用分析推导系统概念。

参见: 分析 (analysis)。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.863

概念漂移 concept drift

决策边界发生变化时发生的漂移。

注: 概念漂移表现为系统行为偏离预期行为, 预期行为基于某种公认的客观事实衡量标准, 并且通常在时间序列或流数据中观察到。

[来源: IEEE 7014-2024, 3.1]

3.864

运营构想 Concept of operations; ConOps

组织对新的、修改的或现有组织系统的一个或一系列运营行为的设想或意图的文字和图形的概略表述。

注 1: 运营构想通常在长远战略规划和年度运营计划中得到体现。在年度运营计划中, 运营构想覆盖了为实现组织绩效目标而同时或相继进行的一系列相关行动。

注 2: 运营构想提供确定运行空间、系统能力、界面和运行环境边界的基础。

注 3: 运营构想通常与组织相关, 而运行概念是关于系统的。

[来源: ISO/IEC/IEEE 15288:2023, 3.9]

3.865

运营构想文档 concept of operations document; ConOps document

面向用户的文档, 从最终用户视点描述系统的运行特性。

参见: 运行概念描述。

[来源: ISO/IEC/IEEE FDIS 24765, 3.876]

3.866

概念阶段 concept phase

系统生存周期中确定用户需求并描述和评估系统概念的时间段。

注: 在需求阶段之前。

[来源: (ISO/IEC/IEEE 24765:2017, X.X)]

3.867

概念数据模型 conceptual data model

从用户的角度解释数据组的数据模型。

[来源: GB/T 42588-2023, X.X]

3.868

概念信息 conceptual information

使受众理解产品操作原理以执行所需任务的解释和描述。

[来源: ISO/IEC/IEEE 26514:2022, 3.1.12]

3.869

概念模型 conceptual model

与某些努力相关的概念模型。

[来源: IEEE 1320.2-1998 (R2004), 3.1.39]]

3.870

概念系统设计 conceptual system design

系统设计活动, 涉及指定系统组织的逻辑方面、流程和系统中的信息流

[来源: ISO/IEC 2382:2015, X.X]。

3.871

关注点 concern

与利益相关方有关的或重要的事项。

示例: 可负担性、敏捷性、可用性、可依赖性、灵活性、可维护性、可靠性、韧性、易用性和可行性是(系统)关注点的示例。生存性、损耗、退化、损失、过时(淘汰)也是关注点的示例。PESTEL 助记符号提示了(存在)关注点的可能方面: 政治、经济、社会、技术、环境和法律。

[来源: ISO/IEC/IEEE 42020:2019, 3.8]

3.872

简洁性 conciseness

用最少代码量实现某个功能的软件属性。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.873

具体类 concrete class

能够直接用来创建实例的类。

并发性 concurrency

系统的属性, 其中事件可以彼此独立发生, 因此没有顺序。

参见: 步骤, 并发使能

[来源: ISO/IEC 15909-1:2004, 2.1.7]

3.874

并发的 concurrent

指在同一时间间隔内发生两个或多个活动，这些活动可以通过交替执行（interleaving）或同时执行（simultaneous execution）来实现。

指一类问题、进程、系统或应用：其中许多活动并行发生，输入事件的顺序通常不可预测，且事件经常重叠。参见 parallel（并行）、simultaneous（同时）。

注：并发系统或应用具有多条控制流（many threads of control）。

3.875

并发通信图 concurrent communication diagram

以异步和同步消息通信、事件同步和访问被动信息隐藏对象的形式描述并发任务及其接口的网络图。

[来源：ISO/IEC 24765:2025, 3.885]

3.876

并行阐述 concurrent elaboration

生存周期过程实例在项目期间并行制定，这些过程实例产生的信息项和工件同步演进。

注1：这被称为信息项的并行详细说明。

[来源：ISO/IEC 30103:2015, 3.1]。

3.877

并发使能 concurrent enabling

对于一组转换模式，当所有相关的输入位置都包含足够的标记，以满足该转换模式中每个输入弧注释对其施加的所有需求的总和时，该状态即为转换模式。

[来源：ISO/IEC 15909-1:2019, 3.5]

3.878

并发使能（针对变迁模式） concurrent enabling (of transition modes)

对于一个变迁模式的多重集，当所有相关的输入库所包含足够令牌，以满足该多重集中每一变迁模式对其输入弧标注所产生的全部需求的总和时所处的状态。

[来源：ISO/IEC 15909-1:2019, 3.5]

3.879

并发进程 concurrent processes

可以同时地在多处理机上执行或异步地在单处理机上执行的若干进程。各并发进程可以相互作用，一个进程在接受另一进程的信息之前或一外部事件出现之前可以把执行挂起。与顺序进程（sequential processes）相对。

3.880

并发子状态 concurrent substate

一种能与包含在同一组合状态中的另一子状态同时保有的子状态。

参见：组合状态、互斥子状态

3.881

并发任务架构 concurrent task architecture

对系统或子系统中并发任务的接口和互联方式的描述。

[来源：ISO/IEC/IEEE 24765:2017, 3.888]

3.882

条件 condition

布尔值不包含布尔运算符的表达式。

示例：“ $A < B$ ”是一个条件，“A 和 B”不是。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.13]

3.883

条件条目 condition entry

表明条件与特定规则的相关性

[来源：ISO 5806:1984, 3.8]。

3.884

条件桩 condition stub

描述问题时需要考虑的素有条件清单。

[来源：ISO 5806:1984, 3.1]

3.885

条件控制结构 conditional control structure

一种程序设计控制结构，它允许程序中使用根据指定条件的满足情况而加以选择的控制流。例如，按情况、若……则……否则……。

3.886

条件信息 conditional information

与每个产品相关的信息。

[来源：ISO/IEC/IEEE24765:2017, 3.892]

3.887

条件跳转 conditional jump

仅当满足指定条件时才会发生跳转。

参见：无条件跳转

[来源：ISO/IEC/IEEE24765:2017, 3.893]

3.888

条件过程 conditional process

在某些规定条件下可以是强制性的，在其他规定条件下可以是可选的，在其他规定条件下可以超出范围或不适用的过程。

[来源：ISO/IEC 29110-5-6-4: 2025, 3.6)]

3.889

条件任务 conditional task

在某些特定条件下为必选，在其他特定条件下为可选，并且在另外一些特定条件下可超出范围或不适用的任务。

[来源：ISO/IEC 29110-1-2:2024, 3.28]

3.890

条件文本 conditional text

标记为排除在最终内容交付的一个或多个版本之外的文本。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.8]。

3.891

实施采购 Conduct Procurements

获得卖方应答、选择卖方并签订合同的过程。

[来源：项目管理知识体系指南（PMBOK®指南）第六版]。

3.892

不确定性锥 cone of uncertainty

项目固有的不确定性随着项目持续时间而减少的表现。

[来源: ISO/IEC/IEEE 24765:2017, 3.896]。

3.893

保密性 confidentiality

产品确保数据仅能被授权访问者访问的能力。

[来源: ISO/IEC 25010: 2023, 3.6.1]

2. IT 服务确保数据仅能被授权者访问的程度。

[来源: ISO/IEC TS 25011: 2017, 3.2.3.1]

3.894

可配置性 configurability

可变性机制支持成员产品配置的程度。

[来源: ISO/IEC 26555:2015, 3.6]

3.895

配置 configuration

〈计算机系统〉由组成部分的数量、性质及互连关系所界定的计算机系统或部件的组成排布形式。

〈配置管理〉硬件或软件在技术文档中载明、或在产品中实现的功能特性与物理特性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.899, 有修改]

3.896

配置审计 configuration audit

详细审查流程、产品定义信息、符合要求的书面验证以及产品检查以确认产品已达到其要求的属性或符合已发布的产品配置定义信息。

[来源: ISO/IEC/IEEE 24748-8:2019, 3.1]

3.897

配置基线 Configuration baseline

在产品、产品组件、服务或服务组件的生存周期内的特定时间正式指定的配置信息。

注: 配置基线加上对这些基线的批准更改, 构成当前配置信息。

[来源: ISO/IEC/IEEE 24765c:2014, 3.772]

3.898

配置控制 configuration control

配置管理的要素, 包括评估、协调、批准或不批准, 以及在正式建立其配置标识后对配置项的变更实施。

参见: 变更控制 (change control, 配置标识 (configuration identification) 和 配置状态报告 (configuration status accounting))。

[来源: ISO/IEC/IEEE 24765: 2017, 3.902]

3.899

配置控制委员会 configuration control board; CCB

负责评估并批准或否决配置项变更提议, 并确保已批准的变更得以实施的一组人员。

注: 该机构可批准变更提案, 使其成为已批准的修改; 也可以否决变更提案; 或推迟作出决定。

同义词: 变更权限。

[来源: IEEE 828-2012, 2.1]

3.900

配置标识 configuration identification

配置管理的组成要素，包括为系统选取配置项，并将其功能特性与物理特性记载于技术文档中。

[来源：ISO/IEC/IEEE FDIS 24765, 3.904]

3.901

配置索引 configuration index

记录构成产品的各项配置项信息的配置管理文档。

参见：配置项开发记录、配置状态记录。

[来源：ISO/IEC/IEEE FDIS 24765, 3.905]

3.902

配置项 configuration item; CI

指定用于配置管理并在配置管理过程中被视为单一实体的硬件、软件或两者的集合或项目。

为配置管理设计的硬件、软件或两者的集合，它在配置管理过程中作为一个单个实体来对待。参见：2.691 条 硬件配置项 (hardware configuration item)、2.290 条 计算机软件配置项 (computer software configuration item)、2.309 条 配置标识 (configuration identification) 和 2.366 条 关键项 (critical item)。[GB/T 11457-2006]。

[来源：ISO/IEC/IEEE 15288:2023, 3.11]

3.903

配置项开发记录 configuration item development record

在配置管理中使用的文档，基于配置审计和设计评审的结果描述配置项的开发状态。

参见：配置索引 (configuration index)，配置状态报告 (configuration status accounting)。

[来源：ISO/IEC/IEEE 24765:2017, 3.907]

3.904

结构管理 Configuration management

运用技术和行政指导及监督的学科，旨在：识别并记录配置项的功能和物理特性，控制这些特性的变更，记录并报告变更处理和实施状态，并验证是否符合规定的要求。

[来源：IEEE 828-2012, 2.1]。

3.905

配置管理 (CM) configuration management; CM

技术和组织性活动，包括配置识别、控制、状态报告和审计。

[来源：IEEE 828-2012 IEEE 系统和软件工程配置管理标准, 2.1]

3.906

配置管理权限 configuration management authority

指定负责确保配置管理活动得到规划和执行的个人或小组。

[来源：IEEE 828-2012, 2.1]。

3.907

配置管理数据库 configuration management database; CMDB

用于存储配置管理信息的特定类型的存储库，通常是数据存储，用于记录配置项的属性以及配置项之间在其整个生存周期中的关系。

参见：配置管理系统。

注：配置管理数据库包含每个配置项的所有相关详细信息以及它们之间重要关系详细信息，用于组织存储有关正在使用的硬件和软件信息。

[来源：IEEE 828-2012 IEEE, 2.1; ISO/IEC 19770-5:2015, 3.8; ISO/IEC TR 7052:2023, 3.10; 有修改]

3.908

配置管理数据库 configuration management database; CMDB

包含每个配置项的所有相关详细信息以及它们之间重要关系的详细信息的数据库。

参见：配置管理系统

[来源：ISO/IEC 19770-5:2015, 3.8]

3.909

配置管理库系统 configuration management library system

存取软件基线库内容的工具和规程。

3.910

配置管理计划 configuration management plan; CMP

项目管理计划的一个组成部分，描述如何在配置控制下识别和说明项目工件，以及如何记录和报告对它们的变更。

[来源：项目管理知识体系指南（PMBOK® 指南）—第六版]

3.911

配置管理记录 Configuration management record

与配置管理相关的数据、信息或知识的永久性、可读形式。

[来源：INCONE 系统工程手册 第4版，附录 E]。

3.912

配置管理报告 Configuration management report

为相关方准备的账户，用于沟通配置管理活动的状态、结果和成效。记录特定变更请求对任何流程、组织、决策（包括任何必要的变更通知）、产品和服务的影响。

[来源：INCONE 系统工程手册 第4版，附录 E]。

3.913

配置管理策略 Configuration management strategy

项目配置管理所需的方法、进度安排、资源和具体注意事项。描述并记录如何以统一且受控的方式对已建立的基线进行授权更改。

[来源：INCONE 系统工程手册 第4版，附录 E]。

3.914

配置管理系统 configuration management system

识别持续演变系统的组件以控制这些组件的变更，并在整个生存周期内保持完整性和可追溯性的规程。

参见：配置管理数据库

[来源：ISO/IEC/IEEE 24765: 2017, 3.911]。

3.915

配置参数 configuration parameter

可变组件或接口提供的参数，以便在发生绑定时选择其值。

[来源：ISO/IEC 26553:2018, 3.9]

3.916

配置状态记录 (CSA) configuration status accounting (CSA)

配置管理要素，包括有效管理配置所需信息的记录和报告。

参见：配置控制、配置标识、配置索引、配置项、开发记录。

注：此信息包括已批准配置标识的列表、配置变更的状态和已批准变更的实施状态。

[来源：ISO/IEC TR 18018:2010, 3.10]

3.917

配置单元 configuration unit

可放入配置管理库系统和从库中检索的一个配置项或部件的最低层次的实体。

3.918

禁闭 confinement

在被核准访问期间，防止对数据做未经核准的变更、使用、破坏和抛弃的措施。

对程序和进程施加的限制，其目的是使它们不能访问或影响未经核准的数据、程序或进程。

参见：完整性

3.919

确认偏误 confirmation bias

证实预先存在的信念或假设的认知偏差。

[来源：《项目管理知识体系指南》第七版]

3.920

冲突 conflict

一个文件版本的更改无法与其所应用的文件版本协调。

注：合并来自不同分支的版本或两个提交者同时处理同一文件时可能会发生。

[来源：ISO/IEC/IEEE 24765:2017, 3.915]

3.921

冲突管理 conflict management

解决冲突而对冲突情形进行的处理、控制和指导。

[来源：项目管理知识体系指南（PMBOK®指南）第五版]

3.922

符合性 conformance

产品、流程或服务对规定要求的满足。

对规定要求的遵循。

结果达到既定质量要求的程度。

注：符合性用于指对自愿性标准要求的满足。

参见：合规性。

[来源：项目管理知识体系指南（PMBOK(R) Guide)—第七版]

3.923

一致性点 conformance point

对于一致性测试目的，可以观察行为的参考点。

[来源：ISO/IEC 10746- 2:2009, 10.7]

3.924

一致性工作 conformance work

质量成本的框架中，为了一次就正确完成计划工作而做的附加工作。一致性工作包括与预防和检查相关的行动。

[来源：项目管理知识体系指南（PMBOK® 指南）第五版]。

3.925

合格评定 conformity assessment

特定要求已得到满足的证明。

注：符合性评定可能得出否定性结论，即证实特定要求未得到满足。

[来源：ISO/IEC 29110-1-2:2024， 3.30]

3.926

符合性评价 conformity evaluation

对产品、过程或服务达到规定要求的程度所进行的系统性考核。

[来源：GB/T 25000.51-2016， 4.1.4]

3.927

符合性评价报告 conformity evaluation report

说明对就绪可用软件产品（RUSP）实施评价的行为和结果的文档。

[来源：GB/T 25000.51-2016， 4.1.5]

3.928

混淆矩阵 confusion matrix

用于描述分类器在真假值已知的一组测试数据上的性能的表格。

[来源：ISO/IEC TR 29119-11:2020， 3.1.24]

3.929

连接 connection

程序的某一部分对程序另一部分的标识符（即，在另外地方发现的标识）的引用。

为了传递信息而在功能部件之间建立的关系。

参见：接口

3.930

连通性 connectivity

系统或设备无需修改即可附加到其他系统或设备的能力。

[来源：ISO/IEC 2382:2015， X.X]

3.931

接续的 consecutive

关于两个顺序的事件或项的发生，而没有任何其他事件或项的插入、即，一个立即跟在另一个之后。

[来源：GB/T 11457-2006， X.X]

3.932

一致性 consistency

系统或部件的文档与组成部分之间的统一程度、标准化程度及无矛盾程度。

参见：可追溯性。

[来源：ISO/IEC/IEEE FDIS 24765， 3.928]

3.933

一致状态 consistent state

处理已经完全执行的点，功能用户需求已经满足，没有什么可以做的了。

[来源：ISO/IEC 20926:2009， 3.10]

3.934

授权合并 consolidation of an entitlement

将两个或多个授权合并为一个统一授权的过程。

注：进行授权合并可以是为了简化对当前授权状态的理解，也可以是作为与许可方谈判的结果。授权模式支持对授权合并操作的记录。

[来源：ISO/IEC 19770-3:2016, 3.1.5]

3.935

常量 constant

数字、词汇、或外部指定的固定（不变）值。

[来源：IEEE 1175.3-2004, X.X]

3.936

恒定失效期 constant-failure period

在系统或组件的生存周期中，硬件以近似恒定的速率发生失效的时间段。

参见：早期失效期、耗损失效期、浴盆曲线。

[来源：ISO/IEC/IEEE 24765:2017, 3.934]

3.937

组成配置项 constituent configuration item

作为较大配置项的组成部分（部分）的要控制的单个项目，例如参考模型、硬件原型或软件构建。

[来源：IEEE 828-2012, 2.1]

3.938

构成系统 constituent system; CS

构成系统之系统（SoS）一部分的独立系统。

注：组成系统可以是一个或多个 SoS 的一部分。每个组成系统本身都是一个有用的系统，有自己的开发、管理、利用、目标和资源，但在 SoS 内相互作用以提供 SoS 的独特功能。

[来源：ISO/IEC/IEEE 21839:2019, 3.1.1]

3.939

约束 constraint

对系统、其设计或实现或用于开发或修改系统的过程的外部限制。

注：约束是通过影响力或强制力施加在解决方案上的因素，可能会限制或修改设计。

[来源：GB/T 45802-2025, 3.1.7]

3.940

约束依赖 constraints dependency

变点之间、变体之间以及变点与变体之间的关系。

[来源：ISO/IEC 26558:2017, 3.5]

3.941

构造 construct

使用过程评估来进行测量的概念，例如抽象想法、意象、潜在主题或主题内容。

注：在过程测量框架中，结构（也指潜在结构）是理论概念，例如过程质量特征和过程属性。人们赋予结构的含义是其理论定义，它描述了其不同的维度（方面）。

[来源：ISO/IEC 33003:2015, 3.5]

3.942

构造的能力 constructed capability

由组织单位的元素或不同组织的元素所构成的能力，为满足某些特殊要求的目的而组合。

[来源：GB/T 11457-2006, X.X]

3.943

构建 construction

编写、组装或生成资产的过程。

注：专注于创建源代码的活动集合。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.944

消费者对象 consumer object

接收所传达信息的对象。

[来源：ISO/IEC 10746-2:2009, 13.4.4]

3.945

消费品 consumer product

可供消费者使用、打算供消费者使用或可能使用的产品。

[来源：IEC/IEEE 82079-1:2019, 3.8]

3.946

消费者软件包 consumer software package

为最终用户设计和销售的实现确定功能的 COTS 软件产品；软件及其相关文档作为一个整体打包出售。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.947

容器 container

在运行时将事务、安全、事件和持久性集成到组件行为中的框架。

[来源：ISO/IEC 19500-3:2012, 4.1]

3.948

容器接口 container interface

允许访问数据的数据存储库接口。

[来源：ISO/IEC 10746-3:2009, 14.1.1.2]

3.949

包容层次 containment hierarchy

由模型元素与其间存在的包容联系所组成的一种命名空间层次。包容层次形成一个图。

[来源：GB/T 11457-2006, X.X]

3.950

内容 content

包含以文本、图像、视频、声音或其他媒体表示的信息的交互式或非交互式对象。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.5]

3.951

内容块 content chunk

满足特定用户特定任务需求的单元内容。

示例：一份研究报告分为五个内容块，分别涉及背景信息、方法论、结果、结论和建议。

注：内容块定义了一个子主题，该子主题值得用户单独考虑。

[来源：ISO TR 25060:2023, 3.2.8]

3.952

内容一致性 content consistency

信息项内容间的语义一致性。

[来源：ISO/IEC 30103:2015, 3.2]

3.953

内容耦合 content coupling

耦合的一种类型，其中，一个软件模块的部分或全部内容包括在另一模块的内容中。

注：与 2.243 条 公共环境耦合 (common environment coupling)、2.336 条 控制耦合 (control coupling)、2.393 条 数据耦合 (data coupling)、2.714 条 混合耦合 (hybrid coupling) 和 2.1123 条 相依耦合 (pathological coupling) 相对。

[来源：GB/T 11457, X.X, 有修改]

3.954

内容管理 content management

控制信息单元及其元数据，以便在具有可变结构和格式的文档或信息产品中进行选择性重用。

注：用户文档的内容管理是指管理帮助主题、概念解释、故障排除程序、合规性声明以及软件产品的名称和主机平台等变量，并应用元数据标签来格式化输出。

[来源：ISO/IEC/IEEE FDIS 24765, 3.952]

3.955

内容管理平台 content management platform; CMP

构建统一、持久且可供其他系统访问的数据库的管理系统。

参见：内容管理系统。

注：内容管理平台 (CMP) 充当融合中心和存储库，用于存储来自各种内部和外部系统的数据。CMP 能够容纳具有不同结构和命名约定的不同数据类型和格式。

[来源：IEEE 7005 2021, 3.1]

3.956

内容管理系统 content management system; CMS

提供组件以供复用和关联、进而构建内容对象，并能将大型内容对象拆解为可独立管理的组件的系统。

参见文档管理系统。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.5]

3.957

内容对象 content object

独立的内容单元。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.6]

3.958

内容类型 content type

内容的具体指标。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.7]

3.959

内容单元 content unit

大型信息对象中可识别和可管理的部分。

注：可供管理的各个内容单元通常由 XML 模式或 DTD 定义。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.8]

3.960

周境 context

功能或图标中的一组功能运行的直接环境。

[来源: ISO/IEC/IEEE FDIS 24765, 3.958]

3.961

周境完备性 context completeness

在所有指定的使用周境中, 产品或系统在有效性、效率、抗风险和满意度特性方面能够被使用的程度。

示例: 在小屏幕、低带宽、非专业人员操作以及软件的容错模式(例如没有网络连接)的条件下软件的可用程度。

注: 周境完备性既可以当作在所有预期的使用周境中产品在有效性、效率、抗风险和满意度特性方面为达到指定目标被指定用户使用的程度也可以通过在所有预期的使用周境中支持使用的产品属性的存在来进行指定或测量。

[来源: GB/T 25000.10-2016, 4.2.2.5.1]

3.962

周境覆盖 context coverage

在指定的使用周境和超出最初设定需求的周境中, 产品或系统在有效性、效率、抗风险和满意度特性方面能够被使用的程度。

注: 使用周境与使用质量和一些产品质量特性或子特性相关(即“指定条件”)。

[来源: GB/T 25000.10—2016, 4.2.2.5]

3.963

使用周境 context of use

用户、任务、设备(硬件、软件和原材料)以及使用某系统、产品或服务的物理和社会环境。

[来源: GB/T 25000.22-2019, 4.6]

3.964

上下文敏感帮助 context-sensitive help

上下文相关帮助

用户屏幕信息类型, 其中显示的内容取决于用户对软件当前状态及用户任务进度的视角。

参见: 嵌入式文档、印刷文档。

[来源: ISO/IEC/IEEE 26514:2022, 3.14]

3.965

上下文因素 contextual factor

情境因素

驱动或影响正在开发的组织或系统的要求的条件。

注: 环境因素可能会发生变化, 并且可以定期审查他们制定的要求或控制措施。

[来源: IEEE 7000:2021, 3.1]

3.966

语境模型 contextual schema

应用了数据模型的使用周境边界的正规描述。

注: 这是业务的信息需求的高层描述, 它比概念模型更一般化; 它包括体系结构的(系统)周境的整体视图。

[来源: GB/T 25000.24:2019, 4.4]

3.967

邻接分配 contiguous allocation

连续分配

一种存储分配技术，存储的程序或数据分配给一等于或大于其尺寸的存储块，所以，逻辑上邻接的程序或数据被赋予物理邻接的存储单元。与 2.1101 条 调页（paging）相对。

[来源：GB/T 11457, X.X, 有修改]

3.968

应急事件 contingency

可能影响项目执行，且可通过相关储备应对的事件或情况。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.969

偶然性因素 contingency factor

对规模、成本或进度计划的一种调整（增加），以便考虑由于不完全的规格说明以及在应用领域上缺乏估计经验等而可能对这些参数估计过低。

[来源：GB/T 11457-2006, X.X]

3.970

应急方案 contingency plan

为某一风险因素（在演变为问题时）制定的处理方案。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.971

应急储备 contingency reserve

在进度或成本基准内，为主动应对已知风险而分配的时间或资金。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.972

条件阈值 contingent threshold

可靠性要求中的一个阈值，由于实现该阈值所带来的权衡取舍存在不确定性，因此该阈值是可取的但并非必需的。

注：条件阈值通常量化一种能力水平，只有在不产生过高成本、不利风险或不利运行影响的情况下才能实现该能力水平。

[来源：IEEE 982:2024, 3.1]

3.973

持续过程改进 continual process improvement

流程改进计划的持续循环，以加强和改进支持业务的流程，包括一个或多个改进项目或举措，可以串联或并行实施。

[来源：ISO/IEC TR 33014:2013, 3.1]

3.974

连续性 continuity

在所有可预见的情况下提供 IT 服务的程度，包括将中断造成的风险降低到可接受的水平。

[来源：ISO/IEC TS 25011:2017, 3.2.4.1]

3.975

持续交付 continuous delivery; CD

软件工程实践，允许通过使用自动化工具将新系统（包括软件）频繁发布到阶段或各种测试环境。

[来源：IEEE 2675-2021, 3.1]

3.976

持续部署 continuous deployment

通过验证既定功能与有效性以降低风险，将变更部署至生产环境的自动化过程。

[来源：IEEE 2675-2021, 3.1]

3.977

持续改进 continuous improvement

持续不断地识别、实施和评估改进实践方式的循环，以增强组织或项目实现其目标的能力。

[来源：ISO/IEC 33202:2024, 3.12]

3.978

持续集成 continuous integration; CI

持续合并工件的技术，包括团队中所有开发人员的源代码更新，进入共享主线以构建和测试开发的系统。

[来源：IEEE 2675-2021, 3.1]

3.979

不断迭代 continuous iteration

持续迭代

没有出口的循环。

[来源：GB/T 11457-2006, X.X]

3.980

连续表示 continuous representation

能力成熟度模型结构，其中能力等级为每个指定过程域内的过程改进提供推荐顺序。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.981

持续风险管理 continuous risk management

定期、持续地分析计划的活动、项目或计划的进展并处理已识别的风险因素的过程。

注：包括制定备选方案与应急预案，以便在风险因素实际发生时，能够通过替代解决方案降低其影响。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.982

合同管理 contract administration

管理获取方与供应商之间合同及关系，包括审查并记录供应商的当前或以往绩效表现、确定所需的纠正措施，以及管理合同变更。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.983

合同变更控制系统 contract change control system

用于收集、跟踪、裁定和沟通有关合同变更的系统。

[来源：项目管理知识体系指南（PMBOK 指南）第五版]

3.984

合同管理计划 contract management plan

用于说明如何监督需求文档的交付和执行工作说明书的执行情况，绩效评估和控制变更的文件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.982]

3.985

合同条款和条件 contract terms and conditions

合同规定的法律、财务和管理方面的内容。

[来源：GB/T 11457—2006, 2.333]

3.986

合同工作分解结构 contract work breakdown structure; CWBS

适用于合同的整体工作分解结构的一部分，由供应商制定和维护。

[来源：ISO/IEC/IEEE FDIS 24765, 3.983]

3.987

合同工作分解结构 (CWBS) contract work breakdown structure (CWBS); CWBS

由供应商提出和维护的，适用于将合同的整体工作分解为部分工作的分解结构。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.988

合同背景 contractual context

已知特定合同生效，因而要求合同各方主体遵循合同约定的特定行为。[定义-英文]翻译。

[来源：ISO/IEC/IEEE FDIS 24765, 3.984]

3.989

合同需求 contractual requirement

对顾客需求开展分析与细化后形成的一组需求，其适配纳入采办方与相关机构之间一份或多份采购文件包、正式合同及供方协议。

参见：采购方、客户需求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.985]

3.990

合同要求的审核 contractually required audit

一般由需方或由独立的机构主持进行。此过程对产品或服务提供一个独立的评价，以决定产品或服务是否符合它们的需求。

[来源：GB/T 11457—2006, 2.334]

3.991

逆变 contravariance

属性覆盖的规则，覆盖属性中输入参数的可接受值集是被覆盖属性中对应输入参数可接受值集的超集（或同集）；同时，覆盖属性中输出参数的可接受值集是被覆盖属性中对应输出参数可接受值集的子集（或同集）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.986]

3.992

控制 control

对比实际绩效与计划绩效，分析偏差，评估趋势以改进过程，评价可能的备选方案，并提出必要的纠正措施建议。

[来源：项目管理知识体系指南（PMBOK®指南），第七版]

3.993

控制账户 control account

一种管理控制点。在该控制点上，把范围、预算、实际成本和进度加以整合，并与挣值比较，以测量绩效。[项目管理知识体系指南（PMBOK®指南），第五版]。

参见：工作包（work package）。

3.994

控制断点 control breakpoint

作为代码断点的许用术语。

执行特定计算机指令时触发的断点。

参见：数据断点(data breakpoint)、动态断点(dynamic breakpoint)、跋断点(epilog breakpoint)、可编程断点(programmable breakpoint)、序断点(prolog breakpoint)、静态断点(static breakpoint)。

[来源：GB/T 11457—2006，2.335 和 2.220]

3.995

控制图 control chart

按时间顺序展示过程数据，并将这些数据与既定的控制界限相比较的一种图形。控制图有一条中心线，有助于观察图中的数据点向两边控制界限偏移的趋势。

[来源：项目管理知识体系指南（PMBOK®指南），第七版]

3.996

控制聚类 control clustering

将控制物/控制目标与其控制对象合并到同一个任务中的任务结构化准则。

[来源：ISO/IEC/IEEE FDIS 24765，3.989]

3.997

控制沟通 control communications

在整个项目生存周期中对沟通进行监督和控制的过程，以确保满足项目干系人对信息的需求。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.998

控制成本 Control Costs

监督项目状态，以更新项目成本，管理成本基准变更的过程。

[来源：项目管理知识体系指南（PMBOK®指南），第五版]

3.999

控制耦合 control coupling

一个软件模块通过向另一个模块传递信息影响后者执行流程的一种耦合类型。

参见：公共环境耦合(common environment coupling)、内容耦合(content coupling)、数据耦合(data coupling)、混合耦合(hybrid coupling)和相依耦合(pathological coupling)。

[来源：ISO/IEC/IEEE FDIS 24765，3.990]

3.1000

控制数据 control data

选择一程序中的操作方式或子方式，给顺序流指向，或者直接影响软件操作的数据。例如，循环控制变量(loop control variable)。

[来源：GB/T 11457—2006，2.337]

3.1001

控制字段 control field

在连续逻辑记录间的取值变化或未变化状态会影响主程序的控制流走向的，包含一个或多个输入变量的一种字段。

[来源：ISO/IEC/IEEE FDIS 24765，3.992]

3.1002

控制流 control flow

在计算机程序执行期间操作执行的顺序。参见数据流(data flow)。

同义词：控制流(flow of control)

[来源：ISO/IEC/IEEE 29119-4:2021，3.14]

3.1003

控制流程图 control flow diagram

展示了系统或程序执行过程中所有可能操作序列集合的图。参见数据流图、调用图及结构图。

注：类型包括框图、流程图、输入-处理-输出图及状态图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.994]

3.1004

控制流子路径 control flow sub-path

测试项中的可执行语句的序列。

[来源：GB/T 38634.4-2020 4.8]

3.1005

控制流踪迹 control flow trace

执行踪迹

代码踪迹

控制流踪迹

在计算机程序执行期间执行的指令序列的记录。

参见：回顾的踪迹、子例程踪迹、符号踪迹、变量踪迹。

注：常取当程序执行时遇到的代码标记的清单的形式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1703]

3.1006

控制信息 control information

通过指定要处理的数据的内容、时间或方式来影响基本过程的数据。

[来源：GB/T 42449-2023, X.X]

3.1007

控制语言 control language

用于标识作业的序列、描述它们对操作系统的要求并控制它们执行的语言。

参见：“作业控制语言”。

[来源：GB/T 11457-2006, 2.826 和 2.341]

[来源：如还有“作业控制语言”术语，则保留 参见]

3.1008

控制环回 control loopback

一个函数的输出环回以控制同一图中的另一个函数。

同义词：反馈(feedback)。

[来源：IEEE 1320.1-1998, 2.1.34]

3.1009

控制程序 control program

监督程序 supervisory program

执行系统 executive

执行程序 executive program

管态模式 supervisor

通常作为操作系统一部分的计算机程序，用于控制其他计算机程序的执行，并调度计算机系统的工作流程。

参见：管理态（supervisor state）、内核（kernel）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4615]

3.1010

质量控制 control quality

监督并记录质量管理活动的执行结果，以评估绩效、确保项目输出完整、正确且满足客户期望的过程。

注：质量控制过程的目的是在用户验收和最终交付之前测量产品或服务的完整性、合规性和适用性。

[来源：GB/T 39266-2020, 3.4]

3.1011

控制资源 control resources

确保按计划为项目分配实物资源，以及根据资源使用计划监督资源实际使用情况，并采取必要纠正措施的过程。

[来源：项目管理知识体系指南（PMBOK(R) 指南）第六版]

3.1012

控制风险 control risks

在整个项目中实施风险应对计划、跟踪已识别风险、监控剩余风险、识别新风险和评估风险过程有效性的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.998]

3.1013

控制风险 control risks

监控风险 monitor and control risks

在整个项目期间，实施风险应对计划、跟踪已识别的风险、监测残余风险、识别新风险以及评估风险管理全过程的过程。

[来源：项目管理知识体系指南（PMBOK®指南）—第七版]

3.1014

控制进度 Control Schedule

监督项目状态，以更新项目进度和管理进度基准变更的过程。

[来源：项目管理知识体系指南（PMBOK®指南）—第6版]

3.1015

控制范围 Control Scope

监督项目和产品的范围状态，管理范围基准变更的过程。

[来源：项目管理知识体系指南（PMBOK®指南）—第6版]

3.1016

控制相关方参与 control stakeholder engagement

全面监督项目干系人之间的关系，调整策略和计划，以调动干系人参与的过程。

[来源：项目管理知识体系指南（PMBOK®指南）—第五版]

3.1017

控制语句 control statement

从多组备选程序语句中进行选择，或影响操作执行顺序的程序语句。

参见：赋值语句（assignment statement）、声明语句（declaration statement）。

示例：if-then-else、case 语句。

[来源：ISO/IEC/IEEE FDIS 24765 3.999]

3.1018

控制存储器 control store

在采用微程序控制的计算机中，用于存放微程序的计算机存储器。

参见：微指令字（microword）、毫微程序存储器（nanostore）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1000]

3.1019

控制结构 control structure

通过计算机程序决定控制流的构造。

参见：条件控制结构（conditional control structure）。

3.1020

控制任务 control task

决策控制其他任务执行的任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1001]

3.1021

控制变量 control variable

用于决定是否从循环退出的程序变量。

参见：循环控制变量（loop - control variable）。

3.1022

控制器 controller

与外围设备进行接口连接的设备或计算机芯片。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1002]

3.1023

控制器区域网络 controller area network; CAN

用于系统内智能设备联网的高完整性总线系统。

注：常用于车辆或医疗设备的嵌入式网络。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1003]

3.1024

约定 convention

用于规定一种规范、统一的方法的要求，以确保软件产品的一致性，即统一的数据组织模式或形式。

参见：实践（practice），标准（standard）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1004]

3.1025

会话式[的] conversational

一种交互式系统或操作模式，其中用户与系统的交互类似于人类之间的对话。

参见：批处理（batch）、交互式（interactive）、在线（online）、实时（real time）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1005]

3.1026

会话式编译程序 conversational compiler

增量编译程序 incremental compiler

能够在输入或扫描源语句的过程中尽可能多地完成对每个源语句的转换的编译器。

注：通常用于在线计算机程序的开发和检查。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2213]

3.1027

转换 conversion

修改现有软件，使其能够在不同的环境中以类似的功能运行。

示例：将程序从 FORTRAN 转换为 Ada，将在一台计算机上运行的程序转换为能在另一台计算机中运行的程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1006]

3.1028

转换功能 conversion functionality

用于转换数据或提供其他用户指定转换需求的事务功能或数据功能。

注：转换功能仅存在应用程序的开发或改进过程中。

[来源：ISO/IEC 20926:2009, 3.12]

3.1029

可转换性 convertibility

多种功能规模测量（FSM）方法在同一组功能用户需求下的测量结果转换能力。

[来源：ISO/IEC TR 14143-3:2003, 3.3]

3.1030

协同交互 coordinated interactions

一对交互，其中第一个交互由一个单元生成，作为对另一个单元的刺激，期望第二个交互由另一个单元（甚至可能是第三个单元）生成作为响应，以使发起单元受益于特定目的。

注：一个单元控制此交换的发起，使其他单元从属于该单元。即使是从属单元生成了触发发起单元行为的刺激，这种主从交互也是可能存在的。

[来源：IEEE 1175.4-2008, 3.11]

3.1031

复制 copy

从源读取数据，保持源数据不变，并以可能不同于源的物理形式将相同的数据写入其他地方。

参见：移动（move）。

示例：将数据从磁盘复制到磁带。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1011]

3.1032

核心 core

计算机或处理器中的处理单元，用于管理指令、数据和操作。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1016]

3.1033

核心报告 core report

描述基准度量活动过程和结果的文档。

注：为报告基准度量活动实例的结果，常产生两种核心报告，即执行总结和详细报告。

[来源：GB/T 41866.1-2022（ISO/IEC 29155-1:2017, IDT）]

3.1034

核心价值 core value

在关注系统的语境中被认定为核心的价值。

示例：透明度和隐私性可以作为信息系统的首要属性（核心价值）。

注：核心价值处于由工具性价值或相关价值以及价值体现者所构成的价值集群的中心。核心价值是一种正向价值。

通常，一个关注系统（SoS）具有多个核心价值。

[来源：IEEE 7000:2021, 3.1]

3.1035

协同例程 **coroutine**

在上次操作暂停点开始执行的例程且不要求返回控制至调用它的程序或子程序。

参见：XXX 条 子例程 (subroutine)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1019]

3.1036

信息技术治理 **corporate governance of IT**

在最高管理层面，制定信息技术使用的战略和政策，并对信息技术的使用实施组织层面的控制。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1021]

3.1037

可纠正性 **correctability**

纠正软件缺陷和处理用户投诉所需的努力程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1022]

3.1038

纠正 **correction**

为弥补缺陷并使软件充分运行以满足既定运行需求而实施的、包含问题解决方案的变更。

参见：增强 (enhancement)。

注：术语纠正 (correction) 主要用作维护类型，并用于对修改请求 (MR) 进行分类。

[来源：ISO/IEC/IEEE 14764:2025, 3.1023]

3.1039

纠正措施 **corrective action**

消除不符合原因并防止其再次发生的措施。

参见：预防措施 (preventive action)。

[来源：ISO/IEC 19770-1:2017, 3.14]

3.1040

纠正性维护 **corrective maintenance**

在软件产品交付后为纠正发现的问题而进行的修改。

注：修改是修正软件产品，以满足定义的系统要求。

[来源：ISO/IEC/IEEE 14764:2021, 3.1.4]

3.1041

纠正性支持 **corrective support**

对运行中的系统进行修改，以消除导致故障或异常的缺陷。

注：改正性支持是对运行中的系统进行更改，使其满足已定义的系统要求。

[来源：IEEE 982:2024, 3.1]

3.1042

正确性 **correctness**

系统或部件在规格说明、设计和实现方面无缺陷的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1027]

3.1043

对应关系 **correspondence**

两个或多个架构描述元素之间的已识别或命名的关系。

示例：对应关系用于表达广泛的关系，如等价性、组合性、精化性、一致性、可追溯性、依赖性、约束性、满意度和义务。

注 1：为了对应的目的，能将一个架构描述视为另一个架构描述里的架构描述元素。

注 2：对应能作为不同架构描述里的各架构描述元素之间的，或以不同符号表述的各架构描述元素之间的识别或命名关系。

[来源：GB/T 45630-2025, 3.11]

3.1044

成本汇总 cost aggregation

在项目 WBS 的某个层次或成本控制账户上，对与各工作包相关的较低层次的成本估算进行汇总。

[来源：项目管理知识体系指南（PMBOK 指南）第五版]

3.1045

成本规避 cost avoidance

通过减少支出而非增加收入而产生的收益（正现金流）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1030]

3.1046

成本基准 cost baseline

经过批准的、按时间段分配的项目预算，不包括任何管理储备，只有通过正式的变更控制程序才能进行变更，并用作与实际结果进行比较的依据。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1047

成本基础 cost basis

购置成本 acquisition cost

获得某项资产的全部成本。

注：包括购买价格、交付、安装和将资产投入使用的任何其他成本。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1032]

3.1048

成本约束 cost constraint

对项目预算施加的限制或约束，例如随时间推移可用的资金。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1033]

3.1049

代价函数 cost function

用于描述决策变量不同取值所对应成本的目标函数。

参见：收益函数（income function）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1034]

3.1050

成本管理计划 cost management plan

项目或项目集管理计划的组件，描述如何规划、安排和控制成本。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1051

成本度量 cost measurement

对软件开发成本的预计值进行估算或对实际值进行测量和分析的过程。

[来源：GB/T 36964-2018, 3.7]

3.1052

质量成本 cost of quality; COQ

产品生存周期内所有因投资预防不符合要求、评估产品或服务是否符合要求以及未能满足要求而产生的成本。

注：预防和评估成本（符合性成本）包括质量规划、质量控制和质量保证的成本，以确保符合要求（例如，培训、质量控制体系等）。故障成本（不符合性成本）包括返工不符合要求的产品、组件或流程的成本、保修工作和浪费的成本以及声誉损失。

[来源：《项目管理知识体系指南》第七版]

3.1053

成本绩效基准 cost performance baseline

处于变更控制下的按时间分段的预算，用于比较实际支出与计划支出。

注：用于确定是否需要采取预防或纠正措施以实现项目目标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1037]

3.1054

成本绩效指数 cost performance index; CPI

测量预算资源的成本效率的一种指标，表示为挣值与实际成本之比。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1055

成本加奖励费用合同 cost plus award fee contract; CPAF

成本补偿合同的一种类型，向卖方支付已完成工作的全部合法实际成本，再加上一笔奖励费用作为卖方的利润。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1056

成本加固定费用合同 cost plus fixed fee contract; CPFF

一种成本补偿型合同，买方需向卖方支付其可允许成本（合同中定义的可允许成本）及固定利润（费用）的差额。

同义词：成本加固定费用合同（cost-plus-fixed-fee contract）。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1057

成本加激励费用合同 cost plus incentive fee contract; CPIF

成本补偿合同的一种类型，买方为卖方报销可列支成本（可列支成本由合同确定），并且在达到规定绩效标准时卖方可赚取利润。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1058

成本偏差 cost variance; CV

在某个给定时间点的预算亏空或盈余量，表示为挣值与实际成本之差。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1059

成本效益分析 cost-benefit analysis

财务分析工具，用于确定项目提供的收益与其成本的对比关系。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1060

成本加费用 **cost-plus-fee; CPF**

买方在合同中补偿供应商履行合同工作时的合理费用，并支付一定费用的合同。

同义词：成本加费用合同。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1061

成本补偿合同 **cost-reimbursable contract; cost reimbursable contract**

一种合同类型，买方向卖方支付其实际成本，外加一笔通常代表卖方利润的费用。

[来源：项目管理知识体系指南（PMBOK 指南）第七版]

3.1062

可计数故障 **countable failure**

不满足系统一项或多项明确或隐含的能力需求；达到或超过其可靠性等级的严重性准则；且发生时刻已被估算或记录的异常行为。

[来源：IEEE 982:2024, 3.1]

3.1063

计数器 **counter**

用于记录在计算机程序执行期间，给定事件发生次数的变量。例如，记录循环执行的时间数的变量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1048]

3.1064

计数规则 **counting rule**

获得测量值所依据的条件和程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1049]

3.1065

计数范围 **counting scope**

需纳入功能点计数中的功能用户需求。

[来源：GB/T 42449-2023, 3.14]

3.1066

耦合 **coupling**

软件模块间相互依赖的方式和程度。

参见：内聚度（cohesion）。

注：类型包括公共环境耦合（common environment coupling）、内容耦合（content coupling）、控制耦合（control coupling）、数据耦合（data coupling）、混合耦合（hybrid coupling）和相依耦合（pathological coupling）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1051]

3.1067

覆盖项 **coverage item**

作为测试重点的测试项的可测量属性。

[来源：ISO/IEC/IEEE 29119-2:2021, 3.29]

3.1068

崩溃 **crash**

计算机系统或组件突然且完全失效。

参见：xxx 条，硬件故障（hard failure）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1066]

3.1069

赶工 crashing

一种通过增加资源，以最小的增量成本来缩短项目进度工期的方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.936]

3.1070

创建工作分解结构 create WBS

将项目交付成果和项目工作细分为更小、更易管理的组成部分的过程。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1071

创建 WBS (工作分解结构) create WBS (work breakdown structure)

将项目可交付成果和项目工作分解为更小、更易于管理的组成部分的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.913]

3.1072

创建 creation

通过模型中对象的动作对 <X> 进行实例化。

参见：导引（introduction）。

[来源：ISO/IEC 10746-2:2009, 9.18]

3.1073

危机 crisis

即将出现决定性，并且很可能是不利结果的一种危急事态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1071]

3.1074

危机管理 crisis management

当应急计划无法解决相关问题时所采取的步骤。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1072]

3.1075

标准 criteria

判断或决策所依据的标准、规则或测试，或者产品、服务、结果或过程可以被评估的标准、规则或测试。

2. 确定为信息项内容的特定数据项，用于在评估、审计、测试或评审中评估一个因素。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1073]

3.1076

关键资产 critical asset

有可能对组织目标的实现产生重大影响的资产。

注：资产可以是安全关键型、环境关键型或性能关键型，并可与法律、法规或法定要求相关。关键资产可指为关键客户提供服务所必需的那些资产。资产系统可以类似于单个资产的方式被区分为关键资产。

[来源：ISO/IEC 19770-1:2017, 3.15]

3.1077

关键链法 critical chain method

一种进度规划方法，允许项目团队在任何项目进度路径上设置缓冲，以应对资源限制和项目的不确定性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1075]

3.1078

关键计算机资源 critical computer resource; CCR

由于对计算机资源的潜在需求有可能超出可用量而可能成为项目风险源的计算机资源参数。例如，目标计算机的内存容量和宿主计算机的磁盘空间。

3.1079

关键设计评审 critical design review; CDR

为验证一个或多个配置项的详细设计满足规定要求而进行的评审；建立配置项与设备、设施、软件和人员的其他项之间的兼容性；评估每个配置项的风险区域；以及（如适用）评估可生产性分析的结果，审查初步硬件产品规范，评估初步测试计划，并评估初步操作和支持文件的充分性。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1080

关键设计评审(CDR) critical design review (CDR); CDR

验证一个或多个配置项的详细设计满足是否满足规定要求的评审：确立各配置项之间，以及配置项与相关的设备、设施、软件及人员之间的兼容性；识别并评估每个配置项的风险领域；评估可生产性分析结果、审查初步硬件产品规范、评价初步测试计划的完备性，评定初步操作与保障文档的充分性。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1081

关键信息 critical information

描述软件安全使用方法、说明由该软件生成信息的安全性保障措施，以及对软件生成或存储的敏感个人信息实施保护的信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1077]

3.1082

关键项 critical item

在配置管理中，由于特殊的工程或逻辑因素，需要通过批准的规格说明，在组件级建立技术或库存控制的配置项。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1078]

3.1083

关键路径 critical path

代表项目中最长路径的活动序列，其决定了项目最短可能工期。

参见：xxx 条，关键路径法（critical path methodology）。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.1084

关键路径法 critical path method; CPM

用于估算项目最短工期，并确定进度模型中逻辑网络路径上进度调度灵活度的方法。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1085

关键部分优先 critical piece first

一种系统开发方法，优先实现系统中最关键的部分。

参见：xxx 条，自底向上（bottom-up）、xxx 条，自顶向下（top-down）。

注：关键部分可依据所提供的服务、风险程度、难度或其他准则来定义。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1081]

3.1086

关键属性 **critical property**

主要利益相关方一致认为具有重要影响性的属性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1082]

3.1087

临界范围 **critical range**

用于将软件划分为可接受、临界、不可接受类别的度量值。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1083]

3.1088

临界段 **critical section**

临界区 **critical region**

与其他任务互斥执行的任务内部逻辑部分。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1084]

3.1089

关键软件 **critical software**

由于安全、性能和保障等因素, 可能对用户或环境造成严重影响的软件。

[来源: ISO/IEC TR 29110-5-1-4:2018, 3.7]

3.1090

关键系统 **critical system**

由于安全、性能和保障等因素, 系统可能对用户或环境造成严重影响。

例如: 安全关键项目、断裂关键项目、任务关键项目、关键特性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1085]

3.1091

临界值 **critical value**

经过验证的度量值, 用于识别存在不可接受质量的软件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1086]

3.1092

关键性 **criticality**

需求、模块、错误、故障、失效或其他项对系统的开发和操作冲击的程度。

[来源: IEEE 1012-2024, 3.1]

3.1093

关键的 **critical**

临界的

由于设计不当, 一个系统或一个软件的某些环节或部分在运行时超出了临界范围, 或存在着潜在的、未检测出的错误, 会导致死机、人员伤害、任务失效、数据丢失、财经上的损失或灾难性的设备损坏等严重后果。或指:。

要使用的软件开发技术的成熟程度和有关的风险。

3.1094

交叉汇编程序 **cross-assembler**

在一台计算机上为另一台不同的计算机产生目标代码的汇编程序。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1088]

3.1095

交叉编译程序 `cross-compiler`

在一台计算机上为另一台不同计算机产生汇编代码或目标代码的编译程序。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1089]

3.1096

交叉引用生成器 `cross-reference generator`

交叉引用 `cross-referencer`

一种软件工具, 以计算机程序的源代码作为输入, 生成一份清单作为输出; 该清单标识出程序中的每个变量、标号及其他标识符, 并指明程序中对每个标识符进行定义、赋值或使用的语句。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1090]

3.1097

交叉引用清单 `cross-reference list`

标识计算机程序中的每个变量、标记及其他标识符, 并指明程序中对每个标识符进行定义、赋值或使用的语句的清单。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1091]

3.1098

交叉引用工具 `cross-reference tool`

一种软件维护工具, 使用户能够确定变量在何处被使用, 或特定过程在何处被调用。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1092]

3.1099

加密哈希 `cryptographic hash`

通过校验码来验证系统元素或软件真实性的方法。

[IEEE 2675-2021, 3.1]。

3.1100

累计流程图 `cumulative flow diagram; CFD`

用于表示随时间已完成的特性、处于其他开发状态的特性以及待办特性的图表。

注: 可以在某些中间里程碑处指示功能, 例如已设计但尚未构建的功能。

[来源: Software Extension to the PMBOK® 指南 Fifth Edition]

3.1101

定制软件 `custom software`

根据用户需求规格说明, 为特定应用所开发的软件产品。

[来源: GB/T 25000.1, 3.4]

3.1102

客户满意度 `customer satisfaction`

在评估时刻, 由顾客自身评定的、顾客的需求被满足或超出其预期体验的满足状态。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1110]

3.1103

客户 `customer`

受益方 `beneficiary`

买方 `purchaser`

能够或实际接收针对其需求或要求的产品或服务的个人或组织。

参见：采购方、利益相关方、用户。

示例：消费者、客户、终端用户、零售商、接收方或内部流程中的产品/服务接收方与开发组织同属一家公司的内部部门（例如系统管理部门）、开发公司外部的其他企业或实体、更高层级的项目，或上述要素的组合。

注：术语“客户”包含“消费者”的含义，但适用范围更广。客户既可来自组织内部，也可来自外部。客户属于利益相关方的子集。应用管理组织的客户群体既包括内部业务信息管理机构，也涵盖外部业务信息管理机构及其他应用管理组织。用户或终端用户指实际使用应用软件的个人，而客户则是决定并采购产品或服务的个人或组织。客户或用户组织与应用管理之间的关系，可通过业务信息管理架构进行可视化呈现。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1108]

3.1104

可定制性 *customizability*

IT 服务可按用户要求进行定制的程度。

[来源：ISO/IEC TS 25011:2017, 3.2.7.1]

3.1105

定制化 *customization*

为适配特定受众需求而对软件或信息产品进行的适配调整。

[来源：ISO/IEC/IEEE 26512:2018, 3.7]

3.1106

定制 *customize*

使软件或信息产品适应特定受众的需求。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.15]

3.1107

截止日期 *cut-off date*

软件变更不再纳入当前软件版本或现行文档版本，转而纳入下一版本的界定日期。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1114]

3.1108

切换 *cutover*

在给定时刻将某一系统的功能转移至其后继系统。

[来源：ISO/IEC 2382:2015, ?]

3.1109

周期 *cycle*

a 完成一组事件所经历的时间段。

b 以相同顺序定期重复进行的一组操作，每次重复可能存在差异。

参见：软件开发周期 (software development cycle)，软件生存周期 (software life cycle)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1120]

3.1110

周期窃取 *cycle stealing*

将中央处理器的操作暂停一个或多个周期，以便允许发生其他操作，例如响应来自输入/输出控制器的输出请求从主存储器传输数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3, 1121]

3.1111

周期时间图 *cycle time chart*

展示工作项平均周期时间随时间变化的图表。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1112

循环搜索 **cyclic search**

一种存储分配技术，每次查找合适的存储块时，都从上次分配块的下一个块开始。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1124]

3.1113

每日站会 **daily standup; daily stand-up**

简短的、每日的、有时间限制的会议，用于与敏捷团队的每个成员讨论进度、计划和任何阻塞问题。

[来源：ISO/IEC TR 24587:2021 3.7]

3.1114

危险状态 **dangerous condition**

一种系统的状态，若与环境中的某些状态相结合，将会导致不良后果。

注：ISO/IEC 指南 51 和 IEC 61508-4 中的危险情况是危险状况的一种实例。引入危险状况的概念是为了涵盖不仅在安全范畴内的危险情况，还包括可靠性、完整性、保密性或可靠性范畴中的错误以及其他可能导致不良后果的系统状态。在可靠性范畴内发生的故障情况通常（但并非总是）会导致危险状况。因此，危险状况至少具有以下属性：a) 相关的不良后果；b) 导致危险状况的触发事件；c) 导致危险状况引发不良后果的触发事件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1128]

3.1115

隐性问题 **dark matter**

原项目计划中遗漏、但为完成交付成果所必需开展的工作。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1129]

3.1116

达尔文信息分类体系结构 **Darwin Information Typing Architecture; DITA**

基于 XML 的架构，用于创作、生成和交付主题导向的信息类型内容，这些内容可被以多种方式重复使用并作为单一来源。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.13]

3.1117

仪表盘 **dashboard**

用于展示项目关键指标进展或绩效的图表集。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1118

数据 **data**

以适合人或自动化手段进行通信、解读或处理的方式对事实、概念或指令进行的表述。

参见：数据类型、信息。

注 1：datum—“数据”的单数形式。

注 2：数据在对其进行分析、解释或处理之前可以是捕获、测量或记录形式的信息。数据可以涉及诸如事实、事件、事物、过程或想法等对象，包括在特定情境下具有特定含义的概念。

[来源：ISO/IEC 19770-1: 2017, 3.16]

3.1119

数据抽象 **data abstraction**

定义数据类型和它们的功能特征，忽视它们的详细表示以抽取数据的基本特征。

参见：封装（encapsulation）、信息隐藏（information hiding）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1133]

3.1120

数据动作 data action

数据元素上的系统操作。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1134]

3.1121

数据分析 data analysis

对现实或计划系统中的数据及其数据流的系统研究。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1135]

3.1122

数据分析技术 data analysis technique

用于组织、评估和评价数据与信息的技术。

[来源：项目管理知识体系指南（PMBOK 指南）第六版]

3.1123

数据属性 data attribute

在已知的数据组中，从软件的用户功能需求角度出发表示某种含义的最小信息块。

[来源：GB/T 42452-2023, ?]

3.1124

数据银行 data bank

与给定主题相关并以订户可以查阅的方式组织的数据集。

[来源：ISO/IEC 2382:2015, X.X]

3.1125

数据断点 data breakpoint

存储断点 storage breakpoint

当访问指定的数据项时启动的断点。

参见：代码断点（code breakpoint）、动态断点（dynamic breakpoint）、尾代码断点（epilog breakpoint）、可编程断点（programmable breakpoint）、前代码断点（prolog breakpoint）、静态断点（static breakpoint）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1138]

3.1126

数据缓冲寄存器 data buffer register; DBR

物理存储器中用于在数据移动（例如从输入到处理）期间临时存储数据的区域。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1139]

3.1127

数据特性 data characteristic

数据固有的、可能是偶然的特征、性质或属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1140]

3.1128

数据通信设备 Data Communications Equipment; DCE

分布式计算环境。

[来源：ISO/IEC 19500-2:2012 3.3]

3.1129

数据配置 data configuration

在不改变软件代码或数据库结构的前提下，对数据库或数据存储中添加、修改或删除参考数据或代码数据信息的过程。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.5]

3.1130

数据损坏 data corruption

对数据项进行错误且无法恢复的篡改，包括恶意加密。

[来源：IEEE 982:2024, 3.1]

3.1131

数据耦合 data coupling

输入输出耦合 input-output coupling

一个软件模块的输出作为另一个模块的输入的一种耦合类型。

参见：公共环境耦合（common-environment coupling）、内容耦合（content coupling）、控制耦合（control coupling）、混合耦合（hybrid coupling）、病态耦合（pathological coupling）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1144]

3.1132

数据日期 data date

记录项目状态的时间点。

同义词：截止日期（as-of date），当前日期（time-now date）。

[来源：PMBOK®指南—第六版]

3.1133

数据声明源语句 data declaration source statement

在编译时保留或初始化内存的源语句。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1145]

3.1134

数据定义计算使用对 data definition c-use pair

数据定义和后续计算的数据使用，其中数据使用是使用数据定义中定义的值。

[来源：GB/T 38634.4-2020, 4.10]

3.1135

数据定义谓词使用对 data definition p-use pair

数据定义和后续谓词的数据使用，其中数据使用是使用数据定义中定义的值。

[来源：GB/T 38634.4-2020, 4.11]

3.1136

数据定义 data definition

变量定义 variable definition

变量赋值语句。

[来源：GB/T 38634.4-2020, 4.9, 有修改]

3.1137

数据定义使用对 data definition-use pair

数据定义和后续的数据使用，其中数据使用是使用数据定义中定义的值。

[来源: GB/T 38634.4-2020, 4.12]

3.1138

数据设计 data design

程序数据的设计，特指数据库应用中的表设计。

[来源: SeVocab]

3.1139

数据字典 data dictionary

数据的信息的集合，如名称、描述、创建者、所有者、来源、不同语言的翻译和使用情况。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1150]

3.1140

数据漂移 data drift

模型部署后数据发生变化时产生的漂移。

注：数据漂移是指当新数据点来源于输入空间中训练数据分布较稀疏的区域时所发生的现象。

[来源: IEEE 7014-2024, 3.1]

3.1141

数据元素类型 data element type; DET

唯一、用户可识别、非重复属性。

[来源: GB/T 42449-2023]

3.1142

数据元素 data element

数据项 data item

在 BFC 中唯一的、用户可识别的、不重复的字段。

注 1：数据元素在 BFC 中可以是字符串，也可以是数字或图形元素。

注 2：数据元素的数量总是大于 0。

[来源: GB/T 42448-2023]

3.1143

数据擦除 data erasure

数据项的不可撤销性丢失。

[来源: IEEE 982:2024, 3.1]

3.1144

数据异常 data exception

当程序企图不正确地使用或访问数据时发生的异常。

参见：寻址异常 (addressing exception)、操作异常 (operation exception)、溢出异常 (overflow exception)、保护异常 (protection exception)、下溢异常 (underflow exception)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1155]

3.1145

数据暴露 data exposure

对数据项的未授权访问，但未实际发生未授权使用或传输。

[来源: IEEE 982:2024, 3.1]

3.1146

数据故障 data failure; <sevo> data failure (DF); DF

在目标系统实例处于运行或移除模式期间，发生数据损坏、擦除、暴露或泄露的情况。

[来源：IEEE 982:2024, 3.1]

3.1147

数据文件 data file

作为一个单元处理的一组相关数据记录。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1158]

3.1148

数据流 data flow

在计算机程序执行过程中，数据传输、使用和转换所遵循的顺序。

2. 数据在系统中从一个组件传递到下一个组件的过程。

参见：控制流（control flow）。

[来源：IEEE 7002:2022, 3.1]

3.1149

数据流图 data flow diagram; DFD

数据流程图 data flowchart

数据流图 data flow graph

描述数据源、数据接收端、数据存储和在数据上（如节点上）执行的处理以及数据的逻辑流（如节点之间的连接）的一种图。

将数据源、数据宿、数据存储以及数据相关处理以节点表示，将数据逻辑流向以节点间链路表示的图。

参见：控制流图（control flow diagram），数据结构图（data structure diagram）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1160]

3.1150

数据流测试 data flow testing

基于执行定义-使用对的一类基于结构的测试用例设计技术。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.19]。

3.1151

数据格式 data format

对要通信或存储和检索的数据的指定排列和编码。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1162]

3.1152

数据函数 data function

为满足内部或外部数据存储需求而向用户提供的功能。

注：FPA 为每个数据功能指定类型，并区分以下类型：内部逻辑文件和外部接口文件。

[来源：ISO/IEC 20926:2009, 3.16]

3.1153

数据函数类型 data function type

FPA 分配给数据功能的两种类别：内部逻辑文件和外部接口文件。

[来源：GB/T 42588-2023, ?]

3.1154

数据收集和分析方法 data gathering and analysis methods

用于收集、评估和分析数据信息以深入理解某一情况的方法。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1155

数据收集技术/数据采集技术 data gathering technique

用于从各种来源收集数据和信息的技术。

[来源：项目管理知识体系指南（PMBOK 指南）第六版]

3.1156

数据治理 data governance

对数据及与数据相关资产的定义、生成和使用的权限的执行与实施。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1165]

3.1157

数据组 data group

数据组类型 data group type

一组不同的、非空的、无序的且无冗余的数据元素集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1170]

3.1158

数据信息 data information

进入或者退出应用程序的信息，这些信息满足了用户的信息需求。[来源：GB/T 42588-2023, ?]。

3.1159

数据输入表 data input sheet

以工作表形式说明系统或部件所需必填与可选输入数据的用户文档。

参见：用户手册。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1166]

3.1160

数据库存 data inventory

在信息处理系统中，所有的数据及其特性，包括相互依赖关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1167]

3.1161

数据项 data item

字段 field

一组定义明确的数据元素，与单个标签相关联，该标签定义了其含义和布局。

注：数据项是数据值的物理“容器”。

[来源：ISO/TR 25060:2023, 3.2.6]

3.1162

数据泄露 data leakage

数据项的未经授权使用或转移。

[来源：IEEE 982:2024, 3.1]

3.1163

数据操作 data manipulation

数据的各种处理，但不包括功能过程内的、以及功能过程与持久存储器之间的数据输入输出。

[来源：GB/T 42452-2023, 2.6]

3.1164

数据地图 data map

系统内数据处理的可视化表示。

[来源：IEEE 7002:2022, 3.1]

3.1165

数据介质 data medium

可在其上或其中记录数据并可从中检索数据的介质。

复数形式：数据介质（data media）。

[来源：ISO/IEC 2382:2015, ?]

3.1166

数据模型 data model

以图形或文字形式对数据进行表示，明确规定数据的属性、结构及相互关系。

注1：。

—数据模型可由计算机进行编码和操作。

—数据模型标识实体、域、属性以及与其他数据之间的关系（关联），并提供数据及数据间关系的概念视图。

—数据模型分为逻辑（或功能）数据模型与技术数据模型。

—逻辑数据模型：以实体和关系形式组织的组织数据表示，独立于任何特定数据管理技术。

—技术数据模型：确定数据在数据库中的存储形式以及数据的访问方式。

[来源：ISO/IEC 16350-2015, 4.15]

3.1167

数据移动 data movement

数据移动类型 data movement type

移动单个数据组的基本功能部件。

注1：COSMIC 功能规模测量方法有4种类型的数据移动：输入、输出、读和写。这些同样是4个BFC类型。

注2：为了到达测量的目的，每个数据移动都被认为可以代表特定相关数据操作。

[来源：GB/T 42452-2023, 3.7]

3.1168

数据隐私 data privacy

个人数据的公平、合法处理。

[来源：IEEE 7002:2022, 3.1]

3.1169

数据处理 data processing ; DP

自动数据处理 automatic data processing ; ADP

对数据实施操作的系统化过程。

示例：对数据进行算术或逻辑运算，对数据进行合并或排序，对程序进行汇编或编译，或对文本进行操作，例如编辑、排序、合并、存储、检索、显示或打印。

注：数据处理并非信息处理的同义词。

[来源：ISO/IEC 2382:2015, X.X]

3.1170

数据保护 data protection

采用适当的管理、技术或物理手段，以防范未经授权的故意或意外泄露、修改或销毁数据。

[来源：ISO/IEC 2382:2015, X.X]

3.1171

数据保护影响评估 data protection impact assessment; DPIA

通用数据保护条例中描述的工具，用于预先评估数据处理过程中的隐私风险，进而能够实施控制措

施以降低这些风险。

[来源：ISO/IEC TR 7052:2023, 3.14]

3.1172

数据提供者 data provider

作为数据来源的个人或组织。

[来源：ISO/IEC/IEEE 15939:2017, 3.5]

3.1173

数据质量 data quality

依照指定条件使用数据时，其特性满足明确的和隐含的要求的程度。

[来源：GB/T 25000.1-2021, 3.6]

3.1174

数据质量特性 data quality characteristic

对数据质量有影响的数据质量属性的类别。

[来源：GB/T 25000.12-2017, 4.4]

3.1175

数据质量测度 data quality measure

变量，为其赋值作为数据质量特性的测量结果。

[来源：GB/T 25000.12-2017, 4.5]

3.1176

数据质量模型 data quality model

已定义的特性集合，提供一个框架用于说明数据质量需求和评价数据质量。

[来源：GB/T 25000.12-2017, 4.6]

3.1177

数据记录 data record

由相关数据元素构成的已定义集合，其中包含表示关注属性所需的全部必要数据元素。

[来源：ISO/IEC 29155-1:2017 3.12]

3.1178

数据仓库 data repository

数据仓库中用于访问数据的接口。

[来源：ISO/IEC 10746-3:2009, 14.1.1.2]

3.1179

数据表示技术 data representation techniques

用于传达数据和信息的图形表示或其他方法。

[来源：项目管理知识体系指南（PMBOK(R) 指南）第六版]

3.1180

数据序列器 data sequencer

必要时可在多个层级结构中用数据替换关键字参数的组件。

[来源：ISO/IEC/IEEE 29119-5:2024, 3.3]

3.1181

数据存储 data store

数据和检索信息的组织有序的和持久的汇集。

[来源：GB/T 20917-2007, 3.6]

3.1182

数据结构 data structure

数据元素之间的物理或逻辑关系，用于支持特定的数据操作功能。

注：数据结构通常在技术数据模型和逻辑数据模型中予以规定。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1190]

3.1183

数据结构图 data structure diagram

数据流程图 data flow diagram

实体关系图 entity-relationship diagram

描述一组数据元素、其属性以及之间逻辑关系的图表。

参见：数据流图、实体关系图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1191]

3.1184

数据结构中心设计 data structure-centered design

一种软件设计技术，通过分析系统必须处理的数据集结构导出系统架构。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1192]

3.1185

以数据结构为中心的设计 data structure-centred design

一种软件设计技术，其系统架构源自对系统所处理数据集结构的分析。

参见：输入-处理-输出 (input-process-output)、模块化分解 (modular decomposition)、面向对象设计 (object-oriented design)、快速原型开发 (rapid prototyping)、逐步求精 (stepwise refinement)、结构冲突 (structure clash)、结构化设计 (structured design)、事务分析 (transaction analysis)、转换分析 (transform analysis)。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1186

数据主体 data subject; personally identifiable information principal

个人可识别信息主体 principal

向系统提供其个人数据的个人。

注：判断数据主体是否可识别，需考虑持有数据的隐私利益相关方或其他任何方，能够合理使用的、可识别该自然人的手段。

[来源：IEEE 7014:2024, 3.1]

3.1187

数据系统 data system

为收集、处理、维护、使用、共享、传播或处置数据而组织的离散资源集。

[来源：IEEE 7002:2022, 3.1]

3.1188

数据追踪 data trace

变量追踪 variable trace

数据流追踪 data-flow trace

值追踪 value trace

计算机程序执行期间，对被访问或被修改的变量名称及其取值的记录。

参见：执行追踪、追溯追踪、子程序追踪、符号追踪。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5182]

3.1189

数据传输控制器 data transfer controller; DTC

不经过中央处理器（CPU）控制数据通信的功能单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1196]

3.1190

数据类型 data type

对取值和操作自变量的分类，通常同时包含行为与表示方式。

示例：整数、实数和字符串。

[来源：ISO/IEC 19500-2:2012, 3.2.4]

3.1191

数据使用 data use

访问变量值的可执行语句。

[来源：GB/T 38634.4-2020, 4.13]

3.1192

数据值 data value

数据项的内容。

注1：数据质量是指数据本身，如数据域值和可能的限制。

注2：数据值是通过进行测量对目标实体的属性所赋予的数值或类别。

[来源：GB/T 25000.24-2017, 4.7]

3.1193

数据库 database

存储于一个或多个计算机文件中、相互关联的数据集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1203]

3.1194

数据库设计规范 database design specification

描述软件为实现其功能所必需的、永久性或非永久性数据的内容与格式的文档。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1204]

3.1195

数据库管理系统 database management system; DBMS

结构化数据的有组织的集合。

注：为了使用数据库管理系统（DBMS），需要用数据模型、数据定义和操作语言对数据和相关的操作进行描述。

[来源：GB/T 25000.24-2017, 4.18]

3.1196

数据库视图 database view

数据存储查询的结果集，数据库用户可以像在持久性数据库集合对象中一样查询。

注：这个预先建立的查询命令保存在数据库字典中。与关系数据库中的普通基表不同，当请求访问该视图时，它是根据数据库中的数据动态计算或整理的虚拟表。应用于相关基础表中数据的更改会反映在视图的后续调用中显示的数据中。

[来源：IEEE 2430-2019, 3.1]

3.1197

数据驱动 data-driven

基于证据开展活动。

[来源：IEEE 2675-2021, 3.1]

3.1198

数据项 DataItem

可以保留或处理的可能值或值的结构。

[来源：IEEE 1175.3-2004]

3.1199

数据关键字 DataKey

特定数据组的标识符。

[来源：IEEE 1175.3-2004]

3.1200

数据部分 DataPart

结构化数据类型的组件，其域通过引用某些其他数据类型来指定。

[来源：IEEE 1175.3-2004]

3.1201

数据角色 DataRole

实体数据类型参与关系的方式的特征。

[来源：IEEE 1175.3-2004]

3.1202

数据敏感故障 data-sensitive fault

模式敏感故障 pattern-sensitive fault

一种在特定数据模式下引发失效的故障。

参见：程序敏感故障 (program-sensitive fault)。

[来源：ISO/IEC/IEEE 24765:2017, ?]

3.1203

面向数据结构的设计 data-structure-oriented design

以程序规格说明中的逻辑数据结构为基础，用于业务应用的设计方法。

示例：Jackson System Design 和 Warnier-Orr 方法。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1204

数据视图 DataView

将超类型数据类型实体划分为子类型实体。

[来源：IEEE 1175.3-2004]

3.1205

DDR2 SDRAM DDR2 SDRAM

双倍数据速率同步动态随机存取存储器单元，性能比 DDR SDRAM 更高，因该器件在一个内部时钟周期内传输四次数据（连续 4 个字）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1214]

3.1206

DDR3 DDR3

双倍数据速率 3 (DDR3)，其数据传输速度为 DDR 的 2 的 3 次方（即 DDR 的 8 倍）。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1207

DDR4 DDR4

双倍数据速率 4：数据传输速度为同步动态随机存取存储器（SDRAM）的 2 的 4 次方（即 16 倍）。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1208

停用/解除激活 deactivation

对集群执行检查点操作，再删除该集群。

[来源：ISO/IEC 10746-3:2009, 8.1.23]

3.1209

死锁 deadlock

由于两个或多个设备或进程，每一个等待赋给另一个的资源引起计算机处理停顿的情况。

参见：锁定（lockout）。

3.1210

反汇编器 deassembler

反汇编排程序。

见：XXX 条 反汇编程序（disassembler）。

[来源：GB/T 11457-2006]

3.1211

解块 deblock

将块的各个部分分开。

参见：块（block）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1222]

3.1212

调试 debugging

检测、定位和改正计算机程序中的故障。

注：技术包括使用断点、桌面检查、转储、审查、逆向执行、单步操作和追踪。

[来源：ISO/IEC 2382:2015, X.X]

3.1213

判定 decision

在两种或者多种可能的结果中作出选择，并决定动作集合的语句类型。

注：典型的判定有简单选择（例如，if-then-else），决定什么时候退出循环（例如，while-loop），以及多路开关语句（例如，case-1-2-3-...-n）。

[来源：GB/T 38634.1-2020, 3.8]

3.1214

判定准则 decision criteria

用于确定是否需要采取措施和进一步调查，或者用于描述给定结果置信度的阈值、目标或模式。

[来源：ISO/IEC 15939:2007, 3.7]

3.1215

判定门 decision gate

批准事件。

注：通常与评审会议关联。每个决策门都建立了准入和退出标准；能否通过决策门继续推进取决于决策者的同意。

[来源：INCONE 系统工程手册，第五版.]

3.1216

判定结果 decision outcome

判定式的结果，可用于决定控制流选择方向。

[来源：GB/T 38634.4-2020，4.14]

3.1217

决策策略 decision policy

用于确定生物特征比对结果为匹配或不匹配的一条或多条规则。

注：决策策略通常包含一个阈值，当比对分数超过该阈值时即视为匹配。

[来源：ISO/IEC TR 29119-13:2022，3.15]

3.1218

判定规则 decision rule

在判定表测试和因果图中产生特定结果的条件（也称为原因）和动作（也称为结果）的组合。

[来源：GB/T 38634.4-2020，4.15]

3.1219

判定表 decision table

以表格形式表示原因（输入描述为布尔条件）和结果（输出描述为布尔表达式）之间判定规则的表格。

[来源：ISO/IEC/IEEE 29119-4:2021，3.23]

3.1220

判定表测试 decision table testing

基于规格说明的测试用例设计技术，通过执行判定表中的判定规则来实现。

[来源：ISO/IEC/IEEE 29119-4:2021，3.24]

3.1221

判定测试 decision testing

基于结构化的测试用例设计技术，通过在测试项的控制流程中执行判定结果来实现。

[来源：ISO/IEC/IEEE 29119-1:2022，3.28]

3.1222

判定树 decision tree

监督学习模型，其推理过程可通过遍历一个或多个树状结构来表示。

[来源：ISO/IEC TR 29119-11:2020，3.1.25]

3.1223

判定树分析 decision tree analysis

一种用于在不确定条件下评估多选项决策影响的图表与计算方法。

[来源：项目管理知识体系指南（PMBOK Guide）第七版]

3.1224

判定变量 decision variable

判定者可选择的不同判定值的表示。

注：例如，在经济寿命计算中，判定变量是资产的持有时间。

[来源：ISO/IEC/IEEE FDIS 24765，3.1235]

3.1225

判定技术 decision-making technique

用于从不同备选方案中选择行动方案的一种技术。

[来源：项目管理知识体系指南（PMBOK® 指南）第六版]

3.1226

声明 declaration

不具备执行性的程序语句，其功能在于影响汇编器或编译器对程序中其他语句的解析方式。

注：声明的本质是，凭借声明行为本身和客体或其委托人的权力，导致在声明客体之外发生事态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1236]

3.1227

声明语言 declarative language

一非过程语言，它允许声明一组事实并表达使用这些事实的疑问或问题。

参见：XXX 条 交互的语言（interactive language）和 XXX 条 基于规则的语言（rule-based language）。

3.1228

反编译 decompile

将已编译的计算机程序从其机器语言版本转换为一种形式，该形式与原始高级语言程序相似，但不一定完全相同。

参见：编译

3.1229

反编译程序 decompiler

反编译计算机程序的软件工具。

3.1230

分解器 decomposer

一种组件，将高层级关键词跨各层级分解为最终构成它的低层级关键词序列。

[来源：ISO/IEC/IEEE 29119-5:2024, 3.4]

3.1231

分解 decomposition

将建模功能划分为其组成功能。

参见：组合。

3.1232

分解图 decomposition diagram

详细说明其父框的图表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1242]

3.1233

解耦 decoupling

使软件模块之间更加独立，以降低单个模块发生变更或错误时带来的影响。

参见：耦合

3.1234

深度学习 deep learning

通过训练具有一个或多个隐藏层的神经网络来创建丰富的层次化表示的方法。

注：深度学习使用由简单计算单元（或神经元）组成的多层网络。在这些神经网络中，每个单元组合一组输入值以产生一个输出值，该输出值随后传递给下游的其他神经元。

[来源：ISO/IEC TR 29119-11:2020, 3.1.26]

3.1235

深度神经网络 deep neural net; DNN

具有两层以上的神经网络。

[来源: ISO/IEC TR 29119-11:2020, 3.1.27]

3.1236

缺陷 defect

工作产品中存在的缺陷或不足, 导致其不符合要求或规格说明, 需要修复或更换。

参见: 故障。

示例: 1) 在生存周期早期阶段发现的遗漏和缺陷; 2) 软件中存在足以用于测试或运行的故障。

[来源: ISO/IEC 23531:2020, 3.1]

3.1237

缺陷密度 defect density

产品单位规模中的缺陷数量。

示例: 每千行代码的问题报告数。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1247]

3.1238

缺陷预防 defect prevention

与标识缺陷或潜在缺陷并预防将它们引入产品有关的活动。

[来源: GB/T 11457-2006]

3.1239

缺陷修复 defect repair

为修改不符合要求的产品或产品组成部分而进行的一种有意活动。

[来源: 项目管理知识体系指南 (PMBOK Guide) --第五版]

3.1240

缺陷根源 defect root cause

致使缺陷得以引入的根本原因 (例如缺乏有效的过程)。

3.1241

防御性编程 defensive programming

一种通用的编程方法, 它假定在初始开发和维护过程中都会出现错误, 因此, 它编写的代码使得程序在出现错误时仍然能够正常运行。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1248]

3.1242

定义活动 Define Activities

识别和记录为完成项目可交付成果而须采取的具体行动的过程。

[来源: 项目管理知识体系指南 (PMBOK® 指南) --第六版]

3.1243

定义范围 Define Scope

制定项目和产品的详细描述的过程。

[来源: 项目管理知识体系指南 (PMBOK® 指南) --第六版]

3.1244

已定义过程 defined process

依据组织的裁剪指南, 对组织标准过程集进行管理和裁剪后所实施的过程。

注：已定义过程具备文档化并受维护的过程描述，并向组织的过程资产库贡献工作产品、度量数据及其他过程改进信息。项目的已定义过程为规划、执行和改进项目各项任务与活动提供了基础。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1249]

3.1245

定义性模型 defining model; MOF

某一存储库基于其中的模型。任何数目的存储库都能有同一定义性模型。

3.1246

定义使用对 definition-use pair

数据定义和后续谓词的数据使用或计算的数据使用，其中数据使用是使用数据定义中定义的值。

[来源：GB/T 38634.4-2020, 4.16]

3.1247

定义使用路径 definition-use path

从变量定义到谓词使用（p-use）或计算使用（c-use）的控制流子路径。

[来源：GB/T 38634.4-2020, 4.17]

3.1248

最终主版本 definitive master version

用于安装软件并创建分发副本的软件原始实例。

注：“安装”既可适用于可执行软件，也可适用于不可执行软件，或相关资产（如字体）。该术语既可指客户端/本地设备上的安装，也可指服务器端安装，例如作为服务型软件资产提供的一部分。

[来源：ISO/IEC 19770-1:2012, 3.4]

3.1249

最终软件库 definitive software library; DSL

由物理介质或一个或多个电子软件仓库构成的安全存储环境，能够对所有软件配置项的最终授权版本以及由 SAM 管控的所有软件母版进行控制和保护。

[来源：ISO/IEC 19770-5:2015, 3.11]

3.1250

影响程度 degree of influence; DI

表示 19 个（或更多）技术复杂度调整系数中关于每个系数的影响的数字指标。

注：其范围从 0（无影响）到 5（强烈影响），用于技术复杂度调整的计算。

[来源：GB/T 42566-2023, 3.10]

3.1251

去标识化 de-identification

从数据集中删除个人身份信息。

[来源：IEEE 7002:2022, 3.1]

3.1252

委派 delegation

将授权、职责或服务提供分配给其他对象的行为。

注：委派作出，可随时撤销。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1253]

3.1253

已删除的源语句 deleted source statement

在构建新产品的过程中，从现有软件产品中删除或修改的源语句。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1254]

3.1254

定界符 **delimiter**

用于标识一组相关比特、字符、字或语句起始位置与结束位置的单个字符或字符组。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1256]

3.1255

交付物 **deliverable**

(1) 为完成某个过程、阶段或项目所必须产出的、任何独特的且可验证的产品、成果或服务能力。

(2) 根据协议规定，需提供给采购方或其他指定接收方的标的物。

参见：采购方、产品、成果。

注：该标的物可以是文档、硬件项、软件项、服务或任何类型的工作产品。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1258]

3.1256

可交付产品 **deliverable product**

经项目发起人或客户批准，提供服务的任何特有的、可验证的系统或软件产品。

[来源：GB/T 25000.41-2018, 4.1]

3.1257

交付源码声明 **delivered source statement**

被整合到交付给客户的最终产品中的源代码声明。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1260]

3.1258

交付 **delivery**

向客户或预期用户交付系统或组件。参见：软件生存周期 (software life cycle)、系统生存周期 (system life cycle)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1261]

3.1259

交付绩效域 **delivery performance domain**

关注活动和职能的绩效域，这些活动和职能与项目既定范围和质量的交付相关。

[来源：项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.1260

德尔菲法 **Delphi method**

用于就某一主题使专家达成共识的信息收集技术。

注：主持人使用问卷征集与该主题相关的重要项目要点的观点。对回复进行总结，然后再次分发给专家以供进一步评论。通过若干轮这样的过程，可以达成共识。

[来源：ISO/IEC TR 7052:2023, 3.16]

3.1261

差量 **delta**

两个版本的区别。

[来源：ISO/IEC TR 18018:2010, 3.11]

3.1262

按需调页 **demand paging**

仅在页面需要时，才将其从辅助存储器调入主存储器的存储分配技术。

参见：预先分页。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1265]

3.1263

简并/去模块化 demodularization

在软件设计中合并相关软件模块、通常用以优化系统性能的过程。

参见：向下压缩、横向压缩、向上压缩。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1267]

3.1264

演示 demonstration

一种依赖于在系统或组件执行期间观察其行为的动态分析技术，无需进行事后分析，即可检测错误、违反开发标准的问题及其他缺陷。

参见：测试（testing）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1268]

3.1265

依赖性 dependability

系统在需要时按要求完成规定功能的能力。

注：依赖性包括可用性、可靠性、韧性、维护性；部分场景下还包含耐久性、安全性、保密性等特性。依赖性是产品与时间相关质量特性的集合。

[来源：IEEE 7000:2021, 3.1]

3.1266

可信性测量 dependability measurement

收集和处理数据，以得出系统的可靠性、可用性、可支持性或可恢复性的数值。

[来源：IEEE 982-2024, 3.1]

3.1267

可信性要求 dependability requirement

要求陈述，为系统的一个或多个可靠性、可用性、可支持性或可恢复性参数规定有界阈值。

参见：能力要求。

[来源：IEEE 982-2024, 3.1]

3.1268

依赖性确定 dependency determination

一种用于识别所采用的依赖类型的技术，该依赖类型用于建立前置活动与后续活动之间的逻辑关系。

[来源：项目管理知识体系指南（PMBOK® 指南）——第五版]

3.1269

依赖项导出 dependency export

将部件及其全部依赖项从配置管理系统以单一流程导出的操作。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.10]

3.1270

依赖实体 dependent entity

标识符依赖实体 identifier-dependent entity

实例的唯一标识依赖于与另一实体间关系的实体。

参见：独立实体。

注：以外键表述，若一个实体的全部外键都完整包含在其主键中，则称该实体为依赖实体。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1273]

3.1271

依赖状态类 dependent state class

实例本身在本质上与其他某些状态类实例存在固有关联的类。

参见：独立状态类。

注：依赖状态类实例不宜孤立存在且与其他类实例无关联；此外，变更其所关联的实例亦无实际意义。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1274]

3.1272

依赖 dependency

两个建模元素之间的一种关系，对其中一个建模元素（独立元素）的更改，将影响另一建模元素（依赖元素）。

3.1273

部署 deployment

在系统分解的同时进行需求指派。

[来源：GB/T 25000.30-2021, 1.509]

3.1274

部署图 deployment diagram

一种图，用于显示运行的处理结点以及居于其上的构件、进程和对象的配置的图。构件表示代码单元在运行时的体现。

参见：XXX 条 构件图（component diagram）。

3.1275

部署包 deployment package; DP

为便于在极小型实体中实施所选框架的一系列实践而开发的工件集。

[来源：ISO/IEC 29110-1-2:2024, 3.35]

3.1276

需求的导出 derivation of requirements

在同一系统级将需求从一种类型的需求转换为另一种需求的步骤。

注 1：需求的类型包括使用质量需求、产品质量需求和数据质量需求。

注 2：该词条为“导出（derivation）”解释说明[来源：GB/T 25000.30-2021 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第 30 部分：质量需求框架]。

[来源：GB/T 25000.30-2021, X.X]

3.1277

派生类 derived class

实例 A 的模板类 CA 与实例 B 的模板类 CB 之间的关系，其中模板 A 是对模板 B 的增量修改。参见：基类。

[来源：ISO/IEC 24765: 2025, 3.1278]

3.1278

派生数据 derived data

除了直接从数据函数中检索和验证信息之外，还包括其他步骤过程中所产生的数据。

[来源：GB/T 42449-2023, 3.17]

3.1279

导出元素 derived element

一种能从另一元素算出的如下模型元素：为明晰起见加以展现，或为设计目的予以纳入，即便它并未添加语义信息。

3.1280

导出测度 derived measure

由至少两个基本测度值的函数定义的测度。

注：改写 2012 年版国际计量学词汇—基本和通用概念及相关术语中“导出量”的定义。

[来源：ISO/IEC/IEEE 15939:2017]

3.1281

派生性质 derived property

派生属性 derived attribute

派生参与者项 derived participant party

用于指代其值由计算确定的性质的定名。

注：派生性质的典型情形为派生属性，但不禁止存在其他类型的派生性质。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1281]

3.1282

派生需求 derived requirement

从集合和组织中推导或推断出一个特定系统技术状态和解决方案的需求。

注 1：更高层级的需求称为“父级”需求，而从该父级派生的需求称为“子级”需求。

注 2：派生需求通常是在利益相关方需求获取、需求分析、权衡研究或确认期间识别的。

[来源：GB/T 45802-2025 系统与软件工程 生存周期过程 需求工程, 3.1.10]

3.1283

派生类型 derived type

一种数据类型，它的成员和操作来自其他数据类型，按照某种特定的规则得到的。

参见：2.1623 条 子类型 (subtype)。

[来源：GB/T 11457-2006, 2.437]

3.1284

描述 description

表示计划中或实际存在的概念、功能、设计或对象的信息项。

[来源：ISO/IEC 24765: 2025, 3.1286]

3.1285

描述标准 description standard

用以规定产品信息或规程特征，为产品的理解、测试、安装、使用或维护提供依据的标准。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1287]

3.1286

描述模型 descriptive model

显示一组相互关联的模型元素的模型，这些模型元素表征了系统的关键方面，包括其结构、行为、参数和需求。

注：描述模型，是对其系统分析模型和设计模型的补充。

[来源：GB/T 45615-2025, 3.1.6; ISO/IEC/IEEE 24641:2023, 3.1.6]

3.1287

设计 design

（动词）〈过程〉界定架构、系统元素、接口以及某一系统或系统元素的其他特性。

注1：包括系统元素及其相互关系的规范，充分完备足以支持架构兼容实现的信息。

注2：设计提供了系统元素的详尽的实施级的物理结构、行为、时间关系和其他属性。

（名词）设计（4.1.15）过程的结果。

[来源：GB/T 22032-2021 系统与软件工程 系统生存周期过程，定义 4.1.15]

3.1288

设计分析器 design analyser

自动化设计工具，接收关于程序设计的信息，并生成诸如模块层次结构图、控制和数据结构的图形化表示以及访问的数据块清单等输出。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1289

设计分析 design analysis

对一设计进行估计以确定其相对于预定需求的正确性、符合设计标准的程度、系统效率和是否符合其他一些准则。

对其他替代性设计途径的估计。

3.1290

设计属性 design attribute

设计视图中的一个元素，用以命名设计实体、设计关系或设计约束的特性或属性。

参见：设计约束、设计实体、设计关系。

[来源：ISO/IEC 24765: 2025, 3.1291]

3.1291

设计特性 design characteristic

属于一个产品或服务的可测量描述的设计属性或特有属性。[来源：GB/T 22032-2021 系统与软件工程 系统生存周期过程，定义 4.1.17]。

3.1292

设计概念 design concept

可用于指导系统设计的基础性思想。

示例：信息隐藏。

[来源：ISO/IEC 24765: 2025, 3.1293]

3.1293

设计关注点 design concern

设计关注点是“在软件设计中关注的领域”，它与一个或多个利益相关方相关。

[来源：ISO/IEC/IEEE 24765:2017, X.X]。

3.1294

设计约束 design constraint

为所关注系统设定的、内部或外部施加的边界条件，组织在概念和开发阶段执行各过程时必须在此界限内运作。

参见：设计属性、设计实体、设计关系。

示例：物理要求、性能要求、软件开发标准、软件质量保证标准。

注：设计约束用于声明某一设计思想是强制性的还是应被排除的。

[来源：ISO/IEC 24765: 2025, 3.1294]

3.1295

设计描述 design description

设计文档 design document

设计规格说明 design specification

一种文档，它描述系统或部件的设计、典型内容包括系统或部件体系结构、控制逻辑、数据结构、输入输出格式、接口描述和算法。

参见：产品规格说明、需求规格说明、设计文档、设计规格说明

[来源：GB/T 11457-2006, 2.437]

3.1296

设计文档 design document

设计规格说明 design specification

一种文档，它描述系统或部件的设计、典型内容包括系统或部件体系结构、控制逻辑、数据结构、输入输出格式、接口描述和算法。

参见：产品规格说明、需求规格说明

[来源：GB/T 11457-2006, X.X]

3.1297

设计元素 design element

设计中的基本部件或构件块。

[来源：GB/T 11457-2006, 2.444]

3.1298

设计实体 design entity

设计的元素（部件），它从结构上和功能上与其他元素相区别并分别命名和引用。

[来源：GB/T 11457-2006, 2.445]

3.1299

设计缺陷 design fault

在系统设计过程中，由人为错误导致的设计故障（源自规格说明、编码），该故障可能导致设计失效。

[来源：ISO/IEC 24765: 2025, 3.1296]

3.1300

设计语言 design language

一种具有专门构造，有时还可验证的规格说明语言。用以开发、分析和说明硬件或软件设计。类型包括硬件设计语言、程序设计语言。

参见：需求规格说明语言

[来源：GB/T 11457-2006, 2.447]

3.1301

设计级别 design level

软件项的设计分解（例如，系统、子系统、程序或模块）。

[来源：GB/T 11457-2006, 2.448]

3.1302

设计方法学 design methodology

进行设计的系统途径。由专门选择的工具、技术、准则的有序应用所构成。

[来源: GB/T 11457-2006, 2.449]

3.1303

实验设计 design of experiments

一种统计方法, 用来识别哪些因素会对正在生产的产品或正在开发的流程的特定变量产生影响。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) —第五版]

3.1304

设计叠加 design overlay

一种附加、详细或派生的设计信息表示形式, 该信息以现有设计视图为参照进行组织。

参见: 设计视图

[来源: IEEE 1016-2009, 3.6]

3.1305

设计模式 design pattern

对问题及其解决方案本质的描述, 旨在使该方案能够在不同情境下被复用。

注: 并非详细的规格说明, 而是对积累的知识与经验的描述。

[来源: ISO/IEC 24765: 2025, 3.1300]

3.1306

设计阶段 design phase

软件生存周期中的一段时间。在这段时间内, 进行体系结构、软件组成部分、接口和数据的设计, 为设计编制文件, 并对其进行验证, 以满足预定需求。

参见: 详细设计、概要设计

[来源: GB/T 11457-2006, 2.450]

3.1307

设计原理 design rationale

设计系统所依据的推理过程的信息, 包括设计备选方案、所考虑的权衡因素、做出的决策以及这些决策的论证依据。

[来源: ISO/IEC 24765: 2025, 3.1302]

3.1308

设计关系 design relationship

命名设计实体之间的连接或对应关系的设计视图元素 [IEEE 1016-2009 IEEE 信息技术标准-系统设计-软件设计说明, 3.8] 参见: 设计实体。

3.1309

设计需求 design requirement

规定或强制系统或系统部件的设计的需求。

参见: 功能需求、实现需求、接口需求、性能需求、物理需求

[来源: GB/T 11457-2006, 定义 2.451]

3.1310

设计评审 design review

把系统、硬件或软件设计提交给项目人员、管理者、用户、客户或有关人士供其评审或批准的过程或会议。类型包括关键设计评审、概要设计评审和系统设计评审。

对现有的或提出的设计所做的正式评估和审查, 其目的是找出可能会影响产品, 过程或服务工作的适用性和环境方面的设计缺陷并采取补救措施, 以及(或者)找出在性能、安全性和经济方面的可能的改进。

参见：代码评审、形式合格评审、需求评审、测试就绪评审

[来源：GB/T 11457-2006，定义 2.452]

3.1311

设计标准 design standard

一种标准，它描述设计或数据或程序部件的设计描述的特性。

[来源：GB/T 11457-2006，定义 2.454]

3.1312

设计策略 design strategy

执行设计的总体计划与方向。

示例：功能分解。

[来源：ISO/IEC 24765: 2025, 3.1306]

3.1313

设计对象 design subject

将为其编写软件设计说明书（SDD）的软件项或系统。

同义词：待设计软件、待设计系统。

[来源：IEEE Std 1016-2009]

3.1314

设计思维 design thinking

解决由人类需要所定义的复杂问题的方法论。

注：通过头脑风暴会议形成解决方案，并在此过程中生成在实践中测试预期解决方案的原型。

[来源：ISO/IEC TR 7052:2023, 3.17]

3.1315

面向成本的设计 Design to cost; DTC

管理系统或软件项目的一种方法，旨在将项目成本控制在预定范围内。

注 1：需密切跟踪实际成本和预计成本，如果成本可能超过目标，则采取诸如删除或推迟较低优先级需求等措施。

同义词：作为独立变量的成本（cost as an independent variable），design-to-cost。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1311]

3.1316

设计可追溯性 Design traceability

设计特性、设计使能要素与系统要素需求之间的双向可追溯性。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.1317

设计单元 design unit

设计元素的逻辑相关的集合。在 Ada PDL 中，设计单元由 Ada 编译单元表示。

[来源：GB/T 11457-2006，定义 2.455]

3.1318

设计视图 design view

特别适合于软件项目活动需要的设计实体属性信息的子集。

[来源：GB/T 11457-2006，定义 2.456]

3.1319

设计视点 design viewpoint

用于构建和使用设计视图的可用元素及约定的规约。参见：设计视图。

[来源: ISO/IEC 24765: 2025, 3.1310]

3.1320

按成本设计（面向成本的设计） **design-to-cost; DTC**

同义词： 成本作为独立变量、按成本设计。

一种管理系统或软件项目的方法，旨在将项目成本控制在预定范围之内。

注：该方法会密切跟踪实际成本与预测成本，若成本可能超出目标，则采取如删除或推迟低优先级需求等措施。

[来源: ISO/IEC 24765: 2025, 3.1311]

3.1321

期望结果 **desirable consequence**

同义词： 积极结果。

与收益、获利或避免不良后果相关的结果。

[来源: ISO/IEC 24765: 2025, 3.1312]

3.1322

预期结果 **desirable consequence**

积极结果 **positive consequence**

与获益、收益或规避不利后果相关联的后果。

[来源: ISO/IEC/IEEE 15026-1:2019, 3.4.4]

3.1323

桌面检查 **desk checking**

<信息技术>手动模拟程序执行，通过逐步检查源程序的功能或语法错误来检测故障。

<软件与系统工程>静态分析技术，通常由代码清单、测试结果或其他文档的生成者进行目视检查，以识别错误、违反开发标准的行为或其他问题。

参见：审查、走查。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.1324

桌面出版 **desktop publishing; DTP**

使用微型计算机生存电子出版物。

[来源: ISO/IEC 2382:2015, X.X]

3.1325

目的地址 **destination address**

数据拟传输至的设备或存储位置地址。

参见：源地址。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1315]

3.1326

破坏性读 **destructive read**

改变所访问位置数据的读取操作。

参见：非破坏性读。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1316]

3.1327

详细设计描述 **detailed design description**

同义词： 详细设计规约。

描述计算机程序确切详细配置的文档。

注：该文档标识软件产品各底层组件的输入、输出、控制逻辑、算法及数据结构，是详细设计阶段的主要产物。

[来源：ISO/IEC 24765: 2025, 3.1319]

3.1328

详细设计描述 detailed design description

详细设计规格说明 detailed design specification

描述计算机程序精确详尽配置的文档。

注：该文档明确标识软件产品中每个底层部件的输入、输出、控制逻辑、算法及数据结构，是详细设计阶段的主要产出物。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1319]

3.1329

详细设计 detailed design description; detailed design specification

对系统或组件的初步设计进行细化和扩展，直至设计足够完整，可以进行实施。

参见：底层设计、初步设计、软件开发过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1318]

3.1330

详细设计阶段 detailed design phase

软件开发生存周期中的一个阶段，在此阶段执行详细设计过程；其利用前一阶段（架构设计）产生的软件系统设计与软件架构，为每个单元产出可供编码的详细逻辑。

[来源：ISO/IEC 24765: 2025, 3.1320]

3.1331

详细设计评审 detailed design review

一项里程碑评审，旨在判定详细软件设计（如详细设计描述中所描绘）在满足软件需求文档要求方面的可接受性。

[来源：ISO/IEC 24765: 2025, 3.1321]

3.1332

详细功能点分析 detailed function point analysis

确定应用程序或者项目规模的最精确的计数，在这样的应用程序或者项目中，FPA 所需规约都已经非常详细。

注：这意味着事务处理已经被定义到引用逻辑文件（也就是所谓的文件类型参考）和数据元素类型的级别，并且逻辑文件已经被定义到记录类型和数据元素类型的级别。至此，每个被认可的功能的复杂度才能够被确立起来。

本文件中描述的详细功能点分析在计数中是最准确的类型，另外两种技术是估算功能点分析和预估功能点分析。

[来源：GB/T 42588-2023, 3.1.15]

3.1333

详细功能点计数 detailed function point count

在进行功能点分析所需的全部规格说明均已详细明确的情况下，用于确定应用或项目规模的最准确的计数方法。

注：这意味着事务功能已细化到被引用的逻辑文件层级（即所谓“被引用文件类型”）和数据元素类型，并且逻辑文件已细化到记录类型和数据元素类型层级。因此，可以确定每一个已识别功能的复杂度。

[来源：ISO/IEC 24570:2005]

3.1334

检测误差权衡 detection error trade-off; DET; DET

随着判别阈值的变化，二元分类系统的漏报与误报错误之间的关系。

注：DET 可用 DET 表或 DET 图表示。

[来源：ISO/IEC TR 29119-13:2022, 3.1.16]

3.1335

制定预算 Determine Budget

汇总所有单个活动或工作包的估算成本，建立一个经批准的成本基准的过程。

[来源：PMBOK® 指南第五版]

3.1336

确定性系统 deterministic system

对于给定的一组特定的输入和初始状态，始终产生相同输出和最终状态的系统。

[来源：ISO/IEC TR 29119-11:2020, 3.1.28]

3.1337

制定项目章程 develop project charter

编写一份正式批准项目并授权项目经理在项目活动中使用组织资源的文件的过程。

[来源：PMBOK® 指南第五版]

3.1338

制定项目管理计划 develop project management plan

定义、准备和协调所有子计划，并把它们整合为一份综合项目管理计划的过程。

[来源：PMBOK® 指南第五版]

3.1339

制定进度计划 develop schedule

分析活动顺序、持续时间、资源需求和进度制约因素，创建项目进度模型的过程。

[来源：PMBOK® 指南第五版]

3.1340

开发源语句 developed source statement

为软件产品新创建、新增或修改的源代码语句。

[来源：ISO/IEC 24765: 2025, 3.1324]

3.1341

开发 development

一个新的应用程序或基于现有应用程序进行独立增加的规范、构造、测试和交付。

[来源：GB/T 42566-2023 系统与软件工程 功能规模测量 MkII 功能点分析方法, 3.11]

3.1342

开发方法 development approach

在项目生存周期中用于创建和改进产品、服务或成果的方法，例如预测性方法、自适应方法或混合方法。

[PMBOK®指南 第七版]另见生存周期模型。

3.1343

开发方法与生存周期绩效域 development approach and life cycle performance domain

关注活动和职能的绩效域，这些活动和职能与项目的开发方法、节奏以及项目生存周期阶段相关。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1344

开发分支 development branch

主要产品开发分支。

注：从开发分支构建的产品将具备最新功能，但可能仍处于不成熟阶段。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3.1345

开发环境 development environment

面向计算机解决方案设计师与开发者的硬件、软件、平台及工具。

注：该设施配备的人员配置、资源、接口、功能及工具，是目标系统环境中所不具备的。

[来源：ISO/IEC/IEEE 24765:2014, X. X]

3.1346

开发方法学 development methodology

编制软件的系统方法。它确定开发的各个阶段，规定每一阶段的活动、产品、验证步骤和完成准则。

3.1347

开发流水线 development pipeline,

构建流水线 build pipeline

旨在管理和控制封包构建的工具集，该封包使得应用能够交付并投入使用。

[来源：ISO/IEC TR 7052:2023, 3.19]

3.1348

开发计划 development plan

指导、实施和控制一个或多个产品或服务的设计与开发的计划。 参见项目计划。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3.1349

开发过程 development process

在软件开发期间，为给定目的（例如构造模型或实现模型）而进行的一组部分有序的步骤。

3.1350

开发项目 development project

用于测量在项目完成后首次安装所开发的软件时，提供给最终用户的功能点计数。[来源：GB/T 42566-2023 系统与软件工程 功能规模测量 MkII 功能点分析方法]。

3.1351

开发项目功能点计数 development project function point count ; DFP

用于测量在项目完成后首次安装所开发的软件时，提供给最终用户的功能点计数。

[来源：GB/T 42566-2023 系统与软件工程 功能规模测量 MkII 功能点分析方法, 3.20]。

3.1352

开发项目功能规模 development project functional size

通过开发项目功能点计数衡量软件首次发布时向用户提供的功能。

注：开发项目的功能尺寸可包含转换功能的尺寸。

[来源：ISO/IEC 20926:2009, 3.19]

3.1353

开发团队 development team

开发或维护软件的团队。

注：采购方、产品负责人以及未来的维护方可以构成开发团队的一部分。开发团队可以按职能划分（例如：设计团队、程序员团队、测试团队），也可以是跨学科的（每个团队都具备设计、编程与测试专长）。

[来源：ISO/IEC TR 7052:2023, 3.20]

3.1354

开发测试 development testing

为确定新软件产品或基于软件的系统（或其组件）是否满足其标准而进行的测试。

参见：验收测试，运行测试，合格性测试。

注：标准将根据所进行的测试级别而有所不同。

[来源：ISO/IEC FDIS 24765, 3.1337]

3.1355

开发工具 development tool

用于开发或修改应用程序的硬件和软件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1338]

3.1356

开发基线 developmental baseline

在特定时间点对正在开发的系统有效的规范。

3.1357 [[来源：ISO/IEC 2382:2015, X.X]

开发配置 developmental configuration

在配置管理中，开发期间用于界定软件配置项演进构型的软件及配套技术文档。

参见：分配的基线、功能基线、产品基线。

注：开发配置由开发人员控制，因此不称为基线。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1340]

3.1358

开发配置管理 developmental configuration management

为指定和控制软件和相关的技术文档，来定义开发期间处在不断进化中的软件工作产品的配置，而运用技术手段和行政的管理手段。开发配置管理处处在开发者的直接控制之下。置于开发配置管理之下的配置项不是基线，虽然在开发过程的某些点上，它们可能被基线化并置于基线配置管理之下。

3.1359

偏离 deviation

〈软件与系统工程〉脱离规定的需求。

〈制造业〉在制造某项产品前预先授予的书面授权，允许针对特定数量的产品或特定时间段，与某项特定的性能或设计要求有差异。

[来源：ISO/IEC FDIS 24765, 3.1341, 有修改]

3.1360

设备接口任务 device interface task

封装外部 I/O 设备特性和接口的并发任务。

[来源：ISO/IEC FDIS 24765, 3.1343]

3.1361

开发运维 DevOps

使利益相关方能够更好地沟通协作，以实现软件和系统产品及服务的规定、开发和运维，并持续改进其生存周期各方面的一套原则和实践。

[来源：IEEE 2675-2021, 3.1]

3.1362

诊断 diagnostic

用于说明故障或失效的检测和隔离。

示例：诊断消息、诊断手册。

3.1363

诊断手册 diagnostic manual

提供系统或部件诊断流程执行、故障识别以及故障排除所需必要信息的文档。

参见：安装手册、操作手册、编程手册、支持手册、用户手册。

注：通常说明系统或部件的诊断功能，以及用于其保障维护的诊断工具。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1348]

3.1364

对角微指令 diagonal microinstruction

能够规定执行一条机器语言指令所需若干同步操作的微指令。

参见：水平微指令、垂直微指令。

注：对角微指令在长度与功能规模上，介于水平微指令与垂直微指令之间。“对角”这一命名，是指代此种折中设计特征，而非微指令的任何物理形态特征。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1349]

3.1365

图边界 diagram boundary

图页中的图边缘。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1350]

3.1366

图形特征 diagram feature

图的组成要素。

注：图形特征包括方框、箭线段、箭头标注、ICOM 编码、ICOM 标注、模型注释以及阅读注释。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1351]

3.1367

图形特征引用 diagram feature reference

在 IDEF0 模型中唯一标识图形特征的表达式。

[来源：ISO/IEC/IEEE 24765:2017, 3.1196]

3.1368

图号 diagram number

图引用中对应图的父功能节点编号的部分。

注：图号用于指代对同一节点编号所标识的功能进行细化或分解的图示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1352]

3.1369

图页 diagram page

包含上下文图或分解图的模型页面。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1353]

3.1370

图引用 diagram reference

唯一标识一张图示，并规定其在特定模型层级中位置的表达式。

注：图引用由模型名称、缩写及图号构成。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1354]

3.1371

图表标题 diagram title

描述图示所表达整体功能的动词或动词短语。

注：子图的图标题为其父方框的方框名称。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1355]

3.1372

图解技术 diagramming techniques

用逻辑链接来呈现信息以辅助理解的方法。

[来源：项目管理知识体系指南（PMBOK® 指南） 第六版]

3.1373

对话 dialog

用户和应用程序之间进行的，执行事务处理必须的信息交互。

[来源：GB/T 42588-2023, 3.1.18]

3.1374

差额现金流量 differential cash flow

两个备选方案或建议的现金流量之差的表示。

参见：增量分析（incremental analysis）。

注：通常采用内部收益率（IRR）作为比较基础进行分析。

[来源：ISO/IEC FDIS 24765, 3.1358]

3.1375

数字字符 digit numeric character; numeric character

数字字符 numeric character

表示非负整数的字符。

示例：十六进制记数制中 0、……、F 中的任一字符。

[来源：ISO/IEC 2382:2015, X.X]

3.1376

数字资产 digital asset

以数字格式电子化表达的 IT 资产。

示例：软件资产和数字信息内容资产。

[来源：ISO/IEC 19770-1:2017, 3.17]

3.1377

数字信息内容资产 digital information content asset

具有信息内容的数字资产。

示例：文档、音频、视频、图形、数据库、独立字典、数字形式的标准集合、媒体集合和信用机构评级信息。

注：通常获得许可，但不被视为软件。IT 资产管理可以包括对这些资产进行许可证合规性管理，但不包括对内容的管理。

[来源：ISO/IEC 19770-1:2017, 3.18]

3.1378

数字产品 digital product

以电子格式交付、使用和存储的产品或服务。

[来源：项目管理知识体系指南（PMBOK® 指南） — 第七版]

3.1379

数字信号处理 digital signal processing; DSP

数字信号处理器 digital signal processor

〈信息技术〉对信号进行数字化分析、运算、加工和使用的技术。

〈电子元器件〉以微处理器为核心的数字信号处理单元，是嵌入式系统的硬件实现之一。

[来源：GB/T 22033-2017，2.018、2.019]

3.1380

数字通用光盘 digital versatile disc.; DVD

使用 8-16 调制编码记录信息，以 650nm 波长的激光读出信息的光盘。

注：数据量分为 DVD5 (4.7GB), DVD10 (9.4GB), DVD9 (8.5GB), DVD18 (17GB) 等 4 种规格。

[来源：GB/T 33663-2017，2.6]

3.1381

维度 dimension

多维构造包含的独立组件。

[来源：ISO/IEC 33003:2015，3.6]

3.1382

直接地址 direct address

一级地址 one-level address

标识一操作数的存储单元的地址。

参见：立即数据(immediate data)，间接地址(indirect address)，n 级地址(n-level address)，直接指令(direct instruction)。

[来源：ISO/IEC FDIS 24765，3.1370，有修改]

3.1383

直接成本 direct cost

为达成软件项目目标而直接付出的各种资源代价总和。

注：如可直接计入软件项目成本的直接材料和直接人工等。

[来源：GB/T 36964:2018，1.65]

3.1384

直接指令 direct instruction

包含操作数直接地址的计算机指令。

参见：立即指令(immediate instruction)、间接指令(indirect instruction)、绝对指令(absolute instruction)、有效指令(effective instruction)。

[来源：ISO/IEC/IEEE FDIS 24765，3.1371，有修改]

3.1385

直接人力 direct labour

与生产单位直接相关的人员劳动。

参见：间接人力。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.1386

直接测量 direct measure

不依赖于任何其他属性的度量的属性度量。

[来源: GB/T 11457-2023, 2.484]

3.1387

直接内存访问 **direct memory access; DMA**

外围设备直接接管中央处理单元的内存总线, 以向内存写入或从内存读取数据的技术。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1374]

3.1388

直接内存访问控制器 **direct memory access controller; DMAC**

执行直接内存访问的功能单元。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1375]

3.1389

直接度量 **direct metric**

不依赖于任何其他属性度量的度量。[来源: IEEE 1061-1998 (R2004) IEEE 软件质量度量方法标准, 2.4]。

3.1390

直接测量值 **direct metric value**

最终产品中需满足的质量因素的数值目标。

示例: 平均无故障时间 (MTTF) 是最终系统可靠性的直接测量指标。

[来源: IEEE 1061-1998 (R2004), 2.5]

3.1391

直接工时 **direct staff-hour**

在创造特定输出产品时直接消耗的工作量。

[来源: ISO/IEC FDIS 24765, 3.1377]

3.1392

直接用户 **direct user**

直接与产品互动的人。

参见: 间接用户。

[来源: ISO/IEC 25019:2023, 3.1.14; ISO/IEC 25002:2024, 3.1.8]

3.1393

有向图 **directed graph; digraph**

节点间的连接隐含方向的图。

参见: 无向图 (undirected graph)。

[来源: ISO/IEC FDIS 24765, 3.1379, 有修改]

3.1394

有向系统之系统 **directed system of systems; directed SOS**

系统的系统 (SoS) 是为实现特定目标而创建和管理的, 其组成系统均从属于该 SoS。

注: 组件系统保持独立运行能力; 但其正常运行模式从属于中央管理目标。

[来源: ISO/IEC/IEEE 21841:2019, 3.2.3]。

3.1395

目录系统代理 **directory system agent; DSA**

一个开放系统互连应用进程, 是目录的一个组成部分。

[来源: GB/T 16264.2-2008, 6.1.10]

3.1396

目录用户代理 directory user agent; DUA

开放系统互连的一个应用进程，它在访问目录过程中代表某一个用户。

注：DUA 可能还会提供一个本地范围的便利工具以帮助用户构成请求并解释响应。

[来源：GB/T 16264.2-2008，6.10]

3.1397

反汇编 disassemble

将已汇编的计算机程序从其机器语言版本转换为一种类同（但未必完全等同于）原始汇编语言程序的形式。

参见：汇编（assemble）。

[来源：ISO/IEC/IEEE FDIS 24765，3.1382]

3.1398

反汇编程序 disassembler

对计算机程序进行反汇编的软件工具。

[来源：ISO/IEC/IEEE FDIS 24765，3.1383，有修改]

3.1399

灾难恢复 disaster recovery; DR

在计算机系统运行中，硬件或软件出现故障后恢复正常运行过程。

[来源：ISO/IEC/IEEE FDIS 24765，3.1384]

2. 组织的信息和通信技术组件，在发生灾难后规定时限内，将关键业务功能恢复至可接受支撑水平的能力。

[来源：ISO/IEC/IEEE 26511:2018，3.1.11]

3.1400

灾难恢复计划 disaster recovery plan; DRP

信息系统灾难恢复过程中筹划所需的任务、行动、数据和资源，用于指导相关人员在预定的灾难恢复目标内恢复信息系统所支持关键业务功能的文件。

[来源：GB/T 25069-2022，2.767]

3.1401

特定学科模型 discipline-specific model

从某一学科的角度，对系统或系统元素的描述，处理源于特定学科模型元素的领域特定的关注点。

[来源：GB/T 45615-2025，3.1.7；ISO/IEC/IEEE 24641:2023，3.1.7]

3.1402

折现回收期 discounted payback period

项目收回初始投资（含利息）所需时间。

注：反映风险敞口的指示。若项目在达到回收期前终止，企业将蒙受亏损。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.1403

离散数据 discrete data

按特定时间间隔到达的数据。

[来源：ISO/IEC/IEEE FDIS 24765，3.1389]

3.1404

独立型活动 discrete effort

独立、明确且与特定工作分解结构组件及可交付成果的完成相关联的活动，能够予以规划并测量。
参见：依附型活动（apportioned effort）。

注：用于计算工作绩效的三种挣值管理活动类型之一。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1390]

3.1405

离散型 discrete type

一种数据类型，它的成员能假定是任何不同值的集合。

注：离散类型可以是枚举类型或整数类型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1391, 有修改]

3.1406

自由依赖关系 discretionary dependency

基于最佳实践或项目偏好的关系。

[PMBOK®指南 第七版]。

3.1407

选择性依赖关系 discretionary dependency

优选逻辑 preferential logic

首选逻辑 preferred logic

软逻辑 soft logic

一种基于最佳实践或项目偏好的关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.192]

3.1408

辨别阈值 discrimination (threshold)

刺激缓慢且单调变化时，未引起测量仪器响应可检测变化的最大刺激变化量。

注：辨别阈值可能取决于噪声（内部或外部）、摩擦力等因素，也可能取决于刺激值本身。

[来源：ISO/IEC TR 14143-3:2003, 3.4]

3.1409

判别器 discriminator

类别判别器 category discriminator

超类的属性，与该超类的簇相关联，其值标识特定实例所属的子类。

注：由于判别器的值（当已声明判别器时）等同于实例所属子类的标识，因此在标识式建模中无需使用判别器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1394]

3.1410

互斥子状态 disjoint substate

一种不能与包含在同一组合状态中的另一子状态同时保有的子状态。与 并发子状态（concurrent substate）相对。

3.1411

处置 disposal

对工件进行移除或归档（而非删除），以便实现可追溯性和可审计性。

[来源：IEEE 2675-2021, 3.1]

3.1412

处置约束 Disposal constraints

处置策略对系统产生的任何约束，包括成本、进度和技术约束。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1413

处置启用系统需求 Disposal enabling system requirements

用于实现所关注的系统处置所需的任何系统的需求。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1414

处置程序 Disposal procedure

一种处置程序, 包括一系列处置操作, 采用特定的处置技术, 并借助特定的处置辅助手段来执行。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1415

处置记录 Disposal record

与处置相关的数据、信息或知识的永久、可读形式。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1416

处置报告 Disposal report

为利益相关方编写的报告, 旨在传达处置活动的状态、结果和成果。其内容可包括: 可供复用/存储的系统要素清单, 以及法规或组织标准要求的任何文件或记录。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1417

处置策略 Disposal strategy

确保系统或系统要素停用、拆除并从运行环境中移除所需的方法、进度、资源及具体考量因素。

[来源: INCONE 系统工程手册 v4, 附录 E]

3.1418

处置系统 disposed system

经过处置流程改造(即状态变更)的系统。[来源: ISO/IEC TR 29110-5-6-2:2014《系统与软件工程——超小型实体(VSEs)生存周期特征》第5-6-2部分: 系统工程——管理与工程指南: 通用特征组: 基础特征3.3]

注: 系统方法论需考量系统的整体架构及其完整生存周期。该方法涵盖系统所有要素, 直至用户最终报废系统且外部企业完成处置流程的全过程。

3.1419

发布信息 distribute information

[过程]使相关信息可用的过程。

[来源: 项目管理知识体系指南(PMBOK®指南)第六版]

3.1420

分布式计算 distributed computing

将计算任务与数据分散部署至由网络连接的多台计算机中。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1400]

3.1421

分布式拒绝服务攻击 distributed denial of service attack,; DDOS attack

未经授权访问系统资源, 或通过攻陷多个系统占用目标系统的带宽或资源, 从而延迟系统运行和功能, 最终导致授权用户可用性损失。

[来源: ISO/IEC TR 7052:2023, 3.22]

3.1422

分布式处理 distributed processing

一种信息处理方式，其离散组件可部署于不同位置，且组件间的通信可能出现延迟或发生故障。

[来源：ISO/IEC 10746-2:2009, 3.2.2]

3.1423

分发副本 distribution copy

为了安装到其他硬件上而由软件确定的主版本制作的副本（驻留在例如服务器上或光盘之类的物理介质上）。

[来源：GB/T 26236.1-2010, 3.5]

3.1424

分布式透明性 distribution transparency

向用户隐藏支撑分布式处理所需的系统复杂度特定方面的属性。

[来源：ISO/IEC 10746-2:2009, 11.1.1]

3.1425

分发单元 distribution unit

一种作为一组分配到某一过程或处理器的对象或构件的集合。分发单元能用运行时间的组合或聚集加以表示。

3.1426

分布偏移 distributional shift**数据集偏移 dataset shift**

〈机器学习〉训练数据分布与期望数据分布之间的距离。

注：随着用户与系统的交互（由此导致期望数据分布）随时间变化，分布偏移的影响通常会增加。

[来源：ISO/IEC TR 29119-11:2020, 3.1.29]

3.1427

干扰 disturbance

操作故障或事件或者任何可能改变系统状态的事件。

示例：导致系统关闭的 OS 进程突然关闭、系统用户数显著增加。

注：在此评价模块的周境下，干扰仅限于外部故障或事件，而不是需要修改应用程序或 OS 代码的内部故障。

[来源：GB/T 25000.45-2018, 1.184]

3.1428

目录信息树 DIT; DIT

目录信息树。

[来源：ISO/IEC 13235-3:1998, 4]

3.1429

分割操作 dividing action

启用两个或多个链的操作。

[来源：ISO/IEC 10746-2:2009, 13.1.4]

3.1430

标准分部 division of standards

该分部制定了一套具备互补性用途的标准族。

[来源：ISO/IEC 25000:2014, 4.13]

3.1431

定义语言 DL; DL

定义语言。

[来源: ISO/IEC 10746-1:1998]

3.1432

零投资备选方案 do nothing alternative

在决策分析中, 不对任何拟议备选方案进行投资的备选方案。

注: 并非指完全无所作为, 而是指将资金投入可随时变现的、能产生预定回报率的投资项目 (如债券、计息账户, 或投入组织中效益更高的部门)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1418]

3.1433

文档 document

供人使用的具有唯一标识的信息单元。

示例: 打印或电子形式的报告、规格说明、手册或书籍。

注1: 一份文档可能是单独的信息项, 也可能是较大信息项的一部分。

注2: 文档包括纸质文档和电子文档。

注3: 术语“文档”的使用并不是为了排除广义“文件”的解释, 尤其是在其适用于基于模型的工程或合规性时。

[来源: ISO/IEC/IEEE 15289:2017, 3.1.10, 有修改]

3.1434

文件分析 document analysis

通过分析现有文档, 识别与需求相关的信息的一种启发式技术。

[来源: ISO/IEC/IEEE 24765:2017, 3.1256]

3.1435

文档控制 document control

将配置管理应用于文档控制。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.1436

文档管理系统 document management system; DMS

支持完整文档、图像及其他介质的存储、检索、版本控制与加工处理的系统。

参见: 内容管理系统 (content management system)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1422]

3.1437

文档集 document set

已分割为独立标识卷或产品的文档, 以便于分发或使用。

[来源: ISO/IEC/IEEE 26514:2022, 3.1.20]

3.1438

文档类型定义 document type definition; DTD

文档的结构、内容和语义的模板。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.12]

3.1439

文档类型定义 (DTD) document type definition (DTD); DTD

文档结构、内容及语义的模板。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.12]

3.1440

文档汇集 documentation

文档编制

文档管理

〈信息技术〉与特定主题相关的文档集合。

〈需求工程〉说明软件、设备、应用程序或服务使用方法的信息。

〈软件与系统工程〉为单次或一系列活动、任务、事件及其相关计划、结果提供证据依据的记录（如报告、计算文件、规范）或其他同类成果，该证据可支撑更大型的产品、过程或项目。

参见：文档、用户使用信息。

示例：印刷手册、屏幕端信息、独立在线帮助、用户指南、参考手册、教程、维基文档、输入表单、错误提示信息、用户界面。

注：可作为独立文档集提供、嵌入式文档集提供，或两种方式同时提供。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3.1441

文档计划 documentation plan

明确系统或软件生存周期内拟编制文档的计划。

[来源: ISO/IEC/IEEE 15289:2015, 5.10]

3.1442

文档审查 documentation reviews

收集信息集并开展评审，以判定其准确性和完整性的过程。

[来源: PMBOK®Guide—Fifth Edition]

3.1443

文档树 documentation tree

描绘特定系统的所有文档并展示其相互间关联关系的图表。

参见：规范树（specification tree）。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1426, 有修改]

3.1444

成文信息 documented information

组织需实施控制和维护的信息及其所依附的载体。

注：成文信息可采用任意格式和载体，亦可来源于任意渠道。成文信息既可以指组织为开展运营活动所编制的信息（文档集），也可以指已达成结果的佐证材料（如记录、关键绩效指标）。

[来源: ISO/IEC 19770-1:2017, 3.19]

3.1445

领域、取值范围 domain

由为业内人士所理解的一套概念和术语所表征的一种知识或活动范围。

[来源: IEEE 7010-2020, 2.1]

2. 参数取值的范围或集合。

[来源: GB/T 38639-2020]

3.1446

领域分析 domain analysis

对领域内系统进行分析，以发现其共性与差异。

2. 通过识别、捕获和组织软件系统开发过程中使用的信息，使其能够用于在特定领域内创建新系统。

[来源：IEEE 1012-2024 IEEE, 3.1]

3.1447

领域架构 domain architecture

参考架构 reference architecture。

产品线架构 product line architecture。

描述软件与系统产品线高层级设计的核心架构，包括约束软件与系统产品线内所有成员产品的架构性结构和构造（如通用规则和约束条件）。

注：产品线成员产品的应用架构重用（可能经修改后）领域架构的通用部分，绑定领域架构的可变部分。成员产品的应用架构可（但不必需）提供可变性。

[来源：GB/T 41865-2022]

3.1448

领域架构、产品线架构、参考架构 domain architecture、product line architecture、reference architecture

软件与系统产品线高层级设计的核心架构，包括约束产品线内所有成员产品的架构性结构和构造（如通用规则和约束条件）。

[来源：ISO/IEC 26550:2015, 3.10]

2. 可容纳成员产品可变性的产品线通用架构。

注：领域架构包含旨在满足领域模型所规定需求的设计。领域架构记录设计，而领域模型记录需求。领域架构可被改编以创建领域内软件系统的设计，并为单个软件系统中的资产配置提供框架。

[来源：ISO/IEC 26552:2019, 3.5]

3.1449

需求领域资产 domain assets in requirements

领域需求工程过程中产生的可复用工件。

示例：资产提案、领域需求规范、领域需求模型。

[来源：ISO/IEC 26551:2016, 3.10]

3.1450

领域资产 domain asset

核心资产 core asset

核心资产 core asset。

领域工程生存周期过程的输出，在应用工程中可被重用于产品的生产。

注 1：领域资产可包括领域特征、领域模型、领域需求规格说明、领域架构、领域组件、领域测试用例、领域过程描述、以及其他资产。

注 2：在系统工程中，领域资产可以在后续系统设计中可重用的子系统或组件。领域资产需考虑其原始需求和技术特征。领域资产包括但不限于用例、逻辑原则、环境运行数据、以及从以前项目中获得的风险或机遇。领域资产并非现货或可随时可供使用的物理产品。物理产品（如机械零件、电子元件、线束、光学镜片）根据其各自学科的最佳实践进行存储和管理。领域资产有其自身的生存周期；GB/T 22032 可用于其生存周期管理。

[来源：GB/T 41865-2022, 1.45]

3.1451

领域组件 domain component

产品线内成员产品之间可重用的组件。

[来源：ISO/IEC 26553:2018, 3.10]

3.1452

领域工程 domain engineering

用于规定和管理产品线通用性和可变性的一系列过程组成的生存周期。

注1：领域资产是在领域工程过程中开发和管理，在应用工程过程中复用。

注2：根据领域资产的类型，即系统领域资产或软件领域资产，所使用的工程过程可由相关规范决定。

注3：IEEE 1517-2010 第三章将领域工程定义为基于可复用的方法，用于为一类系统、子系统或成员产品定义范围（即领域定义）、描述结构（即领域架构）和创建资产（如需求、设计、软件代码、文档）。

[来源：GB/T 41865-2022, 1.46]

3.1453

领域工程过程 domain engineering process

领域资产开发过程。

[来源：ISO/IEC 26555:2015, 3.5]

3.1454

领域接口 domain interface

产品线内成员产品组件之间可重用的接口。

[来源：ISO/IEC 26553:2018, 3.11]

3.1455

领域层 domain layer

测试项最高层级的抽象。

注：由领域专家使用其熟悉的方式选出该层级的关键字。

[来源：GB/T 38634.5: 2024, 4.1]

3.1456

领域模型 domain model

领域分析的产物，用于呈现该领域的需求。

注：领域模型识别并描述该领域内软件系统所包含的数据结构、信息流、功能、约束及控制机制。该模型阐释了该领域内软件系统需求的共性与差异性。

[来源：ISO/IEC/IEEE 24765j:2021]

3.1457

领域实现 domain realization

领域工程中的一个过程，包括详细设计和实现。

[来源：ISO/IEC 26553:2018, 3.12]

3.1458

领域需求分析 domain requirements analysis

对领域需求建模的子过程，以便分析和检查需求中产品线的共性和可变性。

[来源：GB/T 45804-2025, 1.1059]

3.1459

领域需求获取 domain requirements elicitation

识别领域利益相关方对产品线的原始需求的子过程。

[来源：GB/T 45804-2025, 1.1058]

3.1460

领域需求管理 domain requirements management

管理领域需求及与其相关的领域和应用制品的可追溯性和变更的子过程。

[来源：GB/T 45804-2025, 1.1062]

3.1461

领域需求规格说明 domain requirements specification

根据领域分析结果文档化产品线领域需求的子过程。 [来源 1：ISO/IEC 26551:2016 软件与系统工程——产品线需求工程的工具与方法, 3.13]

[来源 2：GB/T 45804-2025 软件与系统工程 产品线需求工程的工具和方法]。

3.1462

领域需求验证与确认 domain requirements verification and validation

验证与确认领域需求正确、一致和完整的子过程。

[来源：GB/T 45804-2025, 1.1061, 有修改]

3.1463

领域范围界定 domain scoping

识别和界定功能领域的子过程。该子过程对构思产品线，以及提供足够的复用潜力以调整产品线创建是非常重要的。

注：用于识别和界定对预期产品线至关重要的功能域，这些功能域需具备足够的复用潜力以证明创建该产品线的合理性。

[来源：GB/T 41865-2022 软件与系统工程 产品线工程与管理参考模型]

3.1464

领域超集 domain supersets

包含特征目录和共享资产超集的集合。

[来源：ISO/IEC 26580:2021, 3.3]

3.1465

领域测试资产 domain test asset

将在应用测试中重用的领域测试制品。

注：领域测试资产，例如测试计划，回归测试制品，可在领域测试中重用。

[来源：ISO/IEC 26554:2018, 3.6]

3.1466

领域测试需求 domain test requirement

领域测试应涵盖的领域工件特定要素。

注：领域测试需求涵盖功能与非功能的共性与变异性，包括其正常状态与错误状态。

[来源：ISO/IEC 26554:2018, 3.7]

3.1467

领域测试 domain testing

领域工程阶段，其作用是测试领域制品。

注：对与产品线（领域）相关制品进行的测试，而不是对单个产品的测试。

[来源：ISO/IEC 26554:2018, 3.5]

3.1468

领域可变性模型 domain variability model

产品线可变性的显式定义。

[来源：ISO/IEC 26558:2017, 3.4]

3.1469

基于领域的需求 domain-based requirement

起源于某个特定应用领域的需求。

[来源: GB/T 25000.30-2021,]

3.1470

决策 dominance

一种决策技术,用于寻找在所有属性上同等优良,且至少在一个属性上更优的备选方案。参见:词典排序法、满意决策法。

[来源: ISO/IEC/IEEE 24765:2017,]

3.1471

完成 done

完成的定义 definition of done

对工作产品在结束特定生存周期活动或任务并准备就绪可供使用之前所需满足的质量属性的声明。

[来源: ISO/IEC 33202:2024, 3.13]

3.1472

空操作 do-nothing operation; no-operation; no-op

一种计算机操作,其执行除了将指令计数器推进至下一条指令外,没有其他作用。

注:用于在程序中预留空间,或者通过重复执行以等待给定的事件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2992]

3.1473

点标记法 dot notation

将父类的名称连接到具有句点字符的从属类的名称的命名技术。

示例: 图表特征引用标识 ABC/A31.3 采用点标记法,将父图表的页面引用标识 ABC/A31 与该图表中 3 号框图的特征引用标识进行衔接。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1456]

3.1474

双倍数据速率同步动态随机存取存储器 double data rate SDRAM,; DDR SDRAM

具有更高存取速度和带宽的同步动态随机存取存储器单元,因为它在一个内部时钟周期内传输两个连续字。

[来源: ISO/IEC/IEEE 24765c:2014]

3.1475

双倍数据速率 x2 double data rate x 2; DDR2

双倍数据速率 x2 (速度是 DDR 的两倍)。

[来源: ISO/IEC/IEEE 24765c:2014]

3.1476

双操作数指令 double-operand instruction

二地址指令 two-address instruction

包含两个地址字段的计算机指令。

参见: 一地址指令 (one-address instruction), 三地址指令 (three-address instruction), 四地址指令 (four-address instruction), 零地址指令 (four-address instruction)。

示例: 将 A 的内容与 B 的内容相加的指令。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5016, 有修改]

3.1477

停机时间 down time

服役期间内，系统未执行或无法执行其规定功能的时段。

参见：开机时间（up time），忙时（busy time），闲时（idle time），平均修复时间（mean time to repair），设置时间（mean time to repair）。

注：停机时间可能由计划内或计划外的支持活动、系统失效、或源于系统运行时间或操作环境的失效所导致。

[来源：IEEE 982:2024, 3.1, 有修改]

3.1478

降级权 downgrade right

授予接收、安装或使用比当前授权版本更早的软件版本的权利。

[来源：ISO/IEC 19770-3:2016, 3.1]

3.1479

向下兼容 downward compatible

硬件或软件可兼容其早期版本或低复杂度版本。

参见：向上兼容（upward compatible）。

示例：可处理其自身早期版本所创建文件的程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1462]

3.1480

向下压缩 downward compression

软件设计中一种去模块化形式，指将上级模块复制至下级模块的程序体中。

参见：横向压缩（lateral compression）、向上压缩（upward compression）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1463]

3.1481

漂移 drift

退化 degradation

陈旧 staleness

〈机器学习〉机器学习模型行为随时间发生的变化。

[来源：ISO/IEC TR 29119-11:2020, 6.1.30]

〈人工智能〉训练数据与采集到的数据输入之间偏差的表现。

参见：算法漂移（algorithmic drift），概念漂移（concept drift），数据漂移（data drift）。

注：这些变化通常会降低预测的准确性，并可能需要用新数据重新训练模型。

[来源：IEEE 7014-2024, 3.1]

3.1482

驱动程序 driver; launcher

外部动态组件或模拟器，用于激活系统元素集合执行其功能。

参见：测试驱动程序（test driver）。

[来源：ISO/IEC/IEEE 24748-6:2023, 3.1.2]

3.1483

文档类型定义 DTD

文档类型定义。

注：适用于可扩展标记语言（XML）或标准通用标记语言（SGML）规范。

[来源：ISO/IEC/IEEE 26531:2023]

3.1484

双系统启动 dual boot

具备多种启动模式，可在同一台计算机上运行两个不同的操作系统。

参见：单系统启动 (single boot)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1481]

3.1485

[程序]对编 dual coding

一种开发技术，由不同的程序员或不同的程序设计小组，根据同一份规格说明书开发出功能上完全相同的程序的两个版本。所获得的源代码可以采用同一种语言，也可以采用不同的语言。双份编码的目的在于提供错误检测，提高可靠性，提供附加的文件说明，或使系统的程序设计错误或编译程序错误影响最终结果的概率降低。

参见：软件多样性 (software diversity)。

3.1486

双列直插式封装 dual inline package; DIP

引脚呈两列排列的微型电路器件。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1487

哑终端 dumb terminal

不可编程终端 nonprogrammable terminal

不具备独立数据处理能力的用户终端。

[来源：ISO/IEC 2382:2015]

3.1488

虚参数 dummy parameter

软件模块中使用的变量。用来表示调用模块传送至模块的数据或程序元素。

参见：形参 (formal parameter)。

3.1489

持续时间 duration; DUR

DU

同一类型任意两个计时点之间的正差值。

参见：工作量 (effort)。

[来源：IEEE 982:2024, 3.1]

3.1490

责任 duty

当某些情况发生时，执行特定行动的义务或期望。

[来源：IEEE 7000:2021, 3.1]

3.1491

责任伦理 duty ethics

确定普遍道德法则来约束所有理性个体的行为的伦理理论。

[来源：IEEE 7000:2021, 3.1]

3.1492

二元选择构造 dyadic selective construct

if-then-else 构造，其中为分支的两个结果都指定处理。

参见：一元选择结构 (monadic selective construct)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1488, 有修改]

3.1493

动态的 dynamic

指在计算机程序执行过程中发生的事件或过程。

参见：静态的 (static)。

示例：动态分析 (dynamic analysis)、动态绑定 (dynamic binding)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1489]

3.1494

动态分配 dynamic allocation

把可编址的存储器和其他资源分配给正在执行的程序。

参见：2.522 条 动态资源分配 (dynamic resource allocation)。

3.1495

动态分析 dynamic analysis

根据系统或组件在执行过程中的行为评估系统或组件的过程。与静态分析 (static analysis) 相对。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1490]

3.1496

动态分析器 dynamic analyzer

借助对程序执行情况的监控，帮助对计算机程序进行估计的软件工具。

示例 1：探测工具、软件监控器和跟踪器。

示例 2：与 XXX 条 静态分析程序 (static analyzer) 相对照。

3.1497

动态属性 dynamic attribute

硬件标识 (HWID) 标签的组成元素，可在产品生存周期内发生变更，或在创建后进行定义。

[来源：ISO/IEC 19770-6:2024, 3.1.1]

3.1498

动态绑定 dynamic binding

在计算机程序执行过程中完成的绑定。

参见：静态绑定 (static binding)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1492]

3.1499

动态断点 dynamic breakpoint

预定义触发事件为程序运行时特征的断点。

参见：静态断点 (static breakpoint)、代码断点 (code breakpoint)、数据断点 (data breakpoint)、尾码断点 (epilog breakpoint)、可编程断点 (programmable breakpoint)、头码断点 (prolog breakpoint)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1493]

[来源：例如：执行任意二十行源语句后触发的断点]

3.1500

动态缓冲 dynamic buffering

为计算机程序分配的缓冲区会根据当前需求，在程序执行过程中动态调整的缓冲技术。

参见：简单缓冲。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1494]

3.1501

动态总线定宽 **dynamic bus sizing**

在操作过程中可根据请求调整总线带宽的能力。

注：适用于直接内存访问（DMA）场景。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1495]

3.1502

动态分类 **dynamic classification**

其中的某一对象可更改其类的泛化的一种语义类。

与 XXX 条 静态分类（static classification）相对。

3.1503

动态转储 **dynamic dump**

在计算机程序执行期间发生的转储。

参见：静态转储（static dump）、变更转储（change dump）、内存转储（memory dump）、事后转储（postmortem dump）、选择性转储（selective dump）、快照转储（snapshot dump）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1496, 有修改]

3.1504

动态差错 **dynamic error**

随输入的时变特性变化的误差。

参见：静态误差（static error）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1497]

3.1505

动态调用 **dynamic invocation**

构造并发出一个请求，该请求的签名在运行前可能是未知的。

[来源：ISO/IEC/IEEE FDIS 24765 3.1498]

3.1506

动态模型 **dynamic model**

描述对象间单个请求或请求模式的模型。

参见：静态模型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1499]

3.1507

动态产品 **dynamic product**

在测试和/或运行环境中运行时，可测量的系统或软件产品。

[来源：GB/T 25000.41-2018, 4.2]

3.1508

动态程序分析 **dynamic program analysis**

基于软件系统或组件在执行期间的行为对其进行评价的过程。

注：软件被编译并在一定数量的输入数据测试用例上运行。然后检查系统的物理响应并与预期结果进行比较。动态程序分析可以手动或使用自动化过程进行。

[来源：ISO/IEC 23643:2020, 3.5]

3. 1509

动态随机存取存储器 dynamic random access memory; DRAM

需通过频繁刷新操作维持数据存储的随机存取存储器。

注：采用单稳态电路架构。

[来源：ISO/IEC/IEEE 24765c:2014]

3. 1510

动态重定位 dynamic relocation

计算机程序在它执行期间的重定位。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1503, 有修改]

3. 1511

动态资源分配 dynamic resource allocation

根据当前需求，在程序执行过程中动态调整为其分配的计算机资源的资源分配技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1504]

3. 1512

动态重构 dynamic restructuring

在程序执行期间，重构数据库、数据结构、计算机程序或系统部件集的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1505, 有修改]

3. 1513

动态模式 dynamic schema

在任意不变模式约束下，对一个或多个信息对象允许的状态变更所做的规范说明。

注：信息系统中的行为可建模为从一种静态模式到另一种静态模式的转换，即实例从一种类型到另一种类型的重新分类。在信息语言体系中，涉及一组对象的状态变更可视为该组对象间的交互行为。并非所有参与交互的对象均需发生状态变更，部分对象可仅以只读方式参与。

[来源：ISO/IEC 10746-3:2009, 6.1.3]

3. 1514

动态骨架 dynamic skeleton

独立于接口的骨架，供服务器用于处理直到运行时才知道其签名的请求。

[来源：ISO/IEC 19500-2:2012, 4.2]

3. 1515

动态存储分配 dynamic storage allocation

根据当前程序及其他正执行程序的需求，在程序执行过程中动态调整为其分配存储资源的存储分配技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1507]

3. 1516

动态测试 dynamic testing

需要运行测试项的测试。

[来源：GB/T 38634.1-2020, 3.9]

3. 1517

最早完成日期 early finish date; EF

关键路径法中，根据网络逻辑和进度约束情况，一项活动可以结束的最早可能时间节点。

[来源：GB/T 23691-2009, 2.6.22]

3.1518

最早开始日期 early start date (ES)

在关键路径法中, 根据网络逻辑和进度约束情况, 一项活动可以开始的最早可能时间节点。

[来源: GB/T 23691-2009, 2.10.18]

3.1519

早期失效期 early-failure period

老化周期 burn-in period

系统或组件生存周期中的一段时间, 在此期间, 随着问题的检测和修复, 硬件失效的发生率逐渐降低。

参见: 定失效率周期 (constant-failure period), 磨损失效周期 (wearout-failure period), 浴盆曲线 (bathtub curve)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1512]

3.1520

挣值 earned value (EV); EV

给定时间段内已完成的实际工作所对应的核准预算。

注1: 通过挣值可以衡量工作的进展情况。

注2: 挣值也可称为“赢得值”。

[来源: GB/T 23691-2009, 2.10.12]

3.1521

挣值 earned value; EV

已完成工作预算成本 budgeted cost of work performed; BCWP

以该工作授权预算为基准衡量工作完成量。

[来源: 项目管理知识体系指南 (PMBOK®指南)-第七版]

3.1522

挣值分析 earned value analysis

将项目已完成工作的计划工作量与实际工作量进行比较, 确定项目进度和成本偏离情况的方法。

[来源: GB/T 36964-2018, 3.22]

3.1523

挣值技术 earned value technique

一种用于衡量工作绩效的特定技术, 用于建立绩效衡量基线 (PMB)。

[来源: 项目管理知识体系指南 PMBOK 指南-第七版]

3.1524

回送 echo

将传输的信号返回到其源, 通常会有延迟, 以表明信号是反射而不是原始信号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1517]

3.1525

编辑 edit

修改计算机代码、数据或文档的形式或格式。

示例: 插入、重新排列或删除字符。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1524, 有修改]

3.1526

有效地址 effective address

对指定地址执行所需的索引、间接寻址或其他地址修改操作后得到的地址。

参见：生成地址、间接地址、相对地址

注：若指定地址无需进行任何修改，该地址即为有效地址。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1529]

3.1527

有效完整许可证 **effective full license**

对于软件，允许完全使用该软件的许可权。

示例：对某一软件产品第1版的基本的完全许可权，加上对该软件产品升级到第2版的基本升级许可权，合并成为对其第2版的有效的完全许可权。

注：一个有效的许可权由一个或多个基本许可权组成。

[来源：GB/T 26236.1-2010, 3.6]

3.1528

有效指令 **effective instruction**

对指定计算机指令中的地址执行所需的索引、间接寻址或其他修改操作后得到的计算机指令。

参见：绝对指令(absolute instruction)、直接指令(direct instruction)、立即指令(immediate instruction)、间接指令(indirect instruction)。

注：若指定指令无需进行任何修改，该指令即为有效指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1530]

3.1529

实际利率 **effective interest rate**

根据更高或更低复利频率进行调整后的利率。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1531]

3.1530

有效过程 **effective process**

实践过的、已文档化的、强制的、经培训的、已测量的和可改进的过程。

参见：XXX条 妥善定义的过程(well-defined process)。

3.1531

传出的 **efferent**

指软件系统中，数据或控制流从上级模块流向下级模块的相关属性。

参见：传入的(afferent)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1533]

3.1532

人工量 **effort**

完成一项活动或其他项目要素所需单位人工的数量。

注：通常以人·小时、人·日或人·周为单位。

[来源：GB/T 23691-2009, 2.3.14]

3.1533

无我程序设计 **egoless programming**

基于团队而非个人观念负责程序开发的软件开发技术。

注：它的目的是防止单个程序员过于关注他们自己的工作，从而影响客观评估。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1536]

3.1534

弹性 elasticity

对于一项服务，该服务根据分配给服务实例的资源量进行快速调整的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.1.5]

3.1535

电可擦除可编程只读存储器 electric erasable programmable read only memory; EEPROM

可通过电信号擦除并重新写入的可编程只读存储器。

参见：闪存（flash memory）。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1536

电子数据交换 electronic data interchange; EDI

将电子数据从数据库传输到数据库的结构化方式，通常使用电信网络。

[来源：ISO/IEC/IEEE 24765: 2025 3.1542]

3.1537

电子设计自动化 electronic design automation; EDA

微机单元和电路板等电子元件的软件驱动设计和开发。

[来源：ISO/IEC/IEEE 24765: 2025 3.1543]

3.1538

元素 element

支配类型的值或支配信息客体类别的信息客体，能分别从相同类型或相同类别信息客体的所有其他值中区别出来。

[来源：GB/T 36328-2018]

3.1539

元素类型 element type

元素的类别 / 种类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1546]

3.1540

基本过程 elementary process; EP

对用户有意义的最小活动单元。

[来源：GB/T 42449-2023,]

3.1541

ELSE 规则 ELSE-rule

针对表格中其他规则未覆盖的所有条件组合，应执行的操作。

注：ELSE 规则功能的使用为可选项。

[来源：ISO 5806:1984, 3.5]

3.1542

嵌入式计算机系统 embedded computer system

作为更大系统的一部分并执行该系统某些需求的计算机系统。

示例：航空器或快速交通系统中使用的计算机系统。

注：嵌入式系统的硬件和软件通常针对特定功能进行最小化和优化。嵌入式系统包括至少一个微控制器、微处理器或数字信号处理器。嵌入式系统旨在优化应用程序的可靠性、成本、尺寸和节能。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1550]

3.1543

嵌入式帮助系统 embedded help system

作为软件组成部分提供给用户的信息。

[来源: ISO/IEC/IEEE 26511:2018, 3.1.14]

3.1544

嵌入式用户信息 embedded information for users

作为软件组成部分访问的用户信息。

参见: 屏幕上的用户信息, 打印出的用户信息。

示例: 在屏幕上弹出帮助和帮助文本。

[来源: ISO/IEC/IEEE 26514:2022, 3.1.22]

3.1545

涌现性 emergence

实体表现出某种性质的原则, 这些性质只有归因于整体而非其组成部分时才具有意义。

注: 这些性质不能被简化或分解为任何单个组成系统的性质。

[来源: ISO/IEC/IEEE 21840:2019, 3.1.3]

3.1546

紧急情况 emergency

需要立即采取应对措施的严重、突发且通常具有危险性的情境。

[来源: ISO/IEC/IEEE 24748-9:2023, 3.1.2]

3.1547

紧急维修 emergency maintenance

执行非计划修改以暂时保持系统运行等待纠正性维护。

注: 可以执行紧急维修以使软件系统部分运行。可包括对软件或其参数的各种修改, 这些修改限制了操作、功能、输入、输出、接口或易用性。紧急维修可以被视为纠正性维护的一部分。

[来源: ISO/IEC/IEEE 14764:2021, 3.1.5]

3.1548

应急支持 emergency support

在纠正性支持实施前, 为使系统保持临时运行而开展的非计划性在役修改。

注: 应急支持可视为一种纠正性支持类型。应急支持可包括对系统、其配置或运行环境实施修改, 以改变运行方式、功能、输入项, 建立临时替代使用方案(变通方法), 从而避免触发异常或故障。

[来源: IEEE 982:2024, 3.1]

3.1549

发射器 emitter

最多可以连接到一个消费者的事件源。

[来源: ISO/IEC 19500-3:2012, 4.1]

3.1550

共情自主智能系统 empathetic autonomous and intelligent system; EA/IS

采用情感感知技术, 以算法方式对情绪、情感、心境、认知视角、注意力或意图进行推断、建模、模拟或激发的技术体系。

注: 指运用情感分析、生物特征识别、自然语言处理等过程, 对情感进行分析、推断或模拟的人工智能。基于自动化推断所形成的数据洞察或采取的相应行为, 通常会为个人、群体或其他系统与共情自主智能系统(EA/IS)之间的后续交互提供支撑。

[来源：IEEE 7014-2024, 3.1]

[来源：ISO/IEC 24765 FDIS, 3.1565]

[来源：同义词：情感计算系统 (affective computing system)、情感人工智能 (emotional AI)、情感检测人工智能 (emotion-detecting AI)]

3.1551

共情 **empathy**

理解或分担他人的情感或体验。

注：机器所实现的共情，应视为对人类共情的仿真，而非与人类共情本质相同的过程。

[来源：IEEE 7014-2024, 3.1]

3.1552

雇员 **employee**

在组织物理边界之内或之外以从属关系安排工作的人员。

[来源：IEEE 7005 2021, 3.1]

3.1553

仿真用户 **emulated user**

通过技术系统实现对用户提交的任务和时间行为的模仿。

[来源：ISO/IEC 14756:1999, 4.6]

3.1554

模拟器 **emulator**

接受与给定系统相同的输入并产生相同输出的设备、计算机程序或系统。

参见：模拟器 (simulator)。

注：通常用于测试或调试。

[来源：ISO/IEC/IEEE FDIS 24765 3.1570]

3.1555

启用行为 **enabled behavior**

表征一组对象的行为，这些对象由于建立行为而成为可能。

[来源：[来源：ISO/IEC 10746-2:2009, 13.2.2]

[来源：同义词：enabled behaviour]

3.1556

使能行为 **enabled behaviour**

表征一组对象的行为，这些对象由于建立行为而成为可能。

[来源：ISO/IEC 10746-2:2009, 13.2.2]

3.1557

使能 **enabling (a transition)**

当满足以下条件时，在特定模式和网络标记下使能变迁：(1) 变迁的每个输入位置的标记满足其针对特定变迁模式评估的弧注释对其施加的要求；(2) 当位置的标记包含（至少）评估弧标注所指示的多组标记时满足需求。

[来源：ISO/IEC 15909-1:2019 系统与软件工程——高级 Petri 网——第 1 部分：概念、定义和图形表示法, 3.8]

[来源：条目注释 1：变迁模式的确定保证变迁条件得到满足。]

3.1558

使能系统 **enabling system**

对所关注的系统在其生存周期阶段提供支持，但在其运行期间不必直接发挥功用的系统。

示例：当所关注的系统进入生产阶段时，需要一个帮助生产的使能系统。

注：每一个使能系统都有自己的生存周期。当使能系统为了其自身的需要也作为所关注的系统对待时，本标准适用于每一个使能系统。[来源：GB/T 22032-2021 系统与软件工程 系统生存周期过程]。

3.1559

支撑系统需求 Enabling system requirements

来自所有适用生存周期过程的支撑系统需求，包括实现支撑系统需求、集成支撑系统需求、验证支撑系统需求、转换支撑系统需求、确认支撑系统需求、运行支撑系统需求、维护支撑系统需求以及处置支撑系统需求。

[来源：INCONE 系统工程手册 v4，附录 E]

3.1560

制定 enactment

通过法律、裁决或其他权威行为建立某物。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1575]

3.1561

封装 encapsulation

访问职责的名称、含义和值的概念类完全独立于对其实现的访问。

参见：数据抽象，信息隐藏。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1576]

3.1562

编码 encoding

语法元素如何使用标识字符集表示的定义。

注：表示的详细信息提供了语法语法中的各种终端符号和数据类型。

[来源：[来源：ISO/IEC 15474-1:2002 , 4.2]

3.1563

编码 1 ENCODING.1

在 CDIF 标准族内定义的主编码。

注：CDIF 标准族支持多种传输格式，每种格式均由一种语法和一种编码构成。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.1564

成品项 end item

准备就绪可供使用的实体。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1579]（备注：因为 24765 中未标注定义的来源，故直接将 FDIS 24765 作为]

3.1565

期末惯例 end of period convention

将离散的现金流实例表示为发生在其所属周期的期末（与期初表示相对）。

注：初始投资显示在第零期期末。

[来源：ISO/IEC/IEEE FDIS 24765 3.1580]

3.1566

最终用户 end user

最终受益于系统结果的单独个体。

注：最终用户可能是软件产品的常规操作员或是公众场合下的临时用户。

[来源：GB/T 25000.40-2018]

3.1567

最终用户代表 end user representatives

挑选出的、代表最终用户的个人或组。

3.1568

研发工作 endeavor

IBD 开发工作旨在通过应用方法论。

示例 1：项目、项目群和基础设施职责。

示例 2：同义词：endeavour。

[来源：ISO/IEC 24744:2014，3.5]

3.1569

研发工作单元 endeavor element

一项工程中的基本组成部分。

示例：客户、发票（类）；姓名、年龄（属性）；高层类模型第 17 号（模型）；系统需求规格说明书（文档）；编码周期第 2 次、编码周期第 3 次（任务）。

注：在执行一项工程期间，开发人员会创建诸多工程元素，例如任务、模型、类、文档等。

[来源：ISO/IEC 24744:2014，3.7]

3.1570

研发工作组成单元 endeavor element; endeavour element

研发工作的基本组成单元。

示例：客户、发票（类），姓名、年龄（属性），17 号高层类模型（模型），系统需求说明书（文档），2 号编码周期、3 号编码周期（任务）。

注：在研发工作的执行过程中，开发人员会创建各类研发工作组成单元，如任务、模型、类、文档等。

[来源：ISO/IEC 24744:2014，3.7]

3.1571

实践 endeavour

贯穿于方法应用过程的始终，旨在交付某一产品或服务的 IBD 开发工作。

示例：项目、程序、基础设施职责。

[来源：ISO/IEC 24744:2014，3.5]

3.1572

实践元素 endeavour element

实践的简单组件。

示例：客户、发票（类）、名称、年龄（属性）、17 号高级别类模型（模型）、系统需求描述（文档）、第 2 编码周期、第 3 编码周期（任务）。

注：在实践的执行期间，开发者创建一些实践元素，如任务、模型、类、文档等。

[来源：ISO/IEC 24744:2014，3.6]

3.1573

持久性测试 endurance testing

性能效率测试的一种类型。用于评价测试项在指定的时间段，是否能够持续承受所需的负载。

[来源：GB/T 38634.1-2020，3.10]

3. 1574

工程化系统 **engineered system**

为实现一个或多个既定目标，在遵守适用约束条件的前提下，经设计或适配、可与预期运行环境交互的系统。

[来源：INCONE 系统工程手册，第五版.，Appendix C]

3. 1575

工程 **engineering**

对结构、机器、产品、系统或过程采用系统化、条理化、可量化的方法。

[来源：ISO/IEC TR 19759:2016]

3. 1576

工程变更 **engineering change**

在正式建立配置标识后，对已交付、待交付或开发中的硬件/软件配置项或多个配置项的配置进行的变更。

[来源：ISO/IEC/IEEE FDIS 24765，3.1587]

2. 在配置管理中，在正式建立配置标识后，对配置项或其他指定项的配置进行的变更。参见：配置控制、工程变更建议、偏离、放弃[来源：ISO/IEC/IEEE FDIS 24765，3.1587]。

（备注：两个推荐定义都来源于 24765，推荐定义 1 提供了状态信息，推荐定义 2 提供了关联术语，无法完全取代，若只能选一个推荐定义 2，原因：适用性更强，且参见项更有利于理解，或者可以综合二者优势重新定义）。

3. 1577

工程变更建议 **engineering change proposal; ECP**

在配置管理中，拟议的工程变更以及描述和建议变更的文件。

参见：配置控制。

[来源：ISO/IEC/IEEE 24765: 2025 3.1588]

3. 1578

工程变更请求 **Engineering change request; ECR**

无（仅是缩略语）。

3. 1579

工程经济学 **engineering economics**

分析软件项目必须做出的与项目成本和成本影响相关的选择的方法和模型。

[来源：IEEE 库]

3. 1580

工程组 **engineering group**

由从事工程学科的人员（包括负责人、管理人员和技术人员）组成的集合。工程学科的例子有系统工程、硬件工程、系统测试、软件工程、软件配置管理和软件质量保证。

3. 1581

工程接口参考 **engineering interface reference**

标识符，在工程接口参考管理域的上下文中，用于可用于分布式绑定的工程对象接口。

注：工程接口参考是建立分布式绑定所必需的，并且与基本工程对象为交互目的使用的绑定端点标识符不同。

[来源：ISO/IEC 10746-3:2009，8.1.16]

3. 1582

工程接口参考管理领域 **engineering interface reference management domain**

为分配工程接口参考而形成命名域的一组节点。

[来源：ISO/IEC 10746-3:2009，8.1.17]

3.1583

工程接口引用管理域 engineering interface reference management domain.

为分配工程接口引用而形成命名域的一组节点。

[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：体系结构，8.1.17]

3.1584

工程接口参考管理策略 engineering interface reference management policy

管理工程接口参考管理域联合的一组许可和禁止。

[来源：ISO/IEC 10746-3:2009，8.1.18]

3.1585

工程接口引用管理策略 engineering interface reference management policy.

管理工程接口引用管理域联邦的一组许可和禁止规则。

[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：体系结构，8.1.18]

3.1586

工程视点 engineering viewpoint

对 ODP 系统及其环境的视点，关注支持系统中对象之间分布式交互所需的机制和功能。

[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：体系结构，4.1.1.4]

3.1587

增强的能力 enhanced capability

可由一个可信赖的过程改进方案证明的，比当前评估的能力更强的能力。

3.1588

增强 enhancement

针对并实现新需求的软件变更。

参见“变更（change）”“修正（correction）”。

注：“增强”一词主要用作一种维护类型，并用于对修改请求进行分类。软件增强分为三种类型：适应性增强、完善性增强和添加性增强。增强不属于软件纠正。这些活动通常也会导致功能规模发生改变（例如，变更请求）。

[来源：ISO/IEC/IEEE 14764:2021，3.1.7]

3.1589

增强项目 enhancement project

对既有应用程序进行功能完善的项目。这就意味着可能会在既有应用程序中添加、修改和删除功能。

[来源：GB/T 42588-2023，3.121]

3.1590

增强项目功能点分析 enhancement project function point analysis

对实现了既有应用程序修改的项目进行测量时进行的计数。

[来源：GB/T 42588-2023，3.1.22]

3.1591

增强项目功能点计数 enhancement project function point count; EFP

应用本文件（GB/T 42449-2023）来测量增强项目功能规模的活动。

[来源：GB/T 42449-2023，3.24；ISO/IEC 20926:2009，3.24]

3.1592

增强项目功能点计数（EFP） enhancement project function point count (EFP); EFP

应用 ISO/IEC 20926:2009 来测量增强项目功能规模的活动。

[来源：ISO/IEC 20926:2009 软件和系统工程——软件测量——IFPUG 功能规模测量方法 2009，3.24]

3.1593

确保 ensure

确保事情发生或事件发生。

参见：确保（assure）。

注：Insure 仅用于保险事项。

[来源：IEEE 730-2014，3.2]

3.1594

授权创建者/许可模式创建者 Ent creator entitlement schema creator

最初创建授权（Ent）的实体。

注：该实体可以是创建或发布授权所关联软件的组织的一部分，在这种情况下，授权创建者与软件创建者为同一实体。授权创建者也可能是持有许可权利的另一独立组织，甚至可以是与软件创建者无关的第三方组织（例如，由顾问或工具开发商为遗留软件授权创建者的情况）。

[来源：ISO/IEC 19770-3:2016，3.1.10]

3.1595

企业 enterprise

为应对复杂且具有挑战性的情势或环境，人类发起的需要努力和勇气的事业。

注 1：处于早期阶段的企业通常具有隐性的或可能构思不全的战略、总体目标和具体目标。一个成熟的企业通常具有显性且清晰定义的战略、总体目标和具体目标。对于正在运营的企业，它作为一个系统，通过生存历史中的生存周期活动来实现其使命。

注 2：将企业等同于组织是很常见的做法。然而，在系统工程和企业建模领域，“企业”一词用于表示具有挑战性的活动，而非参与此类任务或风险事业的组织或组织集合。

注 3：在中文语境下，企业通常指从事商业、工业或专业活动的营利性经济组织。这与系统工程和企业建模领域标准化各领域涉及的 enterprise 的内涵和外延有较大区别。本文件将 enterprise 译为“企业”，取“企”和“业”两个字所分别蕴涵的“谋求”和“事业”之意，正好符合 enterprise 在相关国际标准中的定义。

注 4：虽然企业是为了应对复杂且具有挑战性的情势或环境而发起，但复杂本身不是企业的区别特征（GB/T 15237.1-2000，3.2.7）。

注 5：企业可以作为架构化工作所关注的利益实体。架构化工作可以帮助澄清并明确任务或风险事业的战略、总体目标和具体目标。随着需要的产生，架构化可以帮助随着时间的推移改变企业的规格说明和范围。

[来源：ISO/IEC/IEEE DIS 42024，3.3.5，有修改]

3.1596

企业环境因素 enterprise environmental factors; EEF

不受团队直接控制，但会影响、限制或指导项目、项目群或项目组合的条件。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.1597

企业观点 enterprise viewpoint

ODP 系统及其环境的观点，侧重于该系统的目的、范围和策略。

[来源：ISO/IEC 10746-3:2009，4.1.1.1]

3.1598

授权标签 entitlement

封装软件权利相关信息的数据结构。

[来源：GB/T 36329-2018，？]

3.1599

授权管理 entitlement management

软件权利相关信息的组织和使用方法。

[来源：GB/T 36329-2018，3.1]

3.1600

授权模式 entitlement schema

软件授权模式 software entitlement schema

包含许可交易及其相关授权信息的数字封装的信息结构。

注：单次交易不一定能封装完整（或有效）的授权。有效的授权需通过对多次许可交易（包括初始完整许可及其后续的升级和/或维护交易）进行综合分析来确定。

[来源：ISO/IEC 19770-3:2016，3.1.11]

3.1601

实体 entity

通过测量其属性、表述其特性的对象。

示例：一个对象可能是过程、产品、项目或资源。

[来源：GB/T 25000.40-2018]

3.1602

实体属性 entity attribute

实体固有的、可通过人工或自动化手段进行定量或定性区分的属性或特征。

[来源：ISO/IEC 25000:2014，4.1]

3.1603

实体组件 entity component

具有持久状态、通过主键在架构上对客户端可见的标识，以及可具备事务性行为的 CORBA 组件。

[来源：ISO/IEC 19500-3:2012，4.1]

3.1604

关联实体 entity dependent

〈实体〉在没有其他实体存在的情况下，对业务本身没有意义或不重要，以致：

——实体 X 的发生与实体 y 的发生有必要关联；

——删除一个实体 Y 的发生会导致删除所有相关的实体 X 的发生。

[来源：GB/T 42449-2023，X.X]

3.1605

独立实体 entity independent

〈实体〉在没有其他实体存在的情况下，对业务本身是有意义的或重要的。

[来源：GB/T 42449-2023，X.X]

3.1606

实体实例 entity instance

由实体所表示的一组现实或抽象事物中的一个。

注：实体的每个实例都可通过参与其主键的属性值进行唯一标识。[键样式]。

[来源：ISO/IEC/IEEE FDIS 24765，3.1.56]

3.1607

所关注实体 **entity of interest; Eoi**

架构描述的主题。

示例：企业、组织、解决方案、系统（包括软件系统）、子系统、过程、业务、数据（作为数据项或数据结构）、应用、信息技术（作为集合）、任务、产品、服务、软件项、硬件项、产品线、系统族、系统之系统、系统集成、应用程序集合等。

注 1：在本文件中，术语“所关注实体”是指在编制架构描述时正在考虑其架构的实体。

注 2：本文件将所关注实体与非架构描述主题的其他实体区分开来。

注 3：在本文件中，对一个实体的关注旨在涵盖对该实体的环境、生存周期、架构、需求、设计、实现和运营的关注。这些关注是通过各方面体、关注点和利益相关方角度来捕获的。

[来源：GB/T 45630-2025, 3.12]

3.1608

实体联系图 **entity-relationship (E-R) diagram; E-R diagram**

一组描绘真实世界实体及其之间逻辑关系的图表。

同义词：实体 - 关系图 (entity-relationship map)。

参见：数据结构图 (data structure diagram)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1608]

3.1609

实体关系图 **entity-relationship diagram; E-R diagram; entity-relationship map**

描绘一组现实世界实体及其之间的逻辑关系的图示。

参见：2.406 条 数据结构图 (data structure diagram)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1608]

3.1610

输入(类型) **entry (type)**

将功能用户得到的数据组通过边界移动到需要它们的功能过程的一种数据移动类型。

注：“输入”可以被认为包括特定相关数据操作（例如：输入数据的校验）。

[来源：GB/T 42452-2023, 3.8]

3.1611

进入动作 **entry action**

进入状态机的某一状态（而不管达到该状态采取何种转移）所执行的一种动作。

3.1612

进入准则 **entry criteria**

在某项工作能够成功开始之前必须存在的状态。

[来源：ISO/IEC/IEEE 24765:2017 系统与软件工程——词汇表]

2. 在评审或审计能够进行之前必须完成的工件和其他评审或审计要素。[来源：ISO/IEC/IEEE 24748-8:2019, 系统与软件工程——生存周期管理——第 8 部分：国防项目的技术评审和审计, 3.1] 参见：退出准则。

（备注：推荐定义 1 适用于任何工作/活动，关注“状态”，强调条件的满足情况；推荐定义 2 专门针对评审或审计，关注具体的“工件和要素”，强调可交付物和检查项，各有侧重，如果非要选一个，选择推荐定义 1，通用概念定义更适合作为术语的基础定义）。

3.1613

输入字段 **entry field**

屏幕上或窗口中用户输入数据的区域。

[来源：ISO/IEC 26514:2008, 4.19]

3.1614

入门资料 entry profile

面向初创型超小实体（即运营未满三年的超小实体）或从事小型项目（如项目规模少于六人月）的超小型实体（VSEs）的概况。

[[来源：ISO/IEC 29110-1-2:2024, 3.40]

3.1615

入口 entry; entry point

入口[点] entrance

测试项中开始执行测试项的点。通常是测试项中的第一条可执行语句。

注：入口点是测试项中的可执行语句，外部进程可以选择它作为通过测试项的一条或多条路径的起点。它通常是测试项中的第一个可执行语句。

[来源：GB/T 38634.4-2020, 4.18]

3.1616

枚举型 enumeration type

离散数据类型，其成员可以采用程序员显式定义的值。

参见：字符类型、整数类型、逻辑类型、实型。

示例：一种名为 COLORS 的数据类型，其可能值为 RED、BLUE 和 YELLOW。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1613]

3.1617

环境 environment

周围事物、条件或实体所受影响的周境。

注1：所关注实体（3.12）的环境包含能对某实体产生各种影响，如开发、技术、商业、运营、组织、政治、经济、法律、监管、生态和社会影响的外部实体，以及外部物理效应，如电磁辐射、带电粒子、引力效应、电场和磁场。

注2：术语“环境”加上限定词标签，识别出某一周境内的另一特定周境，如开发环境、测试环境和操作环境。

[来源：GB/T 45630-2025, 3.13]

3.1618

环境契约 environment contract

对象与其环境之间的契约，包括服务质量约束、使用和管理约束。

[来源：ISO/IEC 10746-2:2009, 11.2.3]

3.1619

环境特定 ORB 间协议 environment specific inter-ORB protocol; ESIOP

特定环境的对象请求代理间协议。

[来源：ISO/IEC 19500-2:2012, 5]

3.1620

史诗 epic

将大量相关功能集分解为单个功能或用户故事，并在较长时间内分部分实施。

[来源：ISO/IEC/IEEE 26515: 2018, 3.6]

3.1621

跋断点 epilog breakpoint

从给定程序或例程退出时启动的断点。

同义词：后置断点。

参见：序言断点、代码断点、数据断点、动态断点、可编程断点、静态断点。

[来源：ISO/IEC/IEEE 24765:2017，3.1426]

3.1622

跋断点 **epilog breakpoint**

后同步码断点 **postamble breakpoint**

从退出给定程序或例程时触发的断点。

参见：序言断点、代码断点、数据断点、动态断点、可编程断点、静态断点。

[来源：ISO/IEC/IEEE FDIS 24765，X.X]

3.1623

认知逻辑 **epistemic logic**

与知识或其验证程度相关的逻辑。

[来源：IEEE 7005 2021 3.2]

3.1624

时期 **epoch**

对象显示特定行为的时间段。

[来源：ISO/IEC 10746-2:2009，10.5]

3.1625

方程法 **equation**

基于基准数据建立参数模型，并通过输入各项参数，确定待估算项目工作量或成本估算值的方法。

[来源：GB/T 36964-2018，X.X]

3.1626

设备 **equipment**

旨在实现特定最终目标的关联组件。

[来源：IEC/IEEE 82079-1:2019，3.12]

3.1627

等价类 **equivalence class**

分类轴上的范围，具有判断目标系统是否应映射到该范围的规则。

[来源：ISO/IEC TR 12182:2015 3.8]

3.1628

等价划分 **equivalence partition**

变量或变量集的值域的子集。在测试项或者其接口中，预期测试项以同样的方式处理该子集中所有的值（即被认为是“等价”）。

[来源：GB/T 38634.1-2020]

3.1629

等价类覆盖率 **equivalence partition coverage**

测试集覆盖测试项中已识别等价类的比例。

注：在许多情况下，等价类识别是主观的（尤其是在“无效”等价类），所以对测试项中等价类进行明确计数是不可能的。

[来源：GB/T 38634.1-2020，3.12]

3.1630

等价类划分 equivalence partitioning

测试设计技术的一种。使用每一个等价类中的一个或多个有代表性的成员来设计测试用例。

[来源：GB/T 38634.1-2020，3.31]

3.1631

等效故障 equivalent faults

导致相同失效方式的两个或多个故障。

[来源：ISO/IEC/IEEE FDIS 24765，X.X]

3.1632

等效 IDL equivalent IDL

客户端映射；即组件声明的外部可见组件特征映射，或主页声明的特征。

注：由 IDL 中的组件定义隐式定义。

[来源：ISO/IEC 19500-3:2012 4.1]

3.1633

等效接口 equivalent interface

向客户端显示组件或主页表面特征的接口，允许客户端在组件的各个方面之间导航，并连接到组件的端口，如组件或主页的等效接口定义语言所定义。

[来源：ISO/IEC 19500-3:2012 4.1]

3.1634

实体-关系-属性 ERA

实体-关系-属性建模。

[来源：ISO/IEC 15474-1:2002，5.2]

3.1635

可擦除可编程只读存储器 erasable programming read only memory; EPROM

可通过紫外线擦除原有数据后重新写入的一类可编程只读存储器。

注：该器件可多次重复写入。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1636

可擦可编程只读存储器 erasable programming read only memory (EPROM); EPROM

使用紫外线 (UV) 擦除现有数据后可重写的可编程 ROM 类型。

注：该设备可以多次重写。

[来源：ISO/IEC/IEEE FDIS 24765：2025，3.1631]

3.1637

严重错误/勘误 errata

严重干扰服务的错误，且无已知的解决方法。

注：通常可以在冻结分支上引入对此类错误的修复。

[来源：ISO/IEC/IEEE FDIS 24765，X.X]

3.1638

错误 error

误差

差错

计算值、观测值或测量值（或状态）与真实值、规定值或理论正确值（或状态）之间的差异。

参见：故障 (failure)、缺陷 (defect)。

[来源：ISO/IEC/IEEE 15026-1:2019]

3.1639

出错分析 error analysis

对观察到的软件故障进行调查的过程，调查的目的是跟踪那个故障以找出故障源。

对观察到的软件故障进行调查以找出以下一些信息，例如故障原因。该故障是在开发过程中哪一个阶段发生的，预防或较早地探测出软件故障的方法。

调查软件错误、失效和故障以确定定量速率和趋势的过程。

3.1640

出错类别 error category

故障类别 fault category

错误、故障或失效可能归并到其中的一组类别之一，当错误、故障或失效发生或发现后，可根据其原因、危急程度、效果、故障所属的生存周期阶段或其他特性而确定其类别。

3.1641

出错数据 error data

失效数据 failure data

出错数据通常（但不是精确地）用于：描述软件的问题、故障、失效及其变更，它们的特性，以及遇到或改正这些问题的条件。

3.1642

错误猜测 error guessing

测试设计技术的一种。基于以往失效经验或者失效模式的常识推导出测试用例。

注：相关知识能通过个人经验获得，或以缺陷数据库或“缺陷分类”的形式进行封装。

[来源：GB/T 38634.1-2020, 3.14]

3.1643

出错消息 error message

当错误的数据被输入，或者当其他处理错误发生时，应用程序给出的消息。

[来源：GB/T 42588-2023, X.X]

3.1644

出错模型 error model

出错预测模型 error prediction model

在软件评价中，用于评估或预测系统余下故障数、需要的测试时间和类似特性的模型。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1645

出错预测 error prediction

关于系统或组件中故障的预期数量或性质的定量陈述 参见：错误模型，错误播种。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1638]

3.1646

错误处理 error processing

检测和响应程序错误的过程。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1647

错误植入 error seeding

缺陷植入 bug seeding

故障植入 fault seeding

故意地把已知故障加至已经有故障的计算机程序中，为了监控检测和删除的速度并估计程序中余下的故障数的过程。

[来源：ISO/IEC 25010:2023, 3.5.3]

3.1648

容错 error tolerance; fault tolerance

产品存在硬件或软件故障的情况下，仍能按预期运行的能力。

[来源：ISO/IEC 25010:2023, 3.5.3]

3.1649

转义 escaped

如果需要在文本字符串中包含与 <CloseText> 定界符匹配的模式，则每次出现该模式前都需加上 <EscapeCharacter>转义字符。

[来源：ISO/IEC 15475-3:2002, 7.2.11]

3.1650

建立和维护 establish and maintain

在整个组织中制定、记录和使用[政策或程序]。

参见：保持。

注：该短语的意思不仅仅是其组成术语的组合；它包括文档和用法。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1646]

3.1651

已确立需求 established requirement

项目已验证其满足项目特定标准。

（如清晰性、适用性和可行性）并已确认为利益相关者需求、期望和期待的准确表示的需求。

注：已确立需求被项目接受，构成产品开发的基础。

[来源：IEEE 730-2014, 3.2]

3.1652

建立行为 establishing behavior

在给定对象之间建立特定契约的行为。

[来源：ISO/IEC 10746-2:2009, 13.2.1]

3.1653

完工估算 estimate at completion; EAC

完成全部工作的预期总成本，其数值为截至目前的实际成本与完工尚需估算之和。

参见：挣值技术 (earned value technique)、完工尚需估算 (estimate to complete)。

[来源：ISO/IEC 24765 FDIS, 3.1650]

3.1654

完工尚需估算 estimate to complete; ETC

完成项目全部剩余工作的预期成本。

参见：完工估算 (estimate at completion)。

[来源：ISO/IEC 24765 FDIS, 3.1651]

3.1655

估算的功能点计数 estimated function point count

在应用生存周期的早期阶段，为确定应用或项目的规模而进行的可能的功能点计数，其中假设某些最小规范。

注：通常，按类型记录功能的数量，对于事务功能（事务）使用平均复杂度的默认值，对于数据功能（逻辑文件）使用低复杂度的默认值。

[来源：ISO/IEC 24570:2005 软件工程——NESMA 功能规模测量方法 2.1 版——功能点分析应用的定义和计数指南]

3.1656

估算方法 estimating methods

用于对项目的工作量、工期或成本进行近似测算所采用的方法。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.1657

估算 estimation

对可能数量或结果的进行定量评估的过程。

[来源：ISO/IEC/IEEE 24748-5:2017, 3.5]

3.1658

伦理政策声明 ethical policy statement

由最高管理层认可的高层次声明，用于解释和展示组织在开展活动时尊重核心价值观的承诺。

[来源：IEEE 7000:2021, 3.1]

3.1659

伦理原则 ethical principle

社区成员可以追求和维护的关于伦理价值观的共同主张。

[来源：IEEE 7000:2021, 3.1]

3.1660

伦理需求 ethical requirement.

伦理价值需求(EVR)或基于价值的系统需求。

[来源：IEEE 7000:2021, 3.1]

3.1661

伦理风险 ethical risk

对伦理价值观的风险。

[来源：IEEE 7000:2021, 3.1]

3.1662

伦理价值 ethical value

在人类文化背景下支持对是非判断的价值。

示例：美德是伦理价值的一个例子。

[来源：IEEE 7000:2021, 3.1]

3.1663

伦理价值需求 ethical value requirement (EVR)

满足利益相关者和概念性价值分析识别为与目标系统(SOI)相关的价值的组织或技术需求。

[来源：IEEE 7000:2021, 3.1]

3.1664

评价 evaluation

评估某物价值的行动。

示例：ODP 系统根据系统估计为某物分配相对状态的操作。

注：值可以用术语来考虑有用性、重要性、偏好、可接受性等；例如，评估目标可以是信用评级、系统状态、潜在行为。

[来源：ISO/IEC 15414:2015，6.6.7]

3.1665

评价活动 evaluation activity

利用适当的技术或方法，对系统或软件产品就已确定且适合的质量特性的目标值进行评估。

[来源：GB/T 25000.2-2018，4.2]

3.1666

评价检查表 evaluation checklist

问题列表，其中每个问题旨在检查产品、过程或服务是否符合特定国际标准中的一项或多项条款。

[来源：ISO/IEC 14143-2:2011，3.2]

3.1667

评价覆盖率 evaluation coverage

评价特定软件产品质量需求的覆盖程度。

[来源：GB/T 25000.40-2018，4.17]

3.1668

评价组 evaluation group

负责规定系统与软件质量需求，以及通过提供技术、工具、经验和管理技能来管理和实施质量评价活动的组织。

注：软件质量需求可以被评价请求方预先指定，评价组将验证软件质量需求的存在性和价值。

[来源：GB/T 25000.2-2018，4.3]

3.1669

评价级别 evaluation level

在评价期间所采用的严格度，该严格度根据所用的评价技术和所达到的评价结果界定评价的深度和彻底度。

[来源：GB/T 25000.40-2018，4.18]

3.1670

评价方法 evaluation method

描述评价方为获得适用于产品部件或整个产品规定的测量结果所采取行为的规程。

[来源：GB/T 25000.1-2021，3.11]

3.1671

评价模块 evaluation module; EVM

用于测量系统和软件质量特性、子特性或属性的评价技术包。

注：该包包含评价方法和技术、评价的输入、待测量和待收集的数据以及支撑过程和工具。

[来源：GB/T 25000.1-2021，3.12]

3.1672

评价程序 evaluation procedure

一系列任务和步骤，完成后使评估团队能够确定被评估的产品、过程或服务是否符合特定标准。

[来源：ISO/IEC 14143-2:2011, 3.3]

3.1673

评价记录 **evaluation records**

评价过程中所有的评价活动和取得的所有结果的客观记录证据。

[来源：GB/T 25000.40-2018, 4.21]

3.1674

评价报告 **evaluation report**

描述系统目标如何达成、识别遗留问题并旨在辅助未来开发的系统后续报告。是呈现评价结果和与评价相关的其他信息的文档。

[来源：ISO/IEC 2382:2015, X.X; ISO/IEC/IEEE 24765j:2021, X.X, 有修改]

。

3.1675

评价请求方 **evaluation requester**

请求评价的个体或组织。

[来源：GB/T 25000.40-2018, 4.22]

3.1676

评估（评价）发起方 **evaluation sponsor**

要求执行评估并提供财务或其他资源以实施评估的个人或组织。

[来源：ISO/IEC 14143-2:2011, 3.4]

3.1677

评价严格度 **evaluation stringency**

软件产品质量特性和子特性满足软件产品预期使用临界要求的程度。

[来源：GB/T 25000.40-2018, 4.24]

3.1678

评价技术 **evaluation technology**

用于评价的技术 **technology used for evaluation**

用于评价的技术、过程、工具、测量和相关技术信息。

示例：为开发方、需方或独立评价方设计的产品质量或使用质量测量、或者指定的评价过程。

[来源：GB/T 25000.2-2018, 4.4]

3.1679

评价工具 **evaluation tool**

在评价期间，可用于收集数据、解读数据或执行部分自动评价的设备。

注：这些工具的例子有：计算代码度量的源代码分析工具，生成形式化模型的 CASE 工具，运行可执行程序的测试环境，收集审查数据的检查列表，或者生成综合测度的报表。

[来源：GB/T 25000.40-2018, 4.23]

3.1680

评价方 **evaluator**

实施评价的个体或组织。

[来源：GB/T 25000.1-2021, 3.13]

3.1681

事件 **event**

对在时空上占有某一位置的重要现象的规格说明。在状态图语境中，事件是能触发某一转移的一种现象。

3.1682

事件历史 event history

用于表征重要动作发生历程的对象。

[来源：ISO/IEC 10746-3:2009, 13.1.1.1]

3.1683

事件记录 event record

代表事件属性的概念性数据类型。

[来源：IEEE 982:2024, 3.1]

3.1684

事件序列分析 event sequence analysis

为响应特定外部事件而执行的任务序列的性能分析。

[来源：ISO/IEC/IEEE FDIS 24765 3.1683]

3.1685

事件序列图 event sequence diagram

标识处理外部事件所需任务序列的图表。

[来源：ISO/IEC/IEEE FDIS 24765 3.1684]

3.1686

事件接收器 event sink

一种操作接口，用于接收携带特定类型事件通知的公告消息。

[来源：ISO/IEC/IEEE FDIS 24765 3.1685]

3.1687

事件源 event source

向事件消费者或事件通道发起事件通知的操作接口。

参见：事件接收器。

[来源：ISO/IEC/IEEE FDIS 24765 3.1686]

3.1688

事件同步 event synchronization

通过事件信号实现对任务激活的控制。

注：事件同步可分为三类：外部中断、定时器超时，以及来自其他任务的内部信号。

[来源：ISO/IEC/IEEE FDIS 24765 3.1687]

3.1689

事件跟踪 event trace

对每个外部输入及其发生时间按时间顺序排列的记录。

[来源：ISO/IEC/IEEE FDIS 24765 3.1688]

3.1690

事件驱动的评审或活动 event-driven review/activity

根据项目中某事件的发生而进行的评审或活动。

3.1691

事件排序逻辑 event-sequencing logic

任务如何响应其每个消息或事件输入的描述。

注：特别是每个输入产生的输出。

[来源：ISO/IEC/IEEE FDIS 24765 3.1689]

3.1692

证据 evidence

针对软件开发活动中留下的数据（如文档、各种形式的记录、证言等），进行计算得到的，用于表明活动的表现是否满足要求的测量数据。

[来源：GB/T 37970-2019]

3.1693

考核 examination

作为评价组成部分的机制，根据认证方案规定，通过笔试、口试、实操、观察等一种或多种方式测量被评价者的胜任力。

[来源：ISO/IEC 24773-1:2019, 3.10]

3.1694

异常 exception

导致正常程序执行挂起的事件。

注：类型包括：寻址异常(addressing exception)、数据异常(data exception)、操作异常(operation exception)、溢出异常(overflow exception)、保护异常(protection exception)和下溢异常(underflow exception)。

[来源：ISO/IEC/IEEE FDIS 24765 3.1693]

3.1695

异常处理 exception handling

通过抛出和捕获异常来传递错误信息的编程语言机制。

[来源：ISO/IEC/IEEE FDIS 24765 3.1694]

3.1696

可执行需求规范 executable requirements specification

以可执行需求语言表示的软件需求规格说明书。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1696]

3.1697

可执行源语句 executable source statement

在运行时指导计算机动作的源语句。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1697]

3.1698

可执行语句 executable statement

一种语句，该语句经编译后转换为目标代码；当测试项运行时，此目标代码会按过程式执行，且能够对程序数据执行相应操作。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.30]

3.1699

执行(动) execute

执行指令、过程或计算机程序。

[来源：ISO/IEC/IEEE FDIS 24765 3.1699]

3.1700

执行过程组 executing process group

为完成项目管理计划中定义的工作以满足项目规范而执行的过程集合。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3.1701

执行过程 **executing processes**

为完成项目管理计划中定义的工作以满足项目目标而执行的那些过程。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3.1702

执行（名） **execution**

由计算机运行计算机程序中一条或多条指令的过程。

3.1703

执行效率 **execution efficiency**

系统或组件以最少的时间消耗执行其指定功能的程度。参见：执行时间、存储效率。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1701]

3.1704

执行监控程序 **execution monitor**

与系统或组件并行运行的软件工具或硬件设备，用于监督、记录、分析或验证系统或组件的运行情况。

[来源：ISO/IEC/IEEE FDIS 24765 3.2885]

3.1705

执行时间 **execution time**

执行一个计算机程序所用的消逝时间或中央处理器所用时间。

参见：开销时间（overhead time），运行时间（run time）和运行的时间（running time）。

注：处理器时间通常小于消逝时间，因为处理器可能空闲（例如，等待所需要的资源）或在程序执行期间由别的任务使用。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1702]

3.1706

执行时间理论 **execution time theory**

采用累计执行时间作为估计软件可靠性基础的一种理论。

3.1707

执行踪迹 **execution trace**

代码踪迹 **code trace**

控制流踪迹 **control flow trace**

在计算机程序执行期间执行的指令序列的记录。

参见：回顾的踪迹（retrospective trace）、子例程踪迹（subroutine trace）、符号踪迹（symbolic trace）和变量踪迹（variable trace）。

注：通常以程序执行时遇到的代码标签列表的形式呈现。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1703]

3.1708

执行状态 **executive state**

监督状态 **supervisory state**

特权状态 **privileged state**

在计算机系统运行中，监控程序正在执行的一种状态。

参见：问题状态。

注：该状态通常比应用程序具有更高优先级，并禁止应用程序执行。

[来源：ISO/IEC/IEEE FDIS 24765 3.4614]

3.1709

控制程序 executive; supervisor

控制程序 control program

监督程序 executive program

监督器 supervisory program

一种计算机程序，通常是操作系统的一部分，用于控制其他计算机程序的执行，并调节计算机系统的工作流程。

参见：监督状态、内核。

[来源：ISO/IEC/IEEE FDIS 24765 3.4615]

3.1710

执行器/执行者 executor

一种软件，用于管理指定平台上程序、作业或步骤的运行。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1711

穷尽测试 exhaustive testing

对输入值和前置条件的所有组合都进行测试的测试方法。

注：在几乎所有复杂场景下，由于可能的测试用例数量巨大，穷尽测试是不可行的。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.34]

3.1712

存在约束 existence constraint

约束说明一个实体的实例不能存在，除非另一个相关实体的实例也存在。

注：[关键样式]。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.159]

3.1713

存在依赖 existence dependency

两个相关实体之间的约束，表明一个实体的实例不与另一个实体的实例相关联。

注：以下关联类型表示存在依赖关系：识别关系、分类结构和强制性非识别关系。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1707]

3.1714

现有软件 existing software

已开发完成且可获取的软件，可原样使用或经修改后使用，由供方、需方或第三方提供。

[来源：ISO/IEC/IEEE FDIS 24765 3.1708]

3.1715

出口 exit

软件模块中，模块的执行能终止的点。与入口点（entry point）相对。

参见：返回（return）。

[来源：X.X]

3.1716

退出动作 exit action

退出状态机的某一状态（而不管退出该状态采取何种转移）所执行的一种动作。

[来源: X. X]

3.1717

退出标准 **exit criteria**

努力成功结束之前必须存在的状态。

[来源: ISO/IEC/IEEE FDIS 24765 : 2025, 3.1710]

2、可以声明在技术审查或审计成功完成之前必须评估、完成和关闭的审查或审计要素。

参见: 进入标准。

[来源: ISO/IEC/IEEE 24748-8:2019 , 3.1]

3.1718

退出点/出口点 **exit point**

测试项中的最后一个可执行语句。

注: 出口点是终点通过测试项的路径, 是测试项中的可执行语句, 它要么终止测试项, 要么将控制权返回给外部进程。这通常是测试项中的最后一个可执行语句。

[来源: GB/T 38634.4-2020 , 3.31]

3.1719

退出例程 **exit routine**

当规定的事件(例如错误)发生时接收控制的例程。

[来源: ISO/IEC/IEEE FDIS 24765 : 2025, 3.1712]

3.1720

可扩充性 **expandability**

改进或修改软件功能效率所需付出的努力程度。

参见: 可扩展性。

[来源: ISO/IEC/IEEE FDIS 24765 3.1713]

3.1721

预期结果 **expected result**

根据规格说明或其他来源, 在特定的条件下可获取的测试项的预期行为。

[来源: GB/T 38634.1-2020, 3.15]

3.1722

期望值 **expected value**

估算的结果, 该结果被超过和不被超过的可能性相同。

注释: 概率分布的均值, 累积概率函数等于 0.5 的点。

同义词: 50-50 估算。

[来源: ISO/IEC/IEEE 24765:2017 系统与软件工程——词汇表]

3.1723

等概率期望值 **expected value 50-50; estimate; 50-50 estimate**

被超出与不被超出的可能性均等的估算结果。

注: 概率分布的均值, 即累积概率函数等于 0.5 的点。

[来源: ISO/IEC/IEEE FDIS 24765 3.1716]

3.1724

完美信息的期望值 **expected value of perfect information**

在决策树分析中, 决策树期望值与如果提前知道所有随机结果时决策树价值之间的差值。

注: 帮助决策者确定是否值得投资于减少不确定性的活动。

[来源: ISO/IEC/IEEE 24765:2017, X. X]

3.1725

基于经验的测试 **experience-based testing**

一类基于测试人员经验设计生成测试用例的测试用例设计技术。

示例: 错误推测法。

注: 基于经验的测试可涵盖测试攻击、测试遍历、错误分类法等理念, 这些理念针对安全、性能及其他质量维度的潜在问题。

[来源: ISO/IEC/IEEE 29119-4:2021, 3.33]

3.1726

专家系统 **expert system; ES**

计算机系统, 通过从由人类专业知识开发的知识库中进行推理, 从而在给定领域或应用领域内提供专家级的问题解决。

注: 一些专家系统能够改进其知识库, 并根据其以往问题的经验开发新的推理规则。

[来源: ISO/IEC 2382:2015, X. X]

3.1727

专家系统 (ES) **expert system (ES); ES**

计算机系统, 通过从人类专业知识开发的知识库中得出推论, 专业地解决给定领域或应用领域中的问题。

注: 一些专家系统能够改进他们的知识库并根据他们对以前问题的经验开发新的推理规则。

[来源: ISO/IEC 2382:2015, X. X]

3.1728

可解释性 **explainability**

描述自主智能系统如何做出判定的能力。

注: 解释既可以反映自主/智能系统遵循的规程 (即其输入、方法、模型和输出), 也可以反映该系统做出的具体判定。

[来源: IEEE 7014-2024, 3.1]

3.1729

解释性报告 **explanatory report**

附属于某一产品, 为其提供补充信息, 以帮助对其了解并避免使用不当的文档。

注1: 解释性报告的示例有数据元素界定、数据总量统计、数据源, 这些信息均附属于基准度量数据仓库或基准。

注2: 产品示例有基准度量数据仓库、基准, 或支持基准度量活动的软件工具。

[来源: GB/T 41866.1-2022, 3.14]

3.1730

显式地址 **explicit address**

永久分配给设备或存储位置的地址, 无需转换或计算即可标识该设备或位置。

参见: 相对地址 (relative address)、可重定位地址 (relocatable address)、符号地址 (symbolic address)、绝对汇编程序 (absolute assembler)、绝对代码 (absolute code)、绝对指令 (absolute instruction)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.25]

3.1731

显性知识 **explicit knowledge**

可通过文字、数字、图像等符号进行编码的知识。

[来源：《项目管理知识体系指南》(PMBOK® 指南) 第七版]

3.1732

显式需求 explicit requirement

以书面形式表达的需求，通常使用自然语言或半形式化的记法。

[来源：IEEE 982:2024, 3.1]

3.1733

可利用的 exploitable

可在实际场景中被触发的漏洞特性。

[来源：ISO/IEC 23643:2020, 3.36]

3.1734

探索性测试 exploratory testing

一种基于经验的测试。测试者基于其现有的相关知识、测试项的前期探索（包括以前的测试结果）以及关于通常软件行为和故障类型的启发式“经验法则”，自发地设计和执行测试。

注：探索性测试寻找隐含属性（包括隐含的行为），虽然其自身存在危害的可能性很小，但其可能干扰待测软件的其他属性，并因此产生软件失效的风险。

[来源：GB/T 38634.1-2020, 3.16]

3.1735

引出 export

在包的语境中，使某一元素在其环绕的命名空间之外可见。

参见：可见性（visibility），与引入（import）相对。

[来源：GB/T 11457-2006, 2.591]

3.1736

导出过程 export process

从源环境生成传输文件的过程。

[来源：ISO/IEC 15474-1:2002, 4.1]

3.1737

表达式 expression

一种对某一特别类型求值的串。

示例：表达式“7+5*3”求“数”类型的值。

3.1738

扩展 extend

在UML中，指从扩展用例指向基础用例的一种关系，用于规定如何将扩展用例定义的行为选择性嵌入到基础用例定义的行为中。

[来源：ISO/IEC/IEEE FDIS 24765 3.1729]

3.1739

可扩展性 extendability

可扩充性 expandability

系统或部件能修订以增加它的存储或功能的能力的情况。

参见：灵活性（flexibility）和可维护性（maintainability）。

3.1740

扩展组件 extended component

提供任何类型端口的组件。

[来源：ISO/IEC 19500-3:2012, 4.1]

3.1741

扩展元素 **extended element**

标签中的元素，提供标准中明确记录的信息之外的附加信息。

[来源：ISO/IEC/IEEE 24765:2017, 3.1529]

3.1742

扩展条目表 **extended entry table**

一种判定表，其中条件与动作仅作概括性描述，内容不完整。

注：规范内容由规则中规定的取值予以完善。

[来源：ISO 5806:1984, 3.15]

3.1743

扩展过程集 **extended process set**

成熟度水平高于基本成熟度水平的过程集，有助于确保相关过程概况的实现。

[来源：ISO/IEC 33001:2015, 3.3.5]

3.1744

可扩展标记语言 **extensible markup language; XML**

与平台无关的标记语言，包含生成含有结构化数据的文本格式的规则。

注：在 XML 模式第 1.1 部分结构建议中规定。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.14]

3.1745

可扩展样式表语言转换 **Extensible Style Language Transformations; XSLT**

将 XML 文档转换为其他文档类型（如 PDF 或 HTML）的语言。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1733]

3.1746

扩展设置 **extensional set**

当前范围 **current extent**

包含类当前存在实例的集合。

注：扩展集中的实例对应实例的数据库和数据建模概念。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1734]

3.1747

外延集 **extensional set**

当前外延 **current extent**

包含一个当前存在所有实例集合的类。

注：外延集中的实例对应数据库与数据建模领域中的实例概念。

[来源：IEEE 1320.2-1998 (R2004), 3.1.61]

3.1748

广义逻辑操作 **extensive logical operation**

包含至少四个嵌套级别，或运行该逻辑运算需要超过 38 种数据元素类型的逻辑运算，或两者兼而有之。

注：数据元素类型不一定非要跨越应用程序边界。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.11]

3.1749

广义数学操作 extensive mathematical operation

数学运算，包括一系列或多系列数学方程和计算，与逻辑运算符配合执行或根据逻辑运算符执行，以产生用户可识别的结果。

示例：用于计算项目预期完成日期的计划评审技术（PERT）。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.10]

3.1750

外部的 external

超出项目生存周期范围的输入信息源或输出信息目的地。比照。调用、迭代、映射。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1737]

3.1751

外部属性 external attribute

一个实体的可测量属性，只能从它与环境的关系中推导出来。

注：外部属性是那些与需求相关的属性（软件的外部属性）。外部属性只能从它所属的系统的操作行为中导出。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1738]

3.1752

外部依赖 external dependency

项目活动与非项目活动之间的关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1739]

3.1753

外部事件 external event

来自外部对象的事件，通常是来自外部 I/O 设备的中断。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1740]

3.1754

外部输入输出设备 external I/O device

在软件系统之外，属于外部环境一部分的硬件输入和/或输出设备。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1741]

3.1755

外部输入 external input; EI

处理或控制来自应用程序边界之外的数据或信息的基本过程。

注：外部输入是一种基本功能部件。

[来源：GB/T 42449-2023]

3.1756

外部查询 external inquiry; EQ

向应用程序边界之外发送数据或控制信息的基本过程。

注1：外部查询是一种基本功能部件；

注2：应用程序在此过程中生成一个大小完全确定的输出，不需要进行加工处理。

[来源：GB/T 42449-2023]

3.1757

外部接口文件 external interface file; EIF

用户可识别的一组逻辑相关数据或控制信息，被测量的应用程序所引用，但是在另一个应用程序的边界内维护。

注：外部接口文件是一种基本功能部件。

[来源：GB/T 42449-2023]

3.1758

外部接口需求 external interface requirement

系统需求，规定了系统或组件与之接口的硬件、软件或数据库元素，或者规定了由这种接口引起的对格式、时序或其他因素的约束。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1759

外部逻辑文件 external logical file; ELF

从用户的角度来看的一组永久数据，由一个应用程序使用，但由另一个应用程序维护。

[来源：GB/T 42588-2023, X.X]

3.1760

外部测量 external measure

产品的间接测量源自对其所属系统的行为的测量。

注：测试期间发现的故障数是程序中故障数的外部测量，因为数量在运行该程序的计算机系统运行期间计算失败次数。外部测量可用于评估更接近设计最终目标的质量属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1746]

3.1761

外部测度（系统与软件质量） external measure (of system or software quality)

包含软件产品的系统在指定条件下使用时，该软件产品使得系统行为满足明确和隐含要求的程度的测度。

注1：这种行为，在测试和运行期间，通过执行此软件产品能加以验证和/或确认。

注2：基于 GB/T 25000.1-2010 中对软件外部质量的定义。

注3：此定义改编自 GB/T 25000.10-2016。

3.1762

软件质量外部测度 external measure of software quality

包含软件产品的系统在指定条件下使用时，该软件产品使得系统行为满足明确和隐含要求的程度的测度。

注1：这种行为，在测试和运行期间，通过执行此软件产品能加以验证和/或确认。

注2：基于 GB/T 25000.1-2010 中对软件外部质量的定义。

注3：此定义改编自 GB/T 25000.10-2016。

[来源：GB/T 25000.21-2019, 3.2]

3.1763

系统和软件质量的外部测度 external measure of system or software quality

对于在特定条件下使用的系统或软件产品，其行为能满足明确的和隐含的需求的程度的测度。

示例：相对于在计算机系统中呈现的故障数而言，在测试期间发现的失效数是软件质量的外部测度。这两种测度不必要等同，因为测试不能发现所有的故障，并且在不同的境况下，一个故障会引发不同的失效。

注：在测试和运行期间，通过执行系统或软件产品而验证和（或）确认行为的属性。

[来源：GB/T 25000.1-2021, 3.1]

3.1764

外部输出 external output; E0

向应用程序边界之外发送数据或控制信息的基本过程，包括外部查询之外的额外处理逻辑。

注：外部输出是一种基本功能部件。

[来源：GB/T 42449-2023]

3.1765

外部质量 external quality

产品在特定条件下使用时满足明示和隐含需求的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1749]

3.1766

外部软件质量 external software quality

当系统在特定条件下使用时，软件产品使系统的行为能够满足明示和隐含需求的能力。参见：软件质量的外部度量、软件内部质量。

注：行为的属性可以通过在测试和操作期间执行软件产品来验证和/或验证。

[来源：ISO/IEC 25000:2014]

3.1767

外部可变性 external variability

客户可见的可变性。

[来源：ISO/IEC 26555:2015, 3.4]

3.1768

提取法 extractive approach

从一个或多个现有产品构建产品线初始基线的方法。

[来源：ISO/IEC 26553:2018, 3.13]

3.1769

外联网 extranet

可供授权外部用户访问，用于信息检索或信息交换的企业内部网。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.7]

3.1770

极限编程 extreme programming; XP

敏捷软件开发的一种形式，其中迭代策划和工作的规程与技术规程（如测试驱动设计和结对编程）相结合。

[来源：ISO/IEC TR 7052:2023, 3.24]

3.1771

F1 分数 F1 score; F1-score

用于评估分类器的性能指标，它在召回率和精确率之间提供平衡（调和平均）。

[来源：ISO/IEC TR 29119-11:2020, 3.1.33]

3.1772

面 facet

计算组件扮演服务器角色的操作接口。

[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：体系结构, 7.1.16]

[来源：条目注释 1：组件在正常执行期间向客户端暴露其功能应用行为的主要途径]

3.1773

分面搜索 faceted search

一种渐进式搜索，允许用户通过一个或多个属性选定取值来缩小结果范围。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.15]

3.1774

引导式研讨会 facilitated workshops

把主要的跨职能干系人召集在一起，通过集中讨论 来定义产品需求的一种启发式技术。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版）]

3.1775

设施 facility

促进行动执行的物理手段或设备，例如厂房、仪器、工具。

[来源：GB/T 22032-2021/ISO/IEC/IEEE 15288:2015 系统和软件工程 系统生存周期过程，4.1.20]

3.1776

分解 factoring

将系统分解为一组模块的过程。

2. 从一个模块中移除某个功能，并将其放入独立的模块中。参见：模块化分解，重构。

[来源：ISO/IEC/IEEE 24765 系统与软件工程——词汇表，3.1760]

3.1777

工厂 factory

某对象可响应其外部环境发起的交互行为，创建新对象后，将新对象的引用返回给该环境。

[来源：ISO/IEC 10746-2:2009，9.25]

3.1778

失效安全 fail safe

产品在发生故障时，自动将自身置于安全运行模式，或恢复到安全状态的能力。

产品在发生故障时，自动将自身置于安全运行模式，或恢复到安全状态的程度。

参见：故障弱化、故障安全、故障容忍。

示例：正常运行失效时，交通信号灯所有方向均恢复为红灯闪烁。

[来源：ISO/IEC 25010:2023，3.9.3]

3.1779

故障弱化 fail soft

系统或组件在发生某些失效时仍继续提供部分运行能力的特性。

参见：失效保险、故障安全、故障容忍。

示例：黄灯失效时，交通信号灯仍可继续在红灯与绿灯之间交替切换。

[来源：ISO/IEC/IEEE FDIS 24765，3.1762]

3.1780

失效 failure

系统行为与系统需求的偏离。

注：当遇到故障时可能会产生失效。失效可能在超出规格说明中规定的最低要求值时发生，即超过设计限值或超出安全范围。

[来源：IEEE 982:2024，3.1]

3.1781

失效类别 failure category

错误、故障或失效可能归并到其中的一组类别之一，当错误、故障或失效发生或发现后，可根据其原因、危急程度、效果、故障所属的生存周期阶段或其他特性而确定其类别。

参见：XX 条，出错类别（error category）。

3.1782

失效强度 failure intensity

运行中可靠性的定量表征，为观察到的失效数与观察时间跨度之比。

参见：故障率。

[来源：IEEE 982:2024, 3.1]

3.1783

失效强度目标 failure intensity objective; FIO

作为系统发布准则的一部分，在发布前测试期间要达到的失效强度级别。

[来源：IEEE 982:2024, 3.1]

3.1784

失效模式 failure mode

失效在物理或功能上的表现形式。

示例：处于失效模式下的系统表现为运行缓慢、输出错误或执行完全终止。

[来源：ISO/IEC/IEEE FDIS 24765, 3, 1767]

3.1785

失效方式和影响分析 failure mode and effect analysis; FMEA

通过对产品各组件潜在失效模式及其对组件、产品或系统功能和可靠性的影响进行系统分析，并评估降低失效发生概率及减轻其影响措施，或在系统或更低层级上，检查产品可能发生失效的所有方式的一种分析方法。

注：对每一潜在失效，评估其对整个系统的影响与后果。此外，对旨在降低失效发生概率并减小其影响的计划措施进行评审。

[来源：ISO/IEC/IEEE 24765, 3.1768]

3.1786

故障模式和影响分析 Failure Mode and Effects Analysis; FMEA

一种分析性程序，其中对产品每个组成部件中的每一种潜在失效模式进行分析，以确定该失效模式对该部件可靠性的影响，以及该失效模式单独或与其他可能的失效模式组合时，对产品或系统可靠性以及该部件所需功能的影响；或者，在系统层级或更低层级，对产品进行审查，以识别所有可能发生失效的方式。

注：对于每一种潜在失效，应估计其对整个系统的影响及其影响程度。此外，还应评审为降低失效发生概率以及减轻其后果而计划采取的措施。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1768]

3.1787

失效率 failure rate**失效比** failure ratio

a) 失效数与给定测量单位的比率。例如，每单位时间的失效次数、若干次事务处理中的失效次数，若干次计算机运行中的失效次数。

b) 根据某一模型及其参数表示的单位时间内失效数量的变化率。

参见：失效强度。

示例：每单位时间的失效次数、若干次事务处理中的失效次数，若干次计算机运行中的失效次数。

[来源：IEEE 982:2024, 3.1]

3.1788

故障恢复 failure recovery

系统失效后又回到可靠的运行状态。

3.1789

失效严重性 *failure severity*

在对失效的有害影响进行排序的标度上的步长。

[来源：IEEE 982:2024, 3.1]

3.1790

故障透明性 *failure transparency*

向对象屏蔽其他对象（或自身）的故障及可能的恢复过程，从而实现容错能力的一种分布式透明机制。

[来源：ISO/IEC 10746-3:2009, 4.4.1.2]

3.1791

备用预案 *fallback plan*

包含一组备用的行动和任务，以便在主计划因问题、风险或其他原因而废弃时采用。

[来源：项目管理知识体系指南(PMBOK® 指南)(第5版)]

3.1792

假阴性 *false negative; FN*

实际结果为通过，却错误地报告为失效。

参见：假阳性、真阴性。

[来源：ISO/IEC TR 29119-11:2020, 3.1.34]

3.1793

假阳性 *false positive*

实际为失效，却错误地报告为通过。

参见：假阴性、真阳性。

注：由于分析规则精确度不足，在应用程序分析过程中可能出现假阳性。

[来源：ISO/IEC TR 29119-11:2020, 3.1.35]

3.1794

程序系列 *families of programs*

通过共享需求、设计和代码的重要部分而相互关联的程序集。

注：一个程序系列可能包括为英语受众开发的一个程序版本、为德语受众开发的第二个程序版本，以及为日语受众开发的第三个程序版本。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1788]

3.1795

平台族 *family of platforms*

具有相同用途的软件平台。

示例：操作系统、浏览器。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.16]

3.1796

快速跟进 *fast tracking*

一种进度压缩方法，将正常情况下按顺序进行的活动或阶段改为至少部分按并行方式开展。

参见：进度压缩、赶工。

[来源：项目管理知识体系指南(PMBOK® 指南)第七版]

3.1797

致命错误 fatal error

导致系统或部件完全丧失功能的差错。

3.1798

故障 fault; bug

<信息技术>系统或系统表示中的缺陷，若被执行或触发，可能会导致错误产生。

<元器件>硬件设备或部件中的缺陷。

注：故障一旦被触发，可能导致失效。规格说明中若存在不正确内容，也可能产生故障。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1796]

3.1799

故障字典 fault dictionary

系统或部件中的故障清单和已经指定进行检测的测试清单。

3.1800

故障隔离 fault isolation

子系统防止其内部故障在其他子系统中引发继发故障的能力。

[来源：ISO/IEC/IEEE 24765c:2014]

3.1801

故障屏蔽 fault masking

一种故障阻碍对另一种故障的检测的状况。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1799]

3.1802

故障安全 fault secure

系统或组件在预先规定的故障集下，不会产生故障的属性。

参见：故障容忍、失效安全、故障弱化。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1800]

3.1803

故障注入 fault seeding

缺陷播种 bug seeding

故障插入 fault insertion

故意向计算机程序中已有的故障加入已知故障，以便监测故障的检测率与清除率，并估算程序中剩余故障数量。

参见：内在差错。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1640]

3.1804

故障容忍（名词） fault tolerance

即使在硬件或软件故障存在的条件下，产品仍保持其预期运行功能的能力。

参见：错误容错、失效安全、故障弱化、故障安全、健壮性。

参见：错误容错、失效安全、故障弱化、故障安全、健壮性。

[来源：ISO/IEC 25010:202, 3.5.3]

[来源：故障容错]

[来源：即使在硬件或软件故障存在的条件下，产品仍保持其预期运行功能的能力。]

[来源：ISO/IEC 25010:2023 系统和软件工程 系统与软件质量要求和评价（SQuaRE）产品质量模

型, 3.5.3]

3.1805

故障树分析 **Fault tree analysis; FTA**

无 (仅是缩略语)。

3.1806

无故障性 **faultlessness**

产品在正常操作下无故障执行特定功能的能力。

[来源: ISO/IEC 25010:2023, 3.5.1]

3.1807

可行性研究 **feasibility study**

研究识别和分析问题及其潜在解决方案, 以确定其可行性、成本和收益。

3.1808 [[来源: ISO/IEC 2382:2015, X.X]

特征 **feature**

系统的功能或非功能区分特性。

注: 特性被认为可为用户增加价值。特性可: a) 体现产品线内成员产品之间客户可见或最终用户可见的差异性; 或 b) 体现除价格、性能、噪声、重量、能耗等非功能差异外, 客户或最终用户无法直接直观看到的实现差异性。在基于特性的产品线工程中, 特性用于体现成员产品之间的差异。产品线内所有成员产品共有的能力或其他特性, 不建模为特性。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.9]

3.1809

特征分支 **feature branch**

为开发特定特征集而创建的分支。

注: 该分支通常不发布, 而是在某个阶段合并回其父分支。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1811]

3.1810

特征目录 **feature catalog; feature catalogue**

覆盖整个产品线的所有特征选项及特征约束的集合的模型。

[来源: ISO/IEC 26580:2021, 3.5]

3.1811

功能目录 **feature catalogue**

在整个产品线范围内所有可用特征选项和特征约束的集合模型。

[来源: ISO/IEC 26580:2021, 3.5]

3.1812

特征约束 **feature constraint**

在所有成员产品中必须满足的、两个或多个特征之间的正式关系。

[来源: ISO/IEC 26580:2021, 3.6]

3.1813

特征工程 **feature engineering; feature selection**

在机器学习中, 从原始数据中识别出最能代表模型应体现的潜在关系属性, 以用于训练数据的活动。

[来源: ISO/IEC TR 29119-11:2020, 3.1.36]

3.1814

特征冻结 **feature freeze**

不向特定分支添加任何新特征的阶段。

注：使分支趋于稳定以便发布。

[来源：ISO/IEC/IEEE 24765:2017, 3.1815]

3.1815

特征语言 **feature language**

用于对特征目录、特征清单组合以及共享资产超集的可变点中的概念与构造进行形式化表示、结构化分类，并描述其间关系的语法和语义。

[来源：ISO/IEC 26580:2021, 3.7]

3.1816

特征引用 **feature reference**

在图表中无歧义地标识一个图表特征的表达式。

[来源：IEEE 1320.1-1998 (R2004), 2.1.48]

3.1817

特征集 **feature set**

包含测试项的测试条件的集合，可以从风险、需求、功能、模型等方面收集得到。

注：可能是该项的所有特征（其全部特征集），或为特定的目的而标识的子集（功能特征集等）。

[来源：GB/T 38634.1-2020, 1.37]

3.1818

完成到完成关系 **FF; finish-to-finish; FF**

一种约束关系，即除非另一活动已完成，否则当前活动不能完成。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1836]

3.1819

应用领域（规格） **field of application (of a specification)**

ODP 系统环境必须具备的属性，才能使用该系统的规范。

[来源：ISO/IEC 15414:2015 , 6.1.2]

3.1820

现场可编程门阵列 **field programmable gate array; FPGA**

设计为在获取后进行编程的逻辑器件。

注：通常基于查找表架构。

[来源：ISO/IEC 24765 FDIS, 3.1835]

3.1821

现场总线 **fieldbus**

用于实时分布式控制的工业计算机网络协议。

注：相关标准化接口族。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1826]

3.1822

先进先出队列 **FIFO**

一种特殊的队列，按先进先出方式操作。

[来源：ISO/IEC 15909-3:2021, 3.3]

3.1823

第五代语言 **fifth-generation language; 5GL**

融合了基于知识的系统、专家系统、推理引擎和自然语言处理概念的计算机语言。

[来源: ISO/IEC 24765 FDIS, 3.1829]

3.1824

第五代语言 (5GL) fifth-generation language (5GL); 5GL

包含基于知识的系统、专家系统、推理引擎和自然语言处理概念的计算机语言。

参见: 汇编语言、第四代语言、高阶语言、机器语言。

[来源: ISO/IEC/IEEE FDIS 24765 : 2025, 3.1829]

3.1825

文字式常数 figurative constant

编程语言中为特定常量保留的数据名称。

参见: literal。

示例: 保留数据名称 THREE 以表示值 3。

[来源: ISO/IEC/IEEE FDIS 24765 : 2025]

3.1826

引用的文件类型 file type referenced; FTR

由事务功能读取或维护的数据功能。

注: 文件类型引用包括由事务功能读取或维护的内部逻辑文件, 或由事务功能读取的外部接口文件。代码数据被归入一个额外的文件类型引用中。

[来源: ISO/IEC 20926:2009, 3.31]

3.1827

引用文件类型 (FTR) file type referenced; (FTR); FTR

由事务功能读取或维护的数据功能。

注: 被引用文件类型包括: 由事务功能读取或维护的内部逻辑文件, 或由事务功能读取的外部接口文件。代码数据被归为一个单独的 FTR。

[来源: ISO/IEC 20926:2009, 3.31]

3.1828

决算 final accounts

委托方在项目验收后对项目的成本进行计算的过程。

[来源: GB/T 36964-2018, 3.32]

3.1829

末期功能点分析 final function point analysis

为确定项目结束时的功能点数进行的计数。这个计数可以记录下被定义或者被安装起来的应用程序的规模, 或者能够确定所执行的项目的规模。

[来源: GB/T 42588-2023]

3.1830

完成-完成 finish-to-finish; FF

一项活动只有在另一项活动完成后才能完成的约束。

[来源: ISO/IEC 24765 FDIS, 3.1836]

3.1831

完成-开始 finish-to-start; FS

一项活动只有在另一项活动完成后才能开始的约束。

[来源: ISO/IEC 24765 FDIS, 3.1837]

3.1832

有限状态机 **finite state machine**

由有限数量的状态和这些状态之间的转换组成的计算模型，可能还有伴随的动作。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1838]

3.1833

固定总价合同 **firm-fixed-price contract; FFP contract**

一种固定总价合同类型，即买方向卖方支付约定金额（由合同规定），无论卖方成本如何。

[来源：项目管理知识体系指南（PMBOK® 指南）—第七版]

3.1834

固件 **firmware**

硬件设备与作为只读软件驻留在该设备上的计算机指令及数据的组合。

[来源：ISO/IEC/IEEE FDIS 24765 3.1842]

3.1835

第一个输入例程 **first input routine**

那些需要获取逻辑记录的活动，如果有的话，先处理。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1843]

3.1836

第一范式 **first normal form**

规范化过程的结果，该过程对数据组进行转换，使其具有唯一标识符、一个或多个属性，且不存在重复属性。

[来源：ISO/IEC/IEEE 24765a:2011]

3.1837

一线软件经理 **first-line software manager**

对由软件工程师和其他有关人员组成的单位（例如一个部门或项目组）的人员配备和活动负有直接管理责任（包括提供技术指导及人事和工资管理等职能）的负责人。

3.1838

适用性 **fitness for purpose**

系统、服务或算法满足其约定需求并可靠运行的保证。

[来源：IEEE 7014-2024, 3.1]

3.1839

固定成本 **fixed cost**

与生产率无关的成本。

参见：可变成本。

注：例如设施成本或贷款利息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1846]

3.1840

固定工期 **fixed duration**

完成活动所需时间不随分配人员或资源数量变化的活动类型。

[来源：项目管理知识体系指南（PMBOK® 指南）—第七版]

3.1841

带有经济价格调整条款的固定总价合同 **fixed price with economic price adjustment contract; FP-EPA contract**

带经济价格调整的固定总价合同，即允许因条件变化（如通货膨胀变化或特定商品成本增减）对合同价格进行预定最终调整的特殊条款的固定总价合同。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1842

固定总价加激励费用合同 fixed-price-incentive-fee contract; FPIF contract

固定总价加激励费合同，即买方向卖方支付约定金额（由合同规定），若卖方达到既定的绩效标准，还可获得额外奖励的合同类型。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.1843

标识 flag

根据处理结果或预定条件的发生而被设置为特定状态值（通常为真/假）的状态标记变量。

参见：指示器、信号量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1854]

3.1844

闪存 flash memory NVRAM

非易失性随机存取存储器 non-volatile random access memory

一种更大容量且速度更快的可编程只读存储器，它允许通过电方式擦除存储器中的数据，并能多次重写（参见电可擦可编程只读存储器（EEPROM）所具备的灵活性）。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1855]

3.1845

触发器 flip-flop

锁存器 latch

具有一个或两个稳定状态的电子电路。

注：可用于存储 0 或 1 作为数字数据。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1857]

3.1846

浮动时间 float

松弛时间 slack

非关键路径上的两个连续活动之间未计划时间量，可用于延迟前置活动的完成时间，或提前后续活动的开始时间。

参见：自由浮动时间（Free Float）、总浮动时间（Total Float）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1858]

3.1847

流 flow

一系列交互的抽象，其结果是信息从生产者对象传递到消费者对象。

[来源：ISO/IEC 10746-3:2009, 7.1]

2. 工作通过给定过程或框架移动效率的度量。

示例：多媒体数据广播。

注：流可用于对一系列交互的确切结构进行抽象，也可用于对连续交互（包括模拟信息流的特殊情况）进行抽象。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1859]

3.1848

控制流 flow of control; control flow

计算机程序执行过程中操作所遵循的顺序。

[来源: ISO/IEC/IEEE FDIS 24765, 3.993]

3.1849

流程图 flowchart; flow chart; flow diagram

以图表形式描绘系统内一个或多个流程的输入、处理动作和输出。

参见: 方块图、框图、气泡图、图表、输入 - 处理 - 输出图、结构图。

[来源: ISO/IEC/IEEE FDIS 24765 3.1860]

3.1850

流程图生成工具 flowcharter

一种以程序的设计表示或代码表示作为输入, 并输出该程序流程图的软件工具。

[来源: ISO/IEC/IEEE FDIS 24765 3.1861]

3.1851

绘制流程图 flowcharting

以图表形式展示一个或多个系统内流程的输入、处理步骤以及输出情况。

[来源: 《项目管理知识体系指南 (PMBOK(R) 指南) ——第七版》]

3.1852

焦点类 focus class

对进行支持的一个或多个辅助类定义核心逻辑或控制流的一种衍型类。在典型情况下, 焦点类与一个或多个辅助类一起使用, 在设计阶段, 对规定构件的核心业务逻辑或控制流特别有用。

参见: 2.114 条 辅助类 (auxiliary class)。

3.1853

控制聚焦 focus of control

在时序图中, 展现如下时间段的一种符号: 在这一时间段, 某一对象正直接地或通过下级过程完成一个动作。

3.1854

仅作说明页 for Exposition Only page; FE0 page

包含关于特定图表的图形信息 (而非文本) 的模型页面。

注: 与图表不同, 仅供展示页面 (FE0 页面) 的内容无需遵循 IDEF0 规则。

[来源: ISO/IEC 24765 FDIS, 3.1871]

3.1855

前台 foreground

在作业调度中, 高优先级进程或需要用户交互的进程执行所处的计算环境。

参见: 后台处理、前台处理。

[来源: ISO/IEC/IEEE FDIS 24765 3.1873]

3.1856

前台处理 foreground processing

执行高优先级进程而低优先级进程等待计算机资源的可用性, 或执行需要用户交互的进程。参见: 后台处理。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1874]

3.1857

外键 foreign key migrated key

迁移键

子实体实例或类别实体实例的属性或属性组合，其值与相关父实体实例或通用实体实例的主键中的值匹配。

注：外键是通过泛化结构或关系迁移父实体或通用实体的主键产生的。[按键样式]。

[来源：ISO/IEC/IEEE FDIS 24765 : 2025, 3.1875]

3.1858

外键 foreign key; migrated key

子实体或类别实体实例的属性（或属性组合），其值与相关父实体或通用实体实例的主键中的值相匹配。

注：外键是通过泛化结构或关系将父实体或通用实体的主键迁移而来。[键样式]。

[来源：ISO/IEC/IEEE FDIS 24765 3.1875]

3.1859

分支操作 forking action

分支操作，在此动作中，已启用的链（可能会出现故障）最终会相互合并，即已启用的链不能与其他链合并，且不能单独终止。

[来源：ISO/IEC 10746-2:2009, 13.1.5]

3.1860

形式 form

收集数据的模块或公式。

注：它可以是纸质的（纸质的）形式或数字。

[来源：ISO/IEC 25024:2015, 4.20]

3.1861

形式、适配和功能 form, fit, and function

在配置管理中，构成某一项目作为整体实体所具有的物理特性和功能特性的配置，但不包括项目内部各组成元素本身的具体特性。

参见：配置标识。

[来源：ISO/IEC/IEEE FDIS 24765 3.1878]

3.1862

形式化设计 formal design

使用形式化方法进行软件设计的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1879]

3.1863

正式评估过程 formal evaluation process

根据既定标准评估替代解决方案以确定解决问题的推荐解决方案的结构化方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1880]

3.1864

形式语言 formal language

人工语言 artificial language

一种其使用规则在投入使用之前已被明确规定的语言。

参见：自然语言。

示例：编程语言和数学语言。

[来源：ISO/IEC 2382:2015, X. X]

3.1865

形参 formal parameter

软件模块中使用的变量，表示要由调用模块传递给模块的数据或程序元素。

参见：参数。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1882]

3.1866

正式合格评审 formal qualification review; FQR

通过测试、检查或分析等过程，对构成某一系统的一组配置项进行验证，以确认其已经满足特定合同规定的性能要求。

参见：代码评审、设计评审、需求评审、测试就绪评审。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1883]

3.1867

形式化需求语言 formal requirements language

可验证需求语言 verifiable requirements language

用于表示软件需求的人工语言。

注：所生成的正式需求可以通过正确性证明方法来验证其“正确性”。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1884]

3.1868

正式审查 formal review

遵循具有正式文件化输出的定义过程的审查形式。

[来源：ISO/IEC 20246:2017, 3.5]

3.1869

形式规格说明 formal specification

以正式符号编写的规范，通常用于证明正确性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1886]

3.1870

正式测试 formal testing

根据客户、用户或指定管理层审查和批准的测试计划和测试规程实施的测试。

参见：非正式测试。

[来源：ISO/IEC/IEEE FDIS 24765 3.1887]

3.1871

形式验证 formal verification

使用形式化数学方法证明或反驳预期应用程序相对于形式化规范或属性的正确性的活动。

[来源：ISO/IEC 23643:2020, 3.10]

3.1872

形式化 formalization

根据形式语言（如一阶逻辑）对语言的语义进行精确描述。

[来源：ISO/IEC/IEEE FDIS 24765 3.1889]

3.1873

形成性结构 formative construct

在结构与其度量之间的关系中，由其观测度量形成的结构。

注：结构是其措施的结果，每项措施都是结构的决定因素。

[来源：ISO/IEC 33003:2015, 3.7]

3.1874

形成性评价 formative evaluation

为改善评价客体，尤其是仍在开发阶段的评价客体而设计和使用的评估。

[来源：GB/T 25000.22-2019]

3.1875

向前传递 forward pass

一种关键路径法技术，通过从项目开始日期或给定的时间点开始，按照时间顺序逐步推进的方式，来计算项目的最早开始日期和最早完成日期。

参见：进度网络分析，后向法。

[来源：《项目管理知识体系指南（PMBOK 指南）第五版》]

3.1876

前向（正向）恢复 forward recovery

由用以文件的按时间顺序的变更记录的数据，更新较早的版本以重构文件至给定的状态。

一种恢复类型，在此种类型中，系统、程序、数据库或其他系统资源被恢复为新的、不是以前所有的、能执行所要求的功能的状态。

参见：后向恢复

[来源：GB/T 11457:2006, 2.657]

3.1877

喷泉模型 fountain model

一种软件开发模型，该模型体现了软件创建所固有的迭代和无“间隙”的特征。它主要用于描述面向对象开发过程。

3.1878

四地址指令 four-address instruction

包含四个地址字段的计算机指令。

示例：一条将位置 A、B 和 C 中的内容相加，并将结果存入位置 D 的指令。

参见：单地址指令、双地址指令、三地址指令、零地址指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1894]

3.1879

四加一地址指令 four-plus-one address instruction

包含五个地址段的计算机指令，其中第五个地址段存放下一条待执行指令的地址。

示例：一条指令将位置 A、B 和 C 中的内容相加，将结果存入位置 D，然后执行位于位置 E 的指令。

参见：一加一地址指令、二加一地址指令和三加一地址指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1895]

3.1880

第四代语言 fourth generation language (4GL)

旨在提高高阶（第三代）语言实现的生产力的计算机语言，并且通常使非程序员也可以使用计算能力。

注：通常包含集成数据库管理系统、查询语言、报表生成器、屏幕定义功能、图形生成器、决策支持功能、财务建模、电子表格功能以及统计分析功能等特性。

参见：机器语言、汇编语言、高级语言、第五代语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1896]

3.1881

F 型规范 F-profile

格式与呈现规范。

[来源：ISO/IEC 10746-1:1998]

3.1882

结构 frame

将浏览器窗口分割成多个独立窗口以显示不同内容或同一内容（文档）不同部分的元素。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.9]

3.1883

提出疑虑 / 陈述问题 frame a concern

构建、组合并表达关注点。

注：该术语用于区分通过视角构建关注点与在结果视图中解决这些关注点的不同阶段。这类似于“构建问题”与“解决问题”之间的区别。

[来源：GB/T 45630-2025, 3.8]

3.1884

框架 framework

可复用设计（模型或代码），可对其进行精化（特化）与扩展，用以提供众多应用整体功能中的某一部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1908]

3.1885

自由浮动时间 free float

任务的预定完成日期与后续任务的预定开始日期之间存在的可用间隙。

参见：总浮动时间。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1909]

3.1886

避免经济风险 freedom from economic risk

在预期使用环境中，产品或系统降低对财务状况、高效运营、商业资产、声誉或其他方面潜在风险的程度。

[来源：ISO/IEC 25019:2023, 3.2.2]

3.1887

环境与社会风险免除 freedom from environmental and societal risk

在预期使用环境中，产品或系统降低对环境和整个社会潜在风险的程度。

[来源：ISO/IEC 25019:2023, 3.2.2.2]

3.1888

避免健康风险 freedom from health risk

产品或系统在预定使用周境中，降低对人员健康潜在风险的程度。

[来源：ISO/IEC 25019:2023, 3.2.2.3]

3.1889

避免人员生命安全风险 freedom from human life risk

产品或系统在预定使用周境中，降低对人员生命潜在风险的程度。

[来源：ISO/IEC 25019:2023, 3.2.2.4]

3.1890

抗风险 freedom from risk

产品或系统的质量在经济现状、人的生命、健康或环境方面缓解或规避潜在风险的程度。

注1：风险是一个给定威胁发生的概率及该威胁发生后的潜在不利后果的函数。

注2：GB/T 25000 标准各部分考虑的风险是由产品质量缺陷引起的风险。

注3：抗风险包括不利于用户、组织或项目的潜在风险的减免程度。

[来源：GB/T 25000.22-2019, 4.9]

3.1891

冻结分支 frozen branch

不进行开发活动的分支，该分支用于准备发布，或其上的活跃开发已停止。

[来源：ISO/IEC/IEEE FDIS 24765 3.1916]

3.1892

全双工 full duplex

能够同时在两个方向上进行数据传输。

参见：半双工模式。

[来源：ISO/IEC/IEEE FDIS 24765 3.1930]

3.1893

功能 function

⟨⟩系统或部件的定义的目标或特征动作。

参见：功能需求、功能规格说明和功能测试。

⟨⟩执行特定动作的软件模块，由出现在表达式中的名字引用，可以接收输入值并返回一单值。

参见：子例程。

⟨⟩为实现预定的目的或目标，专门安排的或适宜于个人或工具所充当的角色来完成的一组相关动作。

示例：一系统有库存控制作为它的主要功能。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1894

功能字段 function field

计算机指令的字段，指定要执行的操作。

参见：地址字段。

[来源：ISO/IEC/IEEE FDIS 24765 3.3117]

3.1895

功能点分析 (FPA) function point analysis; FPA

功能规模测量的一种执行方式，用于测量从用户角度观察的、与商业应用相关的软件开发、增强和维护活动的功能规模。

[来源：GB/T 42566-2023, X.X]

3.1896

功能点数 function point count

应用本文件规则测量应用程序或项目功能规模的活动。

注：功能点计数有三种类型：应用程序、开发项目和增强项目。

[来源: GB/T 42449-2023, X. X]

3.1897

功能点表 **function point table**

用于给功能类型分配功能点的数据表，依赖于功能类型和功能复杂度。

[来源: GB/T 42588-2023, X. X]

3.1898

功能点 **function point; FP**

测量软件功能规模的单位。

[来源: GB/T 36964-2018; ISO/IEC TR 7052:2023, 3.25; ISO/IEC/IEEE 32430:2025, 3.1.18]

3.1899

功能类型 **function type**

从 FPA 的角度所看到的组成应用程序的五种类型的组件。这些组件确定了应用程序提供给用户的功能数量。组件类型如下：外部输入，外部输出，外部查询，内部逻辑文件和外部接口文件。

[来源: GB/T 42588-2023]

3.1900

功能适应性 **functional adaptability**

AI 系统能够从数据或先前操作结果中准确获取信息并将该信息用于未来预测的程度。

[来源: ISO/IEC 25059:2023, 3.2.2]

3.1901

功能分析 **functional analysis**

对实际或计划系统的功能进行系统调研。

3.1902 [[来源: ISO/IEC 2382:2015, X. X]

功能适合性 **functional appropriateness**

产品提供有助于完成指定任务和目标的的功能的能力。

示例：产品提供完成任务所必需且充分的步骤，排除任何不必要的步骤。

注：功能适合性对应于任务的适合性。

[来源: ISO/IEC 25010:2023, 3.1.3]

3.1903

功能基线 **functional baseline**

系统性能（功能、互操作性和接口特性）的描述，以及为证明达到这些特定特性所需的验证。参见：分配基线、开发配置、产品基线。

[来源: ISO/IEC/IEEE 24748-7:2019, 3.1]

3.1904

功能配置审计 **Functional configuration audit; FCA**

为验证某一配置项的研制已圆满完成而实施的审计，该审计确认该配置项已达到功能或分配技术状态标识中规定的性能和功能特性，而且其运行与保障文件完整并令人满意。

参见：技术状态管理、物理技术状态审核。

[来源: IEEE 828-2012, 2.1]

3.1905

功能配置标识 **functional configuration identification**

在配置管理中，配置项的当前批准技术文档。

参见：分配配置标识（allocated configuration identification）、产品配置标识功能基线

(product configuration identification functional baseline)。

注：它规定了所有必要的功能特性、证明达到规定功能特性所需的测试、与相关配置项的必要接口特性、配置项的关键功能特性及其关键下级配置项（如有）以及设计约束。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1947]

3.1906

功能设计 functional design

系统各组件的功能说明以及它们之间的工作关系。参见架构设计（architectural design）。

[来源：ISO/IEC 2382:2015, X.X]

3.1907

功能域分类 functional domain categorization,; FDC; FDC

功能域识别的过程。。

[来源：ISO/IEC TR 14143-5:2004, 3.2]

3.1908

功能危险性分析 Functional hazard analysis; FHA

无（仅是缩略语）。

3.1909

功能过程 functional process

功能过程类型 functional process type。

一组用户功能需求的基础组件，包括一组唯一的、紧密相关的、可独立执行的数据移动。

注 1：由功能用户的数据移动（输入）触发，通知软件段的功能用户已确认此触发事件。当软件执行完所有为响应触发事件所需的工作后，该步骤结束。

注 2：除了通知软件段所发生的事件，由事件触发的输入可包括事件相关的利益对象的数据。

[来源：GB/T 42452-2023]

3.1910

功能性需求 functional requirement

规定了系统或系统组件所执行功能的要求。

[来源：GB/T 25000.40-2018]

3.1911

功能规模 functional size; FS

通过功能用户需求进行量化导出的软件规模。

[来源：GB/T 18491.1-2001, 3.6]

3.1912

功能规模测量方法 functional size measurement method; FSMM

一种符合 ISO/IEC 14143-1:2007 强制要求的、由一系列规则所定义的 FSM 的特定实现。

[来源：ISO/IEC 14143-1:2007, 3.4]

3.1913

功能规模测量 functional size measurement,; FSM; FSM

功能规模的量化过程。

[来源：GB/T 42452-2023; ISO/IEC 14143-1:2007, 3.7; ISO/IEC 19761:2011, 2.11]

3.1914

功能规格 functional specification

该文档从用户的角度阐述了系统或组件需执行的必要操作。

注：通常为需求规格说明的一部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1961]

3.1915

功能性系统设计 functional system design

软件系统组件的功能及其之间的工作关系规范。

[来源：ISO/IEC 16350-2015, 4.19]

3.1916

功能用户 functional user

软件用户功能需求中作为数据发送者和/或预期的数据接收者的用户。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1917

用户功能需求/功能性用户需求 functional user requirements; FUR

描述软件在执行任务和提供服务时所做工作的用户需求子集。

注1：用户功能需求包括但不限于：

- 数据迁移（例如：输入客户数据、发送控制信号）；
- 数据转换（例如：计算银行利息、计算平均温度）；
- 数据存储（例如：保存客户订单、随时记录环境温度）；
- 数据检索（例如：列出当前雇员、检索飞机位置）。

注2：用户非功能需求包括但不限于：

- 质量约束（例如：易用性、可靠性、效率和可移植性）；
- 组织约束（例如：操作位置、目标硬件、标准依从性）；
- 环境约束（例如：互操作性、信息安全性、隐私和安全）；
- 实现约束（例如：开发语言、交付计划）。

[来源：GB/T 42452-2023；ISO/IEC 14143-1:2007, 3.8；ISO/IEC 19761:2011, 2.14]

3.1918

<精益>现场 gemba

对实际使用中的工作过程的观察。

[来源：ISO/IEC 33202:2024, 3.15]

3.1919

通用人工智能 general artificial intelligence; general AI

在全部认知能力范围内展现出与人类相当的智能行为的人工智能。

[来源：ISO/IEC TR 29119-11:2020, 3.1.39]

3.1920

通用输入/输出端口 (GPIO) general purpose input/output port (GPIO); GPIO

微型计算机上的通用引脚（端口），其功能（无论是输入引脚还是输出引脚）未预先定义，由用户控制。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1921

通用输入 / 输出端口 general purpose input/output port,; GPIO; GPIO

微型计算机上的通用引脚（端口），其功能（是输入引脚还是输出引脚）未预先定义，由用户控制。

[来源：ISO/IEC/IEEE 24765d:2015]

3.1922

通用寄存器 **general register**

存储地址和数据的寄存器。

[来源: ISO/IE0/IEEE 24765 FDIS, 2025]

3.1923

通用系统特性 **general system characteristics; GSCs**

技术复杂度调整因子的术语。

[来源: ISO/IEC 20968:2002, 10]

3.1924

通用性 **generality**

系统或组件执行广泛功能的能力。

参见: 可重用性。

[来源: ISO/IE0/IEEE FDIS 24765, X. X]

3.1925

泛化 **generalization**

分类法中两个实体的实例代表同一个真实或抽象的事物。

参见: 分类。

注: 一个实体(通用实体)代表事物的完整集合, 另一个(类别实体)代表这些事物的子类型或子分类。类别实体可以具有一个或多个属性, 或者与另一个实体的实例存在关系, 而这些属性或关系并非所有通用实体实例所共有。类别实体的每个实例同时也是通用实体的一个实例。

[来源: ISO/IE0/IEEE 24765 FDIS, 2025]

3.1926

泛化结构 **generalization structure**

超类与其更具体、更直接的子类之间的关联。

[来源: ISO/IE0/IEEE 24765 FDIS, 2025]

3.1927

泛化分类法 **generalization taxonomy**

一组具有共同通用祖先的泛化结构。

注: 在泛化分类法中, 每个实例都由分类法中的一个或多个类别进行完整描述。将类别构建为泛化分类法决定了类别之间的职责继承。

[来源: ISO/IE0/IEEE 24765 FDIS, 2025]

3.1928

通用输入/输出端口 **general-purpose input/output port; GPIO**

微型计算机上的通用引脚(端口), 其功能(输入或输出引脚)未预先定义, 由用户控制。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1979]

3.1929

生成地址 **generated address**

在计算机程序执行过程中计算得出的地址。参见: 绝对地址、有效地址、相对地址、间接地址。

[来源: ISO/IE0/IEEE 24765 FDIS, 2025]

3.1930

生成 **generation**

从特定元模型出发, 定义和描述一种方法论的行为。

注：生成方法论包括使用所选元模型解释每个方法论要素的结构位置和语义。因此，可能存在哪些方法论要素以及它们之间如何相互关联，都受到此类元模型的约束。通常，方法工程师会执行生成过程，从而生成一个完整且可用的方法论。

[来源：ISO/IEC 24744:2014, 3.8]

3.1931

通用实体 generic entity

其实例被归类为一个或多个子类型或子分类的实体（类别实体）。参见：超类、父类。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1932

通用实践 generic practice

一项活动，当持续执行时，有助于实现特定的过程属性。

[来源：ISO/IEC 33001:2015, 3.3.6]

3.1933

通用配置文件组 generic profile group

适用于不开发关键系统或软件产品的极小型实体（VSEs）的概况组。

[来源：ISO/IEC 29110-1-2:2024, 3.43]

3.1934

通用程序单元 generic program unit

以通用方式定义的软件模块，需要替换特定的数据、指令或两者，以便在计算机程序中使用。参见：实例化。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1935

白盒 glass box

内部内容或实现方式已知的系统或组件。

参见：“黑盒”。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1936

白盒测试 glass-box testing

动态测试，其中测试用例源自对测试项结构的分析。参见：结构测试示例：分支测试、决策测试、语句测试。

注1：基于结构的测试不仅限于组件级别，还可以用于所有级别。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.80]

3.1937

全局属性 global attribute

当描述外键的属性与描述关联候选键的属性（及其值）相同时，此条件成立。

[来源：ISO/IEC 15476-4:2005, 6.10]

3.1938

全局压缩 global compaction

在微程序设计中，压缩是指将微操作移出其在单入口单出口顺序块的边界。参见：局部压缩。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1939

全局数据 global data

计算机程序中两个或多个非嵌套模块可以访问的数据，无需在模块间显式传递参数。参见：本地数据。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1940

全局标签 `global label`

与网图本身关联的标签，而不是与网图的对象关联的标签。

[来源：ISO/IEC 15909-2:2011, 4.1.4]

3.1941

全局唯一标识符 `globally unique identifier; GUID`

16 字节字符串，其生成方式可确保该字符串在任何场景下都具有极高的唯一性概率。

注：全大写形式特指 16 字节版本，小写形式则指可使用 URI 或基于 16 字节标识符的通用算法。

[来源：ISO/IEC 19770-5:2015, 3.16]

3.1942

目标 `goal`

预期结果。

[来源：ISO TR 25060:2023, 3.1.6]

[来源：GB/T 25000.22-2019 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第 22 部分：使用质量测量]

3.1943

政府现货 `Government-off-the-Shelf`

政府提供的、可在其他项目中重复使用的软件。

参见：商业现货 COTS。

[来源：IEEE 1012-2024, 3]

3.1944

等级 `grade`

用于区分功能用途相同但质量要求不同的项目的类别或等级。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.1945

粒度 `granularity`

收集数据的深度或详细程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2021]

3.1946

图形信息 `graphical information`

定义对象图形外观的信息和网图的标签，可以是位置、大小、线条颜色、填充颜色、字体或线宽。

[来源：ISO/IEC 15909-2:2011, 4.1.3]

3.1947

图形处理器 `graphical processing unit; GPU`

专用于图像绘制等显示功能的专用集成电路 (ASIC)。

注：图形处理器 (GPU) 专为图像的单功能并行数据处理而设计，这种并行处理方式对于执行神经网络等人工智能软件非常有用。

[来源：ISO/IEC TR 29119-11:2020, 3.1.40]

3. 1948

图示符号 graphical symbol

具有特定含义的视觉可感知图形，用于独立于语言传递信息。

[来源：IEC/IEEE 82079-1:2019，3.13]

3. 1949

图形符号 GraphicSymbol

对表示特定软件行为概念实例的特定图像或绘图的引用。

[来源：IEEE 1175.3-2004，3.17]

3. 1950

绿地系统工程 greenfield system engineering

面向新环境及一套用户场景与需求的系统开发。

注：绿地方法较为罕见，因为该方法通常在系统边界内、或来自外部接口及使能系统方面，不存在显著的遗留约束或依赖关系。

[来源：INCONE 系统工程手册，第五版.]

3. 1951

基本原则 ground rules

项目团队为改善工作关系、效率和沟通而采取的可接受和不可接受的行为清单。

[来源：ISO/IEC/IEEE FDIS 24765，3.2028]

3. 1952

基准真值 ground truth

被选定用于反映事物客观事实的数据，例如用于科学分析或机器学习模型训练的数据。

注：在对人类情绪、情感或共情进行度量时，真值（ground truth）难以获取且具有主观性。

[来源：IEEE 7014-2024，3.1]

3. 1953

组 group; groups

2. 模型元素的数量被视为由单个不同元素的可追溯性关系形成的单元。

[来源：ISO/IEC 19506:2012，4；

[来源：ISO/IEC/IEEE 247651:2024]]

3. 负责一组任务或活动的部门、经理和个人的集合。组的规模不等，可以是一个指派的兼职人员，可以由不同部门指派的若干兼职人员，也可以是若干个专职人员。

[来源：GB/T 11457-2006，X.X]

3. 1954

指南 guideline

官方建议或意见，表明应如何完成某事的政策、标准或程序。

[来源：ISO/IEC/IEEE FDIS 24765，3.2037]

3. 1955

半双工 half duplex

能够双向传递数据，但一次只能传递一个方向。

参见：全双工。

[来源：ISO/IEC/IEEE 24765d:2015]

3.1956

硬拷贝 hard copy

在诸如打印机或绘图仪等输出装置上生成的显示图像的永久副本，可被带走。

[来源：ISO/IEC 2382: 2015]

3.1957

硬故障 hard failure

导致系统完全关闭的故障。

参见：软故障

3.1958

硬件资产管理 hardware asset management, ; HAM; HAM

组织为实现硬件资产价值而开展的协调活动。

注：属于 IT 资产管理的专业化领域。

[来源：ISO/IEC DIS 19770-6, 3.1.2]

3.1959

硬件配置项 hardware configuration item; HCI

用于配置管理的硬件的集合，在配置管理过程中被视为一个整体。另见软件配置项。

注：当功能分配已明确区分出设备功能与软件功能，并且硬件已被确认为一个配置项时，即存在一个 HCI（硬件配置项）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2047]

3.1960

硬件配置项(HWCI) hardware configuration item(HWCI); HCI

硬件集合，用于配置管理，并在配置管理过程中被视为一个单一实体。另请参阅软件配置项。

注：当功能分配已明确区分设备功能和软件功能，且硬件已被确立为配置项时，即存在超融合基础设施（HCI）。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1961

硬件配置项 hardware configuration item, ; HCI

为配置管理指定的且在配置管理过程中作为单个实体对待的硬件的集合。与计算机软件配置项相对。

注：当已完成功能分配，明确将设备功能与软件功能区分开，且该硬件已被确立为配置项时，即存在硬件配置项。

参见：配置项

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.1962

硬件工程 hardware engineering

通过转换一组代表利益相关者需求、期望和约束集合的需求，采用系统的、有纪律的和可量化的方法来设计、实现和维护有形产品；使用文档化的技术，如软件工程、系统工程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2049]

3.1963

硬件标识符生成器 hardware identification creator, ; HWID creator

首次创建硬件标识符记录的实体。

注：该实体可以是与该记录相关的组件制造组织的一部分，此时硬件标识符创建者与组件制造商为同一主体。硬件标识符创建者也可以是独立于制造商的组织或第三方（例如由设备所有者部署的操作系统或工具为现有硬件组件创建硬件标识符记录的情形）。

[来源: ISO/IEC DIS 19770-6, 3.1.3]

3.1964

硬件标识方案 hardware identification schema; HWID schema

一种包含硬件组件的数字描述及其相关信息的信息结构。

[来源: ISO/IEC DIS 19770-6, 3.1.4]

3.1965

硬件 hardware; HW

用于处理、存储或传输计算机程序或数据的物理设备。

参见: 软件。

[来源: ISO/IEC/IEEE 12207:2026, 3.1.29;]

3.1966

损害 harm

对人的健康造成伤害或损害, 或对财产或环境造成损害。

[来源: IEC/IEEE 82079-1:2019, 3.1; IEEE 7009-2024, Fail-Safe Design of Autonomous and Semi-Autonomous Systems, 3.1]

负面事件或负面社会发展对人们造成价值损害或损失。

注: 危害需要对自己或他人产生负面价值影响, 在各自的组织内或之外。

[来源: IEEE 7000:2021, 3.1]

3.1967

使用所致损害 harm from use

由系统使用所导致的涉及健康、安全、财务或环境的负面后果。

注: 该负面后果可能影响用户或其他任何利益相关方。

[来源: ISO/TR 25060:2023, 3.1.15]

3.1968

哈佛架构 Harvard architecture

指令和数据在物理上分开的通信路径的计算机体系结构。

[来源: ISO/IEC/IEEE 24765d:2015]

3.1969

危险 hazard

可能造成危害或损害的内在属性或条件。

[来源: IEEE 1012-2024, 3.1.11]

3.1970

危险识别 hazard identification

识别危害存在并定义其特征的过程。

[来源: IEEE 1012-2012, 3.1.12]

3.1971

危害警示 hazard warning

产品具备的一种能力, 能够就对操作或内部控制而言不可接受的风险提供警示, 使用户能够有足够时间作出反应, 从而维持安全的运行状态。

示例: 交通信号灯会向行人发出信号, 显示从绿灯切换为黄灯或红灯前所剩余的秒数。

[来源: ISO/IEC 25010:2023, 3.9.4]

3.1972

危害事件 hazardous event

可能导致损害的事件。

[来源: IEEE 7009-2024, 3.1]

3.1973

危害情形 hazardous situation

人员、财产或环境暴露于一个或多个危害之中的情形。

[来源: IEEE 7009-2024, 3.1]

3.1974

标题, 报头, 前导文件 (头文件) header

放在计算机程序或例程开始处的注释块。

放在文件或消息开始处的标识或控制信息。

参见: 尾部

[来源: GB/T 11457 2.695]

3.1975

重量级进程 heavyweight process

有独立内存和多重控制线程的进程。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.2070]

3.1976

帮助系统 help system

程序的辅助部分, 有时是单独的程序, 允许用户在请求时查看部分在线文档或帮助文本 (参见在线文档系统)。

3.1977

启发式评估 heuristic evaluation

由一名或多名专家判断是否符合一套公认的原则进行评估。

[来源: ISO/IEC/IEEE 26513:2017, 3.18]

3.1978

分层分解 hierarchical decomposition

一种模块化分解方法, 通过一系列自顶向下的细化步骤, 将系统分解成组件层次结构。参见: 功能分解、逐步细化。

[来源: ISO/IEC/IEEE 24765 FDIS, 2025]

3.1979

层次建模 hierarchical modeling

用于评价计算机性能的技术, 在此技术中, 计算机系统用它的子系统的层次结构表示, 子系统被分析以确定它们的性能, 且其结果用于评价整个系统的性能。

[来源: ISO/IEC/IEEE 24765:2017; GB/T 11457-2006, 2.698]

3.1980

分层连续 hierarchically consecutive

树中两个指定节点之间所有节点的不间断单向遍历。

[来源: ISO/IEC/IEEE 24765m:2024, 2.1.57]

3.1981

分层[结构] hierarchy

一种结构，其中，部件排列为它们的下属构成的级、每个部件有零个、一个或多个下属，直至无部件有多于一个的下属部件。

参见： 层次结构分解和层次模型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2078]]

3.1982

层次结构图 hierarchy chart

从高层信息开始，逐步分解为较低层级细节的图表。

[来源：ISO/IEC/IEEE 24765 FDIS, 2025]

3.1983

高级模型 high level model

一种虚拟样机系统的顶层模型。

注：它对整个虚拟样机系统的组成和行为进行顶层宏观描述，包括系统的结构模型和行为模型。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1984

高级建模方法 high level modeling method

一种基于不同领域异构模型搭建、描述复杂系统的建模理论与方法。

注：它以系统论为基础，通过组件化和层次化体现系统论的思想。

[来源：GB/T 36457-2018, X.X]

3.1985

高层功能点分析 high level; function point analysis

在应用程序生存周期的早期阶段进行一项可能的功能点分析，以确定应用程序或项目的规模（假设具有某些最低规格）。

[来源：ISO/IEC 24570:2018, X.X]

3.1986

高层管理 higher-level management

为过程提供政策和总体指导但不提供过程的直接日常监督和控制的人员。

注：此类人员属于以上组织中的管理层负责流程的直接级别。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2084]

3.1987

高层设计 high-level design

定义指导低级设计与实现的高级概念的过程。

参见：架构。

注：高级设计通常涉及将系统划分为子程序，并明确它们之间的接口。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2081]

3.1988

高层级关键字 high-level keyword

由其他关键字组成，涵盖复杂活动的关键字。由领域专家使用高层级关键字编写关键字测试用例。

[来源：ISO/IEC/IEEE 26513:2017, 4.2]

3.1989

高层网 high-level net

一种代数结构，包括一组位置；一组转换；一组类型；一个将每个类型与每个位置关联起来的函数，以及为每个转换关联一组模式（一种类型）；一个预函数，为每个转换模式对位置施加令牌要求（令牌

的多重集合)；一个后函数，为每个转换模式确定位置的输出令牌(令牌的多重集合)；以及一个初始标记。

参见：高级 Petri 网，HLPN。

3.1990

高级 Petri 网图 high-level Petri Net graph

一张网图及其相关标注，包括位置类型、弧标注和过渡条件和它们在一组声明中的相应定义，以及网的初始标记。

[来源：ISO/IEC 15909-1:2004, 2.1.12]

3.1991

高级语言 high-order language; HOL

高级语言是一种编程语言，它对运行程序的计算机了解甚少，可以翻译成多种不同的机器语言，允许对操作和地址进行符号命名，提供旨在简化数据结构和程序逻辑表达的功能，并且通常每个程序语句都会生成多条机器指令。参见：汇编语言、第五代语言、第四代语言、机器语言。

示例：Ada、COBOL、FORTRAN、ALGOL、PASCAL。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.1992

高阶语言(HOL) high-order language (HOL)

需要对运行程序的计算机了解很少的编程语言，可以翻译成多种不同的机器语言，允许对操作和地址进行符号命名，提供旨在促进数据结构和程序逻辑表达的功能，以及通常会为每个程序语句生成多个机器指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2083]

3.1993

历史信息 historical information

以往项目的文件和数据，包括项目文件、记录、通信、已结束的合同、已结束的项目。

[来源：项目管理知识体系指南(PMBOK®指南) 第五版]

3.1994

历史文件 history file

一种概念性数据类型，它由一组事件记录构成，可对这些记录进行处理以获得在感兴趣时间范围内对支持行为、事件和模式的持续时间、计数和统计数据。

[来源：IEEE 982:2024, 3.1]

3.1995

同构冗余 homogeneous redundancy

在容错方面，用相同的手段实现相同的功能比照。

示例：使用两个相同的处理器。

参见：多样性

[来源：ISO/IEC/IEEE 24765:2017, 3.2092]

3.1996

宿主计算机 host computer

用于开发软件的计算机。

参见：目标计算机

3.1997

[宿]主机 host machine

程序或文件所装入的计算机。

用以开发供另一台计算机用的软件的计算机。与 2.1677 条 目标机 (target machine) (a) 相对。

用以模仿另一台计算机的计算机。与 2.1677 条 目标机 (target machine) (b) 相对。

在计算机网络中, 为该网络的用户提供处理能力的计算机。

[来源: GB/T 11457-2006 2.710]

3.1998

恶意回退 hostile backout

由非引入原始更改的提交者在未事先协商的情况下执行的回退操作。

注: 这通常是提交冲突的开始。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.1999

人因 human factors

对人类能力、特征、行为、动机及表现相关信息的系统化应用。

注: 涵盖人体工程学、人体测量学、工效学、工作绩效技能与辅助设备以及人类绩效评估等领域的原则与应用。

[来源: INCONE 系统工程手册, 第五版.]

3.2000

人力资源规划 human resource planning

识别和记录项目角色、职责和报告关系, 以及按时间段估计所需人员, 并制定人员配置管理计划。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2106]

3.2001

人因系统工程 human systems engineering

贯穿系统生存周期的各项活动, 旨在解决系统设计中的人为因素 (包括可用性、效能指标、性能指标以及总拥有成本)。

注: 这些活动包括人力、人员、培训、人因工程、健康危害及安全问题的定义与综合。

[来源: ISO/IEC/IEEE 24765f:2016]

3.2002

人因系统集成 Human systems integration; HSI

跨学科技术与管理的整合过程, 旨在将人的考量融入并贯穿所有系统要素。

[来源: ISO/IEC/IEEE 29148:2018, 3.1.13]

3.2003

以人为本的设计 human-centered design

一种旨在通过关注系统的使用情况来提高交互系统可用性的系统设计与开发方法; 应用人因学、人体工程学和可用性方面的知识和技术。

[来源: ISO/IEC 25063:2014, 3.6]

3.2004

以人为本的设计 human-centered design

以用户为中心的设计 user-centered design

系统设计和开发的方法, 旨在通过关注系统的使用使交互系统更可用; 应用人为因素、人体工程学和可用性知识和技术。

[来源: ISO/IEC 25063:2014, 3.6]

3.2005

以人为本的设计 human-centred design

一种系统设计与开发的方法，通过关注系统的使用情况，并运用人因工程、人体工效学及可用性知识和技术，旨在提高交互式系统的可用性。

注：条目中使用“以人为中心的设计”一词而非“以用户为中心的设计”，以强调设计影响众多相关方，而不仅仅是通常所指的用户；但在实践中，这两个术语常被互用。可用系统可带来多种益处，包括提高生产效率、增强用户福祉、避免压力、提升可访问性及降低伤害风险。

同义词：以人为中心的设计（human-centered design）、以用户为中心的设计（user-centered design）、以用户为中心的设计（user-centred design）。

[来源：ISO/IEC 25063:2014, 3.6]

3. 2006

以人为中心的质量 human-centred quality

与易用性、可达性、用户体验以及避免因使用而造成伤害相关的要求得到满足的程度。

注 1：用户与系统交互的预期结果的统称。

同义词：以人为中心的质量（human-centered quality）、以人为中心的质量（human centered quality）。

[来源：ISO TR 25060:2023, 3.1.13]

3. 2007

混合方法 hybrid approach

由两个或更多敏捷和非敏捷要素组合而成，并具有非敏捷最终结果的方式。

[来源：项目管理知识体系指南（PMBOK(R) Guide）— 第七版]

3. 2008

混合耦合 hybrid coupling

一类耦合，在其中，数据项值的范围的不同子集，假定能用于不同的软件模块中不同的和不相关的目的。

[来源：ISO/IEC FDIS 24765, 3.2118]

3. 2009

超链接 hyperlink

从一个文档中某一部分到另一文档（或同一文档中其他部分）的引用。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.15]

3. 2010

超文本标记语言 hypertext markup language; HTML

一种用于创建网页的语言。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.17]

3. 2011

超文本传输协议 HyperText Transfer Protocol; HTTP

一种用于分布式、协作式和超媒体信息系统的应用层协议。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.18]

3. 2012

I/O 任务结构化标准 I/O task-structuring criteria

任务结构化标准的类别，用于解决如何将设备接口对象映射到 I/O 任务以及何时激活 I/O 任务。

3. 2013

ICOM 代码, ICOM code,

箭头参考 arrow refrence

图中的表达式，用于无歧义地标识出另一图中的箭头。

注：用 ICOM 代码关联子图的边界箭头与附着到亲代框的箭头。

[来源：ISO/IEC/IEEE 24765m:2024, 2.1.58]

3.2014

ICOM 标签 ICOM label

不带波浪线的箭头标签，直接附着在输出边界箭头的箭头上，或在输入、控制或机制边界箭头的箭尾上。

注：ICOM 标签把子图的边界箭头关联到亲代框附着箭头的箭头标签。

[来源：ISO/IEC/IEEE 24765m:2024, 2.1.59]

3.2015

图标 icon

屏幕上显示的代表计算机系统功能的图形

[来源：ISO/IEC 26514:2008, 4.22]。

3.2016

ICT 产品, ICT product,

信息通信技术产品 information and communication technology product

采用信息通信技术（ICT）并能使其成为信息系统组成部分的产品。

注：[来源：GB/T 25000.30-2021]图 3 描述了 ICT 产品的组成及其与信息系统的关系。

[来源：GB/T 25000.30-2021, 3.8; ISO/IEC 25030:2019, 3.8]

3.2017

ICT 需求 ICT requirement, information and communication technology requirement

在设计过程中通过采纳某种信息通信技术（ICT）的技术解决方案而得到的需求。

注：ICT 技术性解决方案包括基于 Web 的技术、云服务器等。

[来源：GB/T 25000.30-2021, 3.7; ISO/IEC 25030:2019, 3.7]

3.2018

思维导图, idea/mind mapping,

创意导图 idea mapping

思维导图, 创意导图。

用于把由独立头脑风暴产生的创意整合为一张图的技术，以反映理解上的共性和差异并产生新创意。

[来源：ISO/IEC/IEEE 24765h:2019——PMBOK v5]

围绕一个中心理念、图形化地连接创意和概念以展示其关系的技术。

参见：实体-关系（E-R）图。

注：用于产生新想法并就概念间关系达成共识。

[来源：项目管理知识体系指南（PMBOK(R) Guide）— 第七版]

3.2019

想法 idea/

思维导图 mind mapping

想法导图 idea mapping

思维导图 mind mapping

用于将通过个人头脑风暴会议产生的想法整合到一张地图中以反映理解的共性和差异并产生新想法的技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2148]

3. 2020

理想工时 `ideal time`

对开发人员或团队完成任务或交付特性所需时间的最佳情况估计。

[来源: Software Extension to the PMBOK® 指南 Fifth Edition]

3. 2021

识别事务 `identification transaction`

由一次或多次捕获尝试与生物特征搜索构成的序列，用于查找并返回归属于单个个体的生物特征参考标识符。

[来源: ISO/IEC TR 29119-13:2022, 3.1.30]

3. 2022

标识关系 `identifying relationship`

非多对多的具体关系，其中父实体主键中的每个属性都包含在子实体主键中。

参见: 非标识关系[键类型]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.79]

3. 2023

标识 `identity`

实例固有的、使其区别于所有其他实例的性质。

注: 标识是实例固有的，并且与实例的性质值或实例所属的类无关。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.80]

3. 2024

身份风格视图 `identity-style view`

使用身份样式建模构造生成的视图。

[来源: IEEE 1320.2-1998, 3.1.81]。

3. 2025

“若-则-否则”语句 `if-then-else`

一种单入口单出口，定义了条件的两路分支。若规定的条件满足则要执行的处理、若不满足，如何处理是任选的、在两种情况下都返回控制至立即跟随整个结构的语句。与 2.186 条 (分)情况语句(case)、2.830 条 跳移(jump)和 2.683 条 “转至”语句(go to)相对。参见: 2.510 条 二元选择构造(dyadic selective construct)和 2.980 条 一元选择构造(monadic selective construct)。

图 13 if-then-else 构造。

[来源: GB/T 11457 2.718]

3. 2026

插图 `illustration`

与正文分开的图形元素，通常在正文中引用。

[来源: ISO/IEC 26514:2008, 4.23]

3. 2027

豁免条款 `immunity provision`

由法律规定或政府机关授予的，免除法律责任、处罚或起诉的豁免权。

[来源: IEEE 7005:2021, 3.1]

3. 2028

不可变类 `immutable class`

实例集合是固定的类。

参见：可变类，值类。

注：不可变类的实例不会随时间的推移而产生或消失。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.83]

3. 2029

影响分析 impact analysis

识别变更请求影响的所有系统和软件产品，并估算完成变更所需的资源。

注：这包括确定变更范围以计划和实施工作，准确估算完成这项工作需要的资源，并分析所请求的变更的成本和收益。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2176]

3. 2030

成效导图 impact mapping

一种战略规划方法，在产品开发期间为组织提供可视化路线图。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3. 2031

减损 impairment

非终止性系统失效，或运行时环境资源或运行条件的变化，导致系统能力退化、但不造成即时终止性失效。

注：减损可以由系统故障、按设计的系统响应或环境条件引起。它可能触发其他减损或失效。每个减损都具有持续时间，并可能导致损失。

[来源：IEEE 982:2024, 3.1]

3. 2032

障碍 impediment

一种妨碍团队实现其目标的障碍。

同义词：阻碍 (blocker)。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3. 2033

不完全排错 imperfect debugging

在可靠性模拟中，纠正或清除已经发现故障的意图并非总是成功的一种假定。

3. 2034

可执行标准 implementable standard

技术对象模板。

[来源：ISO/IEC 10746-3:2009, 9.1.1]

3. 2035

实现 implementation

把设计转换为硬件部件、软件部件或两者的过程。

参见：编码。

[来源：ISO/IEC/IEEE 90003:2018 3.4; GB/T 11457-2006, 2.726, 有修改]

3. 2036

实现类 implementation class

一种如下衍型类：为其中实例可不具有多个类的某一程序设计语言（如 C++，Smalltalk, Java）中的某一类规定实现。当实现类提供的具有如同为该类型的运算所规定的同一行为的某一类型来定义的全部操作时，即称该实现类实施了此类型。

参见：类型

3.2037

实施阶段 **implementation phase**

软件生存周期中根据设计文档创建软件产品并进行调试的时间段。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2183]

3.2038

实施评级模块 **implementation rating module**

可直接应用于目标实体的质量评级模块。

[来源：ISO/IEC 25040:2024, 3.2]

3.2039

实施要求 **implementation requirement**

规定或约束系统或系统组件的编码或构建的要求。

参见：设计需求、功能需求、接口需求、性能需求、物理需求

[来源：ISO/IEC/IEEE FDIS 24765, 3.2185]

3.2040

隐含的要求 **implied need**

可能未明确阐述，却是实际需要的要求。

示例：隐含的要求包括：尽管没有明确阐述但通过其他明确的要求而隐含的需要，以及由于被认为是显然的或显而易见的而未明确阐述的需要。

注：当软件产品用于特定场合时一些隐含的要求才会成为显然的。

[来源：GB/T 25000.40-2018, X.X]

3.2041

引入 **import**

它表明在包的语境中，一个指定的包（包括递归地嵌入其内的包）可以引用另外一些包中的类的一种依赖。

3.2042

导入流程 **import process**

将传输文件的内容合并到目标环境中的过程。

[来源：ISO/IEC 15474-1:2002, 4.1]

3.2043

导入代理 **importer**

导入流程代理。

[来源：ISO/IEC 15474-1:2002, 4.1]

3.2044

不可能区间 **impossible zone**

估计范围内，在任何情况下都不可能实现的区间那一部分区域。

示例：不可能在1小时内开车行驶500英里，因此对于“开车行驶500英里需要多长时间”的估算而言，“1小时”的结果位于不可能区间。

[来源：ISO/IEC/IEEE 24765:2017, 3.1900]

3.2045

可改进性 **improvability**

组织支持持续过程改进的能力。

[来源：ISO/IEC TR 33014:2013, 3.3]

3. 2046

待办改进事项 **improvement backlog**

有意识形成的、涌现式的、描述如何进行改进的事项有序列表。

[来源：ISO/IEC 33202:2024, 3.16]

3. 2047

软件过程改进计划 **Improvement plan**

从软件过程评估建议中导出的一种计划，它标识为改进软件过程而将采取的具体措施，并且概述实施这些措施的计划。

[来源：GB/T 11457 2.1516]

3. 2048

应用程序内编程 **in-application programming; IAP**

当系统运行时，微控制器单元获取新的程序代码并对自身进行重新编程的能力。

参见：系统内编程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2129]

3. 2049

偶发事件 **incident**

在项目、产品、服务或系统生存周期的任何时刻发生的异常或意外的事件、事件集、状况或情形。

[来源：ISO/IEC/IEEE 12207:2017, 3.1.23]

3. 2050

事件管理者 **incident manager; IM**

有权管理所有事件并管理事件相关任务的角色。

[来源：ISO/IEC TR 29110-5-3:2018, 3.14]

3. 2051

事件报告 **incident report**

记录事件的发生、性质和状态。

[来源：ISO/IEC/IEEE 29119-1:2013, 4.18]。

3. 2052

在线仿真器 **in-circuit emulator; ICE**

用于调试嵌入式系统软件的硬件设备。

[来源：ISO/IEC/IEEE 24765d:2015]

3. 2053

包括 **include**

在 UML 中，从基用例到被包含用例的关系，指定如何将基用例定义的行为插入到为基用例定义的行为中。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3. 2054

包容性 **inclusivity**

产品可供不同背景人群使用的能力。

示例：不同年龄、能力、文化、种族、语言、性别、经济状况、教育程度、地理位置与生活境遇的人群。

[来源：ISO/IEC 25010:2023, 3.4.6]

3. 2055

收入函数 income function

表征由决策变量的不同取值所产生收入的目标函数。

参见：成本函数。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3. 2056

不完整的过程 incomplete process

未执行或仅部分执行的过程。

注：过程的一个或多个特定目标未得到满足。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2207]

3. 2057

不一致率 inconsistency ratio

在层次分析法（AHP）中，衡量决策分析师将值分配给成对比较的一致性的函数。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2208]

3. 2058

增量 increment

经过测试、可交付的软件产品版本，提供新的或修改后的功能。

[来源：项目管理知识体系指南（PMBOK(R) Guide）-第五版]

3. 2059

增量分析 incremental analysis

考量备选方案之间的相对差异。

注：如果第二个备选方案相对于第一个备选方案的增量收益大于它们之间的增量投资，则第二个备选方案是比第一个方案更具投资价值的选择。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2210]

3. 2060

增量方法 incremental approach

一种自适应开发方法，通过逐步生成可交付成果并持续增加功能，直至可交付成果具备必要且充分的能力，方可视为完成。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3. 2061

增量收入 incremental benefit

备选方案相较于另一备选方案产生的额外收入。

注：如果备选方案 A 产生 10000 美元，备选方案 B 产生 12000 美元，则 B 对 A 的增量收入为 2000 美元。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3. 2062

增量编译程序 incremental compiler

一种编译程序，它在源语句输入或扫描时，尽可能多地完成每个源语句的转换。典型地用于联机计算机程序的开发和检验。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2213]

3. 2063

增量编译器 incremental compiler

会话编译器 conversational compiler

交互式编译器 interactive compiler

在线编译器 online compiler

在输入或扫描源语句期间尽可能完成每个源语句翻译的编译器。

注：通常用于联机计算机程序开发和检查。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2213]

3.2064

增量开发 incremental development

软件开发技术，其中需求定义、设计、实现和测试以重叠的、迭代的（而非顺序的）方式进行，从而实现整个软件产品的增量完成。参见：瀑布模型（3.5257）、以数据结构为中心的设计（3.1192）、输入-处理-输出（3.2311）、模块分解（3.2877）、面向对象的设计（3.3057）。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2065

增量投资 incremental investment

备选方案相较于另一备选方案、可避免的额外投资。

注：如果备选方案A的成本为10000美元，备选方案B的成本为12000美元，则B对A的增量投资为2000美元。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2066

增量生存周期 incremental life cycle

一种项目生存周期模式，其中项目范围通常在项目生存周期的早期阶段确定，但时间与成本的估算会随着项目团队对产品理解的加深而定期调整。迭代通过一系列重复的循环开发产品，而增量则会相继增加产品的功能。

[来源：项目管理知识体系指南（PMBOK®指南）——第五版]

3.2067

增量生产率 incremental productivity

开发过程中周期性计算的生产率。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2068

不定交付不定量合同 indefinite delivery indefinite quantity; IDIQ

在固定期限内，规定下限与上限、提供不确定数量的产品或服务的合同。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.2069

独立性 independence

就软件质量保证（SQA）而言，不受技术、管理和财务方面有意或无意影响的情况。

[来源：IEEE 730-2014, 3.2]

3.2070

独立的 independent

由不受供方、开发方、运营方或维护方控制的组织所执行的。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2071

独立估计 independent estimates

使用第三方获取和分析信息以支持成本、进度或其他项目预测的过程。

[来源：项目管理知识体系指南（PMBOK®指南）第五版]

3.2072

独立评估方 independent evaluator

独立于开发方和需方，实施某评价的个体和组织。

注：个体或组织作为待评价目标系统的开发方或需方时不能成为系统的独立评价方。独立评价方可以是某个组织，只要独立于开发方和需方。独立评价方可属于与开发方相同的组织。

[来源：ISO/IEC 25040:2011, 4.30]

3.2073

独立状态类 independent state class

非依赖状态类的状态类。

参见：依赖状态类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.85]

3.2074

独立验证和确认 independent verification and validation; IV&V

由在技术、管理和财务上独立于开发组织的机构所执行的验证和确认。

[来源：IEEE 1012-2024, 3.1]

3.2075

独立验证和确认(IV&V) independent verification and validation(IV&V)

由在技术上、管理上和财务上独立于开发组织的组织执行的验证和确认。

[来源：IEEE 1012-2024, 3.1]

3.2076

指标 index

反映某一概念的复合度量集。

[来源：IEEE 7010-2020, 2.1]

3.2077

预估功能点分析 indicative function point analysis

仅仅基于概念数据模型或者符合第三范式的数据模型得到一种表示应用程序或者项目的估算规模的说明。

注：预估 FPA 在本文件中是最不精确的计数类型。另外两种是详细功能点分析和估算功能点分析。

[来源：ISO/IEC 24570:2018, X.X]

3.2078

指示性功能点计数 indicative function point count

指示应用程序或项目的估计大小，完全基于概念数据模型或第三范式的数据模型

[来源：ISO/IEC 24570:2005, X.X]。

3.2079

指标 indicator

对由规定信息需要相关模型导出的指定属性，提供估算或评价的测度。

[来源：ISO/IEC 15939:2007]

3.2080

指标值 indicator value

赋予某一指标的数值型或分类型结果。

[来源：ISO/IEC/IEEE 15939:2017, 3.11]

3. 2081

内在差错 indigenous error

未作为错误播种过程的一部分而被有意插入的计算机程序错误。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3. 2082

间接开销 indirect cost

与达成软件项目目标相关，但同一种投入可以支持一个以上项目的联合资源代价总和。

注：如开发管理人员工资、开发设备折旧和停工损失等。

[来源：GB/T 36964-2018]

3. 2083

间接人工 indirect labour

与生产单位无直接关联的人力投入。

参见：直接人工（direct labour）。

[来源：ISO/IEC 24765 FDIS, 3. 2230]

3. 2084

间接量测 indirect measure

从一个或多个其他属性的度量值派生出的属性度量值。

注：计算系统属性（如对用户输入的响应时间）的外部度量值是软件属性的间接度量值，因为该度量值会受到计算环境属性以及软件属性的影响。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2231]

3. 2085

间接用户 indirect user

接收某一系统的输出，但不与系统交互的个人或组织。例如执行管理者、服务需方。

[来源：GB/T 25000.10-2016]

3. 2086

单个界面 individual interface

在某个时间点，所呈现的一个或多个用户界面元素和内容块的物理实现。

[来源：ISO TR 25060:2023 3. 2. 9]

3. 2087

归纳断言法 inductive assertion method

一种正确性证明技术，在该技术中编写描述程序输入、输出和中间条件的断言，建立一组将输入断言的满足与输出断言的满足相关联的定理，并使用归纳法证明或证伪这些定理。

[来源：ISO/IEC 20968:2002]

3. 2088

推理 inference

在指定的上下文中，从一组断言（前提）推导出项断言的推理步骤。

注：通过推理得到的断言为结论，推理所依据的已支撑断言为前提。存在一类特殊推理，称为显式推理。

[来源：ISO/IEC/IEEE 15026-2:2022, , 3. 1. 4]

3. 2089

影响图 influence diagram

对变量与成果之间的因果关系、事件时间顺序及其他关系的图形表示。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版, Glossary]

3. 2090

非正式小组审查 informal group review

由三人或三人以上共同进行的非正式评审。

[来源: ISO/IEC 20246:2017, 3.7]

3. 2091

非正式审查 informal review

一种不遵循规定过程且无正式书面输出的审查形式。

[来源: ISO/IEC 20246:2017, 3.6]

3. 2092

信息 information

〈开放分布式处理〉在特定论域内, 可在用户间进行交换的、关于事物、事实、概念等客体的知识。

[来源: ISO/IEC 10746-2:2009, 3.2.6]

〈信息处理〉在信息处理场景中, 关于客体(如事实、事件、事物、过程或思想, 包括概念)的知识, 在一定的场合中具有特定的意义。。

参见: 数据(data)。

注: 信息具备可传播的表示形式以实现传递, 但其核心关联要素为对该表示形式的解读(即语义内涵)。

[来源: ISO/IEC 2382:2015, X.X]

3. 2093

信息分析 information analysis

对实际或规划系统中的信息及其流动进行的系统性调研。

[来源: ISO/IEC 2382:2015, X.X]

信息架构 information architecture

(以人为中心的) 信息空间结构和访问所需任务对象、系统对象和其他信息的语义。

[来源: ISO/IEC TR 25060:2010 , 2.8]

3. 2094

信息内容 information content

在 CDIF 传输中找到的一组元模型和模型实例。

[来源: ISO/IEC 15474-1:2002 , 4.2]

3. 2095

信息设计 information design

开发满足受众需求的内容的过程。

[来源: ISO/IEC/IEEE 26511:2011, 4.11]

3. 2096

信息开发 information development

与确定产品文档中应提供哪些内容和视觉效果以及信息的性质有关的开发过程。

[来源: ISO/IEC/IEEE 26513:2017, 3.21]

3. 2097

使用信息 information for use

供应商提供的信息, 为目标受众提供在其生存周期内安全、有效和高效使用受支持产品的概念、程序和参考资料。

注: 不包括补充信息, 补充信息不在使用信息范围内。

[来源: IEC/IEEE 82079-1:2019 , 3.17]

3. 2098

用户信息 information for users

供应商提供的信息，为目标受众提供概念、程序和参考资料，以便在其生存周期内安全、有效和高效地使用受支持的产品。

注：可单独提供或嵌入产品中。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.29]

3. 2099

信息收集技术 information gathering techniques

用于收集和组织各种来源数据的可重复过程 [项目管理知识体系指南 (PMBOK® 指南) 第五版]。

3. 2100

信息隐蔽 information hiding

一种软件开发技术，其中，每一模块的接口尽可能少泄漏模块的内部工作并其他模块防止使用不在模块的接口规格说明中的关于模块的信息。

[来源：GB/T 11457 2.751]

3. 2101

信息项 information item

为人类使用而生产、存储和交付的可单独识别的信息体。

[来源：ISO/IEC/IEEE 15289:2017, 3.1.13, 有修改]

3. 2102

信息项内容 information item content

信息项中包含的信息，与系统、产品或服务相关，以满足要求或需要。

[来源：ISO/IEC/IEEE 15289:2015, 5.13]

3. 2103

信息项类型 information item type

一组符合预先安排的通用标准的信息项。[来源：ISO/IEC/IEEE 15289:2019, 5.14]。

3. 2104

信息管理计划 information management plan

为满足信息管理的各个项目过程而量身定制的计划。信息管理计划确定了要收集、接受、保护和传播的与系统相关的信息，并制定了处理计划。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2256, 有修改]

3. 2105

信息管理系统 information management system

用于以物理或电子格式在信息的生产者和消费者之间收集、存储和分发信息的设施、过程和程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2256, 有修改]

3. 2106

信息模型 information model

信息体的概念、关系和规则的表示。

[来源：IEC/IEEE 82079-1:2019, 3.18]

3. 2107

信息需要 information need

为管理目标、目的、风险和问题所需要的信息。

[来源：ISO/IEC 25000:2014, 4.14]

3.2108

信息部分 information part

为供人与机器使用而生成、存储和传递的、可单独识别的信息体。

参见：信息项（information item）。

[来源：GB/T 45630-2025, 3.14]

3.2109

信息处理要求 information processing requirements

应用软件产品的调试用户所需的功能集（不包括任何技术和质量要求）。

[来源：ISO/IEC 20968:2002, 1.1]

3.2110

信息处理系统 information processing system

一种或多种进行信息处理的数据处理系统和设备，如办公和通讯设备。

3.2111 [[来源：ISO/IEC 2382:2015, X.X]

信息产品 information product

说明信息需要的一个或多个指标及其相应的解释。

示例：已测的失效率与计划的失效率的比较结果连同对这种差异是否指示出问题的评估。

[来源：GB/T 25000.40-2018]

3.2112

信息供应 information provisioning

为业务提供信息的所有基础施工具、软件应用程序、非自动化元素、数据集、用户文档和组织结构的集合。

[来源：ISO/IEC 16350-2015, 4.20]

3.2113

信息发射源 information radiator

为组织内其他相关方提供信息、支撑知识及时共享的可视化物理展示载体。

示例：燃尽图、累积流图、停车场图。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版，Glossary]

[来源：ISO/IEC 24765 FDIS, 3.2266]

3.2114

信息检索 information retrieval; IR

从存储数据中获取给定主题信息的动作、方法和程序。

[来源：ISO/IEC 2382:2015]

3.2115

信息安全 information security

保护信息的机密性、完整性和可访问性。

3.2116

信息安全事件 information security incident

单个或一系列不需要或意外的信息安全事件，这些事件极有可能危及业务运营并威胁信息安全。

[来源：ISO/IEC 27000:2009]

3.2117

信息安全政策 information security policy

以书面形式说明组织计划如何保护其物理和信息技术资产的文件。

[来源：ISO/IEC TR 29110-5-3:2018, 3.15]

3.2118

信息结构 information structure

提供有关 IT 资产的信息以促进其管理的结构。

同义词：信息结构 (info struct)

[来源：ISO/IEC 19770-4:2017, 3.13]

3.2119

信息结构创建者 information structure creator

初始创建信息结构的实体。

示例：软件创建组织单元、为遗留软件创建信息结构的第三方组织。

同义词：信息结构创建者 (info struct creator)

[来源：ISO/IEC 19770-5:2015, 3.19]

3.2120

信息结构标识符 information structure id

每个信息结构被创建时被分配的全局唯一值。

同义词：信息结构标识符 (info struct ID)

[来源：ISO/IEC 19770-5:2015, 3.20]

3.2121

信息系统要求 information system needs

能通过外部测量、有时通过内部测量，规定为质量需求的要求。

[来源：GB/T 25000.40-2018, 4.34; ISO/IEC 25040: 2011, 4.34]

3.2122

信息技术资产 information technology asset; IT asset

物品、事物或实体，可用于获取、处理、存储和分发数字信息，并对组织具有潜在或实际价值。

示例：软件、媒体（包括物理和数字形式）；IT 设备（包括物理和虚拟设备）；许可证（包括许可证证明）；合同；IT 资产管理系统、工具以及用于管理所有 IT 资产的所需的数据元；满足 IT 资产管理要求的服务，如：软件即服务、硬件维护、软件支持和培训。

注：独立于 IT 硬件和软件资产的信息本身可以被视为资产，但不被视为 IT 资产。IT 资产的集合也被称为信息技术基础设施。

[来源：ISO/IEC 19770-1:2017, 3.25]

3.2123

信息技术资产管理体系 information technology asset management system; ITAMS

确立 IT 资产管理方针与 IT 资产管理目标。

[来源：ISO/IEC 19770-1:2017, 3.28]

3.2124

信息技术项目 information technology project

为创建或变更独特的信息技术产品、系统或服务而承担的临时性工作。

注：《项目管理知识体系指南》中将“项目”定义为“创建独特的产品、服务或结果而承担的临时性工作”。“IT 项目”是所规定项目子集。

同义词：IT 项目，信息技术 (IT) 项目

[来源：GB/T 41866.1-2022, 3.16]

3.2125

信息技术服务 information technology service

以信息技术系统为工具，通过助力用户或企业达成预期成果，进而为个人用户或企业提供价值的服务。

同义词：IT 服务（IT service）

[来源：ISO/IEC TS 25011:2017, 3.3.2]

3.2126

信息类型 information type

解决特定用户问题的主题类别。

[来源：ISO/IEC/IEEE 26531:2015, 4.23]

3.2127

信息视点 information viewpoint

有关 ODP 系统及其环境的视点，聚焦于信息语义和信息处理。

[来源：ISO/IEC 10746-3:2009, 4.1.1.2]

3.2128

信息领域 information-based domain; IBD

以信息为最核心资产的活动领域。

注：信息创建、操作与传播是信息领域内最重要的活动。典型基于信息的领域包括软件与系统工程、业务流程再造、知识管理。

[来源：ISO/IEC 24744:2014 3.1]

3.2129

知情同意 informed consent

在知晓可能后果的情况下授予的许可。

注：知情同意是在执行涉及个人身份信息的操作之前获得许可的过程。接受知情同意可以作为延长聘用的条件。

[来源：IEEE 7005 2021, 3.1]

3.2130

基础设施 infrastructure

〈软件与系统工程〉支持计算机系统与软件的设计、开发和改进的硬件和软件环境。

[来源：ISO/IEC/IEEE 12207:2017, 3.1.25; GB/T 8566-2022, 3.1.25]

〈DevOps 和 IT 运营管理〉支撑系统生存周期、维持信息技术（IT）服务所需的数据中心供电、冷却、物理安全等设施，以及网络、硬件和软件。

参见：信息技术基础设施（IT infrastructure）、软件工程环境（software engineering environment）。

注：不包含相关的人员或过程。在 DevOps 场景中，软件定义基础设施可实现弹性。

[来源：IEEE 2675-2021, 3.1]

3.2131

基础设施即代码 infrastructure as code; IaC

通过软件对基础设施组件进行定义、管理与资源供给。

注：在 DevOps 中，基础设施即代码有助于系统生存周期的自动化，实现跨系统和资源的一致性、性能和安全性。

[来源：IEEE 2675-2021, 3.1]

3.2132

基础设施生态系统 infrastructure ecosystem

支持系统生存周期和维护信息技术（IT）服务所需的设施，如电力、冷却、数据中心的物理安全、网络、硬件和软件。

参见：IT 基础设施、软件工程环境。

注：不包括相关的人员或过程。在 DevOps 中，软件定义的基础设施实现了弹性。

[来源：IEEE 2675-2021, 3.1]

3.2133

继承属性 `inherited attribute`

由于是通用祖先的属性而成为类特征的属性。

[来源：IEEE 1320.2-1998 (R2004) 3.1.87]

3.2134

抑制弧 `inhibitor arc`

一种反转输入位置逻辑的特殊类型的弧。

注：它不是测试相关位置是否存在某些标记，而是测试是否存在这些标记。

[来源：ISO/IEC 15909-3:2021, 3.4]

3.2135

初始授权项 `initial Ent`

被后续授权项所引用的授权项。

注：初始授权项通常为软件许可方与最终客户之间首次交易的记录。初始授权项属于主授权项的一种。

同义词：初始授权模式（initial entitlement schema）

[来源：ISO/IEC 19770-3:2016, 3.1.14]

3.2136

初期功能点分析 `initial function point analysis`

在项目开始时执行的功能点计数。

[来源：ISO/IEC 24570:2018]

3.2137

初始功能点计数 `initial function point count`

在项目开始时进行的功能点计数。

[来源：ISO/IEC 24570:2005, 2.1]

3.2138

初始投资 `initial investment`

启动一项活动所需的投资。

[来源：ISO/IEC/IEEE 24765:2017, 3.1976]

3.2139

(网的)初始标记 `initial marking (of the net)`

与高级网定义一并给出的初始库所标记集合。

[来源：ISO/IEC 15909-1:2019, 3.10]

3.2140

库所初始标记 `initial marking of a place`

随网一同定义的、某一库所的特定标记。

[来源：ISO/IEC 15909-1:2019, 3.11]

3.2141

初始风险 `initial risk`

在采取风险降低措施之前所估计的风险。

[来源：ISO/IEC/IEEE 15026-1:2019, 3.3.3]

3.2142

初始段 initialization section

在检验第一个条件之前，一个顺序执行的无条件动作的任选表列。

注：可写在紧靠表头之后的行中。

[来源：ISO 5806:1984, 3.13]

3.2143

初始化 initialize

将变量、寄存器或其他存储单元位置设置一起始值。

参见：清除（clear）、复位（reset）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2295]

3.2144

启动对象 initiating object

引起通信的对象。

[来源：ISO/IEC 10746-2:2009, 13.4.1]

3.2145

启动过程组 initiating process group

旨在通过获得启动项目或阶段的授权来定义新项目或现有项目的新阶段而执行的那些过程。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3.2146

启动过程 initiating processes

通过获得启动项目或阶段的授权来定义新项目或新阶段而执行的过程。

[来源：项目管理知识体系指南（PMBOK® 指南）——第七版]

3.2147

启动触发器 initiation trigger

负责激活预期事务的控制信息或功能控制数据。

注：启动触发器的例子有：按钮、功能键或菜单项。

[来源：GB/T 42588-2023, 3.142]

3.2148

主动性 initiative

信息技术服务识别用户目标，并主动提出变更建议以满足用户需求的程度。

[来源：ISO/IEC TS 25011:2017, 3.2.7.2]

3.2149

内部基数 inner cardinality

从担当某一角色的数据对象的单个实例的视角，关系所允许的实例数量。

参见：外部基数（outer cardinality）。

[来源：ISO/IEC 15476-4:2005, 6.6.2]

[来源：ISO/IEC 24765 FDIS, 3.2302]

3.2150

输入 input

将数据输入信息处理系统或其任何部分进行存储或处理的过程。

[来源: ISO/IEC 2382: 2015]

3.2151

(变迁的) 输入弧 **input arc (of a transition)**

从库所指向该变迁的有向弧。

[来源: ISO/IEC 15909-1:2019, 3.12]

3.2152

输入变量 **input argument**

调用时, 始终具有取值的自变量。

参见: 输出变量 (output argument)。

[来源: ISO/IEC/IEEE 24765:2017, 3.1991]

3.2153

输入断言 **input assertion**

一种逻辑表达式, 它规定: 为了有效, 程序的输入必须满足的一个或多个条件。参见: 循环断言、输出断言、归纳断言方法。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2306]

3.2154

输入回送 **input loopback**

同一图中的一个函数的输出作为另一函数输入所构成的回环。

[来源: ISO/IEC/IEEE 24765m:2024, 2.1.64]

3.2155

(变迁的) 输入库所 **input place (of a transition)**

通过输入弧与该变迁相连的库所。

[来源: ISO/IEC 15909-1:2019, 3.13]

3.2156

输入原语 **input primitive**

以工时为单位表示的、开发软件产品所需的工作量。

[来源: ISO/IEC/IEEE 24765:2017, 3.1996]

3.2157

输入程序 **input routine**

获取下一个要处理的逻辑记录所需的那些活动。

3.2158

输入-输出耦合 **input-output coupling**

一个软件模块的输出作为另一个模块的输入的一种耦合类型。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1144]

3.2159

输入-处理-输出 **input-process-output**

一种软件设计技术, 它由标识在每一要执行的处理中包括的步骤并标识每一步的输入和输出组成。

参见: 以数据结构为中心的设计 (data structure-centered design)、输入—处理—输出图 (input-process-output chart)、模块分解 (modular decomposition)、面向对象的设计 (object-oriented design)、快速原型 (rapid prototyping)。

注: 称为分层的输入-处理-输出的细化在通用和详细级上标识步骤、输入和输出。

[来源: ISO/IEC/IEEE 24765:2017, 3.1998]

3.2160

输入-处理-输出图 input-process-output chart; IPO chart

一种软件系统或模块的图示：由左侧列出输入的矩形、中间列出处理步骤的矩形、右侧列出输出的矩形组成，并用箭头将输入连接至处理步骤、处理步骤连接至输出。

参见：框图(block diagram)、盒图(box diagram)、泡图(bubble chart)、流程图(flowchart)、图(graph)和结构图(structure chart)。

同义词：IPO图。

注：总结过程活动及其与外部参与者之间的输入和输出；部分输入被归类为控制项和使能项。

[来源：ISO/IEC 24765 FDIS, 3.2312]

3.2161

在役 in-service

系统生存周期的一部分，该部分包括建立、维持和停用一个 SOI 实例而进行的支持行为，以及该实例为其预期目标而进行的主动使用。

注：“在役”模式由支持中、阻塞中、准备完毕、运行中和无法运作等模式组成，可与开发中/维护中 并存，通常以卸载结束，进入已移除模式。

[来源：IEEE 982:2024, 3.1]

3.2162

在役期/使用期 in-service period; ISP

从系统首次安装开始，到下列任一事件最先发生时结束的时间段：1) 被该系统本质上不同的版本替换；2) 被卸载、废弃；或 3) 其运行时或运行环境消亡。

[来源：IEEE 982:2024, 3.1]

3.2163

在役参考模型 in-service reference model; ISRM

描述在役软件系统的支撑活动、事件与模式的模型，用于支持对受评对象行为的分类，以及可信性度量所需在役时间段的子集划分。

[来源：IEEE 982:2024, 3.1 与 3.2]

3.2164

审查 inspection

一种静态分析技术，它依靠对开发产品的可视的检查，以检测错误、对开发标准的违反和其他问题。类型包括：代码审查、设计审查。

[来源：GB/T 11457, 2.764]

3.2165

检查评估 inspection-based evaluation

基于一位或多位评估人员的判断来进行的评估，评估人员检查或使用系统以识别潜在的可用性问题，包括与既定标准的偏差。

示例：启发式评估、认知走查、标准检查、多元走查、一致性检查。

注：进行检查的评估人员通常为可用性专家，但也可包括最终用户和设计团队成员。在某些情况下，基于检查的评估可以由机器进行，例如在评估与所需术语的一致性时。既定标准通常包括用户需求、标准中的可用性指南、制造商指南和样式指南中包含的设计约定、要支持的任务模型以及标准化原则。

[来源：ISO/IEC 25066: 2016, 3.10]

3.2166

检查和审计 inspections and audits

根据商定的要求观察合同工作或承诺产品绩效的过程。

[来源：项目管理知识体系指南（PMBOK® 指南）]

3.2167

易安装性 installability

产品或系统在指定环境中成功安装或卸载的有效性和效率程度。

[来源：ISO/IEC 25010:2011，4.2.8.2]

3.2168

易安装性测试 installability testing

可移植性测试的一种类型。用于评价一个或一组测试项是否可以安装在所有指定的环境下。

[来源：GB/T 38634.1-2020]

3.2169

安装与检验阶段 installation and checkout phase

软件生存周期中的一个时间段，在此期间软件产品会被集成到其运行环境中，并在该环境中进行测试，以确保其能够按要求运行。

[来源：ISO/IEC/IEEE FDIS 24765，3.2316]

3.2170

安装检验阶段 installation and check-out phase

安装检验阶段。

[来源：ISO/IEC/IEEE FDIS 24765，3.2316]

3.2171

安装手册 installation manual

提供为安装系统或部件、设置初始参数和为了操作使用准备系统或部件所必需的信息的文档。参见：2.476 条 诊断手册（diagnostic manual）、2.1068 条 操作员手册（operator manual）、2.1250 条 程序员手册（programmer manual）、2.1631 条 支持手册（support manual）和 2.1813 条 用户手册（user manual）。

[来源：GB/T 11457，2.766]

3.2172

已安装的功能点计数 installed function point count

与一组已安装的系统有关的应用程序功能点计数。

[来源：GB/T 42566-2023]

3.2173

基准度量实例 instance of benchmarking

按给定方法在执行特别基准度量（3.3）中使用，并经特定描述的操作集。

[来源：GB/T 41866.1-2022]

3.2174

实例级属性 instance-level attribute

从一个类的实例映射到一个值类的实例。

[来源：ISO/IEC/IEEE FDIS 24765 3.2321]

3.2175

实例级操作 instance-level operation

从类的实例和输入参数类型的实例到其他（输出）参数类型的实例的（叉积）映射。

[来源：ISO/IEC/IEEE FDIS 24765 3.2322]

3.2176

实例化跟踪链接 instantiated trace link

通过在成员产品中进行应用绑定操作所导出的追踪链接。

[来源：ISO/IEC 26559:2017, 3.2]

3.2177

实例化 instantiation

将特定数据、指令或两者替换为通用程序单元以使其可用于计算机程序的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2325]

3.2178

机构知识 institutional knowledge

来自公认来源的知识，包括标准、学术资源、领域和行业知识体系以及组织知识。

[来源：ISO/IEC 30103:2015, 3.4]

3.2179

使用指导信息 instructional information

说明如何使用产品、系统或服务执行相关任务的信息。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.30]

3.2180

插装 instrument

在软件和系统测试中，将设备或指令安装或插入硬件或软件以监控系统或组件的运行。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2336]

3.2181

插装[工具] instrumentation

安装或插入硬件或软件以监控系统或组件运行的设备或指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2337]

3.2182

在系统编程 in-system programming; ISP

微控制器允许用户下载新代码（对单元重新编程）的能力，通过重启单元激活。

参见：在应用程序编程。

[来源：ISO/IEC /IEEE 24765 FDIS” 或 “ISO/IEC/IEEE FDIS 24765”]含序号 38388、390、3940420、426、430、435

3.2183

集成电路 integrated circuit; IC

内部包含相互连接的电子元件的一小块半导体材料。

同义词： 芯片(chip)、 微芯片(microchip)

[来源：ISO/IEC 2382:2015, X.X]

3.2184

集成电路 integrated circuit; IC

芯片 chip

微芯片 microchip

包含互连电子元件的小型半导体材料。

[来源: ISO/IEC 2382:2015]

3.2185

集成开发环境 integrated development environment; IDE

为软件开发提供全面支持的一组软件工具或应用程序。

[来源: ISO/IEC 24765 FDIS, 3.2341]

3.2186

集成开发环境(IDE) integrated development environment (IDE); IDE

为软件开发提供完备支撑能力的一组软件工具或应用程序。

[来源: ISO/IEC/IEEE 24765d:2015]

3.2187

集成储藏库 integrated repository

与系统工作相关的信息的计划和受控存储。

注: 集成存储库通常包括关键数据, 例如模式、模型、工具、技术管理决策、过程分析信息、需求变更、过程 and 产品指标、权衡和其他分析。

[来源: ISO/IEC/IEEE 24748-4:2016, 4.6]

3.2188

集成 integration

将软件组件、硬件组件或两者组合成一个整体系统的过程。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2189

功能建模的集成定义 integration definition for functional modeling; IDEF

系统和软件工程领域的一系列建模语言, 提供功能、数据和过程描述 (IDEF0 到 IDEF5)。

[来源: INCOSE. Appendix.C.Terms and Definitions, 有修改]

3.2190

集成计划 Integration plan

定义已实现的系统元素的预期集合的配置以及这些集合的组装顺序, 以执行有效的验证操作和确认操作(例如, 检查或测试)。

[来源: 系统工程手册]

3.2191

集成测试 integration testing

把软件部件、硬件部件或两者组合起来进行的测试, 并测试以评价它们之间的交互。参见: 部件测试、接口测试、系统测试和单元测试。

[来源: ISO/IEC 29110-5-1-2:2025, 3.16]

3.2192

完整性级别 integrity level

项目的某个特性的取值范围的一种表示, 该特性的取值范围表示子系统或子模块对整体软件项目可能带来风险的影响程度。

Note 6: 注: 改写自 GB/T 18492-2001 定义 3.9。

[来源: GB/T 36964-2018]

3.2193

完整性级别声明 integrity level claim

关于相关系统的风险处理过程中所识别的某项风险降低措施的要求声明。

[来源: ISO/IEC/IEEE 15026-3:2023, 3.4]

3.2194

完整性级别定义权威 integrity level definition authority

负责定义完整性级别和完整性级别要求的个人或组织。

[来源: ISO/IEC/IEEE 15026-3:2023, 3.3]

3.2195

完整性等级要求 integrity level requirements

一组要求, 当满足时, 将提供与其所对应完整性级别相称的完整性级别声明的可信度。

[来源: ISO/IEC 15026-1:2025, 3.3.5]

3.2196

完整性级别方案 integrity level scheme

由一组对利益相关方至关重要的系统特性构成, 并按性能或合规程度划分为离散的等级, 用于确定软件开发或交付过程中应实施的质量控制级别。

[来源: ISO/IEC/IEEE FDIS 24765 Systems and software engineering-Vocabulary, 3.2355]

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2197

智能体 intelligent agent

能够感知环境并对其做出响应, 并采取行动以实现其目标的自动化实体。

[来源: ISO/IEC 22989:2023, 3.1.1, 有修改]

3.2198

智能视觉系统 intelligent vision system; IVS

智能视觉系统。

[来源: ISO/IEC/IEEE 24748-9:2023, 3.2]

3.2199

预期用途 intended use

由产品供应商定义和设计的一整套功能或预期应用范围。

注: 供应商未列出的功能或应用不包括在产品的预期用途中。未经产品供应商批准的修改所产生的附加或修改功能或应用不包括在预期用途中。

[来源: IEC/IEEE 82079-1:2019, 3.21]

3.2200

交互能力 interaction capability

产品被指定用户进行交互, 通过用户界面在用户与系统之间交换信息, 以完成预定任务的能力。

注: 交互能力是易用性的前提。交互能力使得用户能够在各类使用场景下通过交互完成特定任务; 而易用性则关注使用结果, 判断用户在特定使用场景下是否有效、高效且满意地完成任务。

[来源: ISO/IEC 25010:2023, 3.4]

3.2201

互动小组 interaction group

群体功能所管理的绑定关系中, 参与对象的一个子集。

[来源: ISO/IEC 10746-3:2009, 13.4.1.1]

3.2202

相互作用点 interaction point

一组接口所在的位置。

[来源: ISO/IEC 10746-2:2009, 8.12, 有修改]

3.2203

交互序列 interaction sequence

用户与交互系统之间通过用户界面进行的信息交换, 以完成预定任务或在交互系统中进行导航。

[来源: ISO TR 25060:2023, 3.2.3]

3.2204

交互式 interactive

系统或操作模式, 其中每个用户输入都会引起系统的响应或动作。

[来源: ISO/IEC 20968:2002]

3.2205

交互语言 interactive language

非过程语言, 通过用户和计算机系统之间的交互对话创建程序, 参见: 2.417 条 声明语言 (declarative language) 和 2.1403 条 基于规则的语言 (rule-based language)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2365]

3.2206

交互系统 interactive system

指这样一个系统。在这种系统中, 每一个用户的输入均能得到该系统的响应。

[来源: GB/T 11457, 2.794]

3.2207

互换参考点 interchange reference point

可以将外部物理存储介质引入系统的参考点。

[来源: ISO/IEC 10746-2:2009, 15.3.4, 有修改]

3.2208

界面, 接口 interface

一个共享的边界。信息跨越边界传送。

连接两个或多个其他部件, 为了相互间传送信息的硬件或软件部件。

连接两个或多个部件, 为了在相互间传送信息。

作为如 b 中连接的或被连接的部件。

[来源: GB/T 11457, 2.795]

3.2209

接口控制 interface control

在配置管理中, (1) 标识由一个或多个组织提供的两个或多个配置项的接口的所有功能的和物理的特征和 (2) 保证对这些特征的的建议的变更在实现前评价和批准的过程。

在配置管理中, 管理的和技术的过程和文档需要在由不同的开发者提供的配置项之间或配置项之内标识功能的和物理的特征并解决关于规定的接口的问题。参见: 2.307 条 配置控制 (configuration control)。

[来源: GB/T 11457, 2.796]

3.2210

界面控制文件 Interface control document; ICD

描述在系统、子系统和其他元素之间的边界处必须加以控制的特性的规格说明。

[来源: ISO 15388:2012, 3.1.27, 有修改]

3.2211

接口设计文档 interface design document; IDD

描述系统与组件之间接口架构与设计的文档。

参见：接口需求规格说明（IRS）。

注：描述内容包括控制算法、协议、数据内容与格式、性能等。

[来源：IEEE 1012-2024 3.1.16]

3.2212

接口储藏库 interface repository; IR

接口信息的存储位置。

[来源：ISO/IEC 19500-2:2012, 3.2.9]

3.2213

接口需求 interface requirement

指定系统或系统组件必须与之交互的外部项目的要求，或规定格式、时间或由此类交互引起的其他因素的限制。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2371]

3.2214

接口需求规格说明 interface requirements specification; IRS

规定系统与组件之间或系统、组件相互之间接口需求的文档。

参见：接口规格说明、接口设计文档。

注：这些需求包括对格式和时序的约束。

[来源：IEEE 1012-2024, 3.1.17]

3.2215

界面角色 interface role

一种社区角色，指涉及社区外对象参与的、具有标识意义的特征行为。

[来源：ISO/IEC 15414:2015, 6.3.5, 有修改]

3.2216

接口特征标记 interface signature

与接口交互相关的一组动作特征标记。

[来源：ISO/IEC 10746-2:2009, 9.15, 有修改]

3.2217

接口规格说明 interface specification

规定已存在的或计划的系统或部件的接口需求的规格说明。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2375]

3.2218

接口任务 interface task

作为应用程序一部分的任务，它与外部环境交互。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2376]

3.2219

中期功能点分析 interim function point analysis

在新的功能开发或者功能完善项目进行期间，为确定完成的临时功能完善规模进行的计数。

示例：为了确定对于功能规约的一个添加、更改或者删除的范围所进行的计数。

注：在应用程序功能点计数和项目功能点计数中的发生的改变都是临时功能点计数的目标。

[来源: GB/T 42588-2023, 3.1.43]

3.2220

集成电路间总线 inter-integrated circuit bus; I2C; IIC

一种为集成电路之间提供通信链路的双向二线串行总线。

[来源: ISO/IEC 24765 FDIS, 3.2358]

3.2221

内部集成电路总线 (I2C) inter-integrated circuit bus (I2C)

提供集成电路之间通信链路的双向两线串行总线。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2358]

3.2222

交错 interleave

将一个序列的元素与一个或多个其他序列的元素交替, 以便每个序列保持其身份。

示例: 交替执行两个不同任务的步骤, 以实现任务的并发操作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2378]

3.2223

中间产品 intermediate product

中间软件产品 intermediate software product

中间系统产品 intermediate system product

开发过程中, 用作其他阶段输入的系统或软件产品。

[来源: GB/T 25000.41-2018 (ISO/IEC 25041:2012, MOD)]

3.2224

中间软件产品要求 intermediate software product need

能通过内部测量, 规定为质量需求的要求。

[来源: GB/T 25000.40-2018]

3.2225

间歇故障 intermittent fault

组件中的临时或不可预测的故障。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2382]

3.2226

内部动作 internal action

在没有对象环境参与的情况下发生的动作。

[来源: ISO/IEC 10746-2:2009, 8.3, 有修改]

3.2227

内部箭头 internal arrow

两端 (来源和使用) 连接到图中方框的箭头。

参见边界箭头。

[来源: IEEE 1320.1-1998 (R2004), 2.1.65]

3.2228

内部属性 internal attribute

某一实体的可测量属性, 该属性完全可以通过该实体自身来推导得出。

注: 内部属性是指与软件的内部结构及其开发过程相关的属性。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.2385]

3. 2229

内部依赖关系 internal dependency

两个或多个项目活动之间的关系。

[来源：项目管理知识体系指南（PMBOK®Guide）—第七版]

3. 2230

内部逻辑文件 internal logical file; ILF

一组用户可辨认的在被测应用程序边界内维护的逻辑相关数据或控制信息。

参见：外部接口文件。

注：内部逻辑文件的主要目的是通过被测应用的一个或多个基本过程维护数据。内部逻辑文件是一种基本功能部件。

[来源：GB/T 42449-2023, 3. 39, 有修改]

3. 2231

内部度量（软件质量） internal measure (of software quality)

软件产品在指定条件下使用时，其静态属性满足明确和隐含要求的程度的测度。

示例：代码行数、复杂度和在走查中发现的故障数，都是产品自身的软件内部质量测度。

注 1：静态属性包括那些与软件架构、结构及其组件有关的属性。

注 2：静态属性可通过评审、审查、模拟和/或自动化工具来验证。

[来源：ISO/IEC 25000:2014, 4. 16]

3. 2232

内部服务提供方 internal service provider

向超小型实体（VSE）内部提供服务的个人或组织。

[来源：ISO/IEC 29110-1-2:2024, 3. 49]

3. 2233

内部任务结构标准 internal task-structuring criteria

任务构建标准的类别，说明内部对象如何映射到内部任务以及何时激活内部任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2392]

3. 2234

内部可变性 internal variability

从工程师的角度定义的可变性，对客户不可见。

[来源：ISO/IEC 26555:2015, 3. 6]

3. 2235

物联网 internet of things; IoT

信息社会全球基础设施将基于现有和正在出现的、信息互操作和通信技术的物质相互连接，以提供先进的服务。

注 1：通过使用各项能力，IoT 充分利用物体提供服务，同时确保满足安全和隐私要求。

注 2：从广义而言，IoT 可被视为技术和社会影响方面的愿景。

[来源：Recommendation ITU-T Y. 2060 (06/2012) | 有修改]

3. 2236

互操作能力、互操作性 interoperability

产品与其他产品交换信息，并相互使用所交换信息的能力。

[来源：ISO/IEC 25010:2023, 3. 3. 2]。

2. 各对象协同工作的能力；即相互进行信息通信，以实现事件、提议、请求、结果、承诺及业务流交互的能力。

[来源：ISO/IEC 10746-2:2009, 9.5]。

3. 各功能单元之间可实现通信、程序执行、数据传输，且用户无需了解这些单元自身独有特性即可使用的能力。

[来源：ISO/IEC 2382:2015]。

参见：兼容性。

注：在表述中优先使用互操作性（Interoperability）而非兼容性（compatibility），以避免与可替换（replaceability）产生语义歧义。

3.2237

互操作性测试 interoperability testing

为确保修改后的系统仍具备与异构系统交换信息、并使用所交换信息的能力而开展的测试。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2397]

3.2238

互操作系统 interoperating system

与所关注的系统交换信息并使用所交换信息的系统。

[来源：ISO/IEC/IEEE 15288:2023 3.20; ISO/IEC/IEEE 12207:2026 3.1.34]

3.2239

解释 interpret

逐条翻译并执行计算机程序的每条语句或构造单元，翻译执行完一条再进行下一条。

参见：汇编、编译。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2401]

3.2240

可解释性 interpretability

对底层技术工作原理的理解程度。

[来源：ISO/IEC TR 29119-11:2020, 3.1.42]

3.2241

解释程序（解释器） interpreter

一种计算机程序，逐句翻译并执行计算机程序的每条语句或程序结构，完成一条再翻译执行下一条。

参见：汇编程序、编译程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2403]

3.2242

解释代码 interpretive code

以解释器可以识别和处理的形式表达的计算机指令和数据定义。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2404]

3.2243

相互关系图 interrelationship digraphs

一种管理工具，用于描绘复杂情境中各因素之间的关系；又称“关联图”或“关系图”。

[来源：American Society for Quality Quality Glossary]

3.2244

询问 interrogation

由客户端对象发起的一次交互（即调用）组成的交互，导致将信息传送到服务器对象，并请求执行一项功能；然后是第二次交互（即终止），导致将信息从服务器对象传送到客户端对象以响应调用。

示例：请求/响应交换。

注：在询问中，调用和终止始终是配对的。公告没有终止，因此不存在由调用及一系列关联终止组成的操作。

[来源：ISO/IEC 10746-3:2009, 7.1.4, 有修改]

3.2245

中断控制器 interrupt controller

确定中断请求的来源和优先级并管理其执行的功能单元（集成电路）。

[来源：ISO/IEC/IEEE 24765d:2015]

3.2246

中断潜伏期 interrupt latency

计算机系统收到中断请求和处理请求之间的延迟。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2408]

3.2247

中断掩码 interrupt mask

用于通过保留或抑制代表中断请求的位来启用或禁用中断的掩码。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2409]

3.2248

中断优先级 interrupt priority

分配给给定中断请求的重要性。

注：这个重要性决定了请求是否会导致当前进程挂起，如果有几个未完成的中断请求，将首先处理。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2410]

3.2249

中断请求 interrupt request

请求暂停当前执行的进程以允许执行另一个进程的信号或其他输入。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2411]

3.2250

中断服务例程 interrupt service routine ISR

通过存储关键寄存器的内容来响应中断请求的例程，执行中断请求所需的处理，恢复寄存器内容，并重新启动被中断的进程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2412]

3.2251

间隔标度 interval scale

一种测量尺度，其测量值之间的单位距离是相等的，且这些相等的数值间距对应于被测量属性（特征）的等量差异。

参见序数标度、标称标度、比率标度。

示例：圈复杂度的最小值为 1，但每个增量代表一条附加路径。零值是不可能的。

[来源：ISO/IEC/IEEE 24765:2017, 有修改]

3.2252

可干预性 intervenability

操作员能够及时干预人工智能系统运行，以防止伤害或危险的程度。

[来源：ISO/IEC 25059:2023 3.2.4]

3.2253

访谈 interviews

访谈是通过直接与利益相关方交谈来获取信息的一种正式或非正式方法。

注：它通常通过询问预先准备的及自发的的问题并记录响应来执行。访谈通常在访谈者与受访者之间以个人形式进行，但也可能涉及多个访谈者和/或多个受访者。访谈有经验的项目参与者、发起者及其他高层管理人员以及领域专家，有助于识别和定义所需产品交付物的特征和功能。访谈对于获取机密信息也很有用。

[来源：项目管理知识体系指南（PMBOK Guide）Fifth Edition, 5.2.2.1]

3. 2254

互通参考点 interworking reference point

可以建立接口以允许两个或多个系统之间进行通信的参考点。

[来源：ISO/IEC 10746-2:2009, 15.3.3]

3. 2255

内在的、固有的 intrinsic

规定某一属性为全集（即必选）、单值、常量的规范说明。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2417]

3. 2256

内在关系 intrinsic relationship

从（至少）一个参与类的视角看，是完全的、单值的且恒定的联系，该类被称为依赖类。

参见：非内在联系。

示例：一个事务与其相关的账户具有内在联系，因为一个事务实例“切换”到另一个不同的账户是没有意义的，因为那会改变该事务本身的性质。

注：此类联系被认为是依赖类本质的一个组成部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.95;ISO/IEC/IEEE 24765 FDIS, 3.2418]

3. 2257

不变式 invariant

计算机程序的特定段或在一规定点始终为真（true）的断言。

[来源：ISO/IEC 10746-2:2009]

3. 2258

不变模式 invariant schema

一组关于一个或多个信息对象的谓词必须始终为真。

注：谓词约束它们所应用的对象的可能状态和状态变化。因此，不变模式是一个或多个信息对象的类型的规范，无论对象可能表现出什么行为，这些信息对象总是会得到满足。

[来源：ISO/IEC 10746-3:2009 6.1.1]

3. 2259

招标邀请 invitation for bid; IFB

向供应商发出的价格请求，其中可交付成果为指定商品，价格为唯一判定因素。

[来源：ISO/IEC 24765 FDIS, 3.2422]

3. 2260

招标书 invitation to tender; ITT

招标书。

[来源：ISO/IEC/IEEE 15289:2019, 3.2]

3. 2261

调用传递 invocation deliver

服务器计算对象的隐式定义信号接口中的信号，其名称和参数与原始操作接口中的询问或通知调

用相同。

[来源：ISO/IEC 14752:2000 3.3.8]

3.2262

调用提交 invocation submit

客户端计算对象隐式定义的信号接口中的信号，其名称和参数与原始操作接口中询问或通知的调用相同。

[来源：ISO/IEC 14752:2000 3.3.7]

3.2263

输入-处理-输出图 IPO diagram

软件系统或模块示意图，其结构为：左侧矩形列出输入，中间矩形列出处理步骤，右侧矩形列出输出，箭头连接输入与处理步骤以及处理步骤与输出。

参见：框图、盒图、气泡图、流程图、图形、结构图。

注：该图概括了流程活动及其与外部参与者的输入/输出关系；部分输入被归类为控制因素和使能因素。

[来源：ISO/IEC 24765 FDIS, 3.2422]

3.2264

等时性 isochronicity

序列中相邻动作对之间的关系，其中每个相邻动作对在时间上占据唯一的、大小相等的相邻间隔。

[来源：ISO/IEC 10746-2:2009, 11.3.2]

3.2265

隔离 isolation

对于云服务，在共享物理和虚拟资源的情况下，计算和数据与其他客户隔离和无法访问的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.5.3]

3.2266

问题 issue

问题跟踪系统中描述问题或改进的唯一可识别条目。

[来源：ISO/IEC 20246:2017, 3.9]

3.2267

信息技术资产管理 IT asset management; ITAM

用于信息技术（IT）资产管理的管理体系，其功能是制定信息技术资产管理方针和信息技术资产管理目标。

[来源：ISO/IEC 19770-1:2017 3.28]

3.2268

IT 资产管理计划 IT asset management plan; information technology asset management plan

规定单项 IT 资产或一组 IT 资产的活动、资源和进度的文件化信息，旨在实现组织的 IT 资产管理目标。

[来源：ISO/IEC 19770-1:2017, 3.27]

3.2269

IT 资产管理系统 IT asset management system; ITAMS

用于信息技术（IT）资产管理的管理体系，其功能是建立 IT 资产管理方针和 IT 资产管理目标。

[来源：ISO/IEC 19770-1:2017, 3.28]

[来源：同义词：信息技术资产管理系统]

3.2270

IT 资产管理体系 IT asset management system; information technology asset management system; ITAMS

IT 资产管理方面用于建立 IT 资产管理方针和 IT 资产管理目标的管理体系。

[来源: ISO/IEC 19770-1:2017, 3.28]

3.2271

IT 资产管理 IT asset management; information technology asset management; ITAM

组织为从 IT 资产中获取价值而开展的协同活动。

[来源: ISO/IEC 19770-1:2017, 3.26]

3.2272

IT 资产组合 IT asset portfolio; information technology asset portfolio

属于 IT 资产管理体系范围内的 IT 资产。

注: 通常会建立并分配一个资产组合以用于管理控制目的。IT 硬件的组合可以根据类别进行定义(如: 服务器、个人电脑、移动设备)。软件组合可以根据软件发布商或平台进行定义(如: 个人电脑、服务器、大型机)。IT 资产管理体系可以涵盖多个 IT 资产组合。

[来源: ISO/IEC 19770-1:2017, 3.29]

3.2273

IT 基础设施 IT infrastructure

用于提供信息的所有技术组件、系统软件、数据库和数据文件以及已部署的应用软件、技术程序和技术文档。

[来源: ISO/IEC 16350-2015, 4.22]

3.2274

IT 基础设施管理 IT infrastructure management; information technology infrastructure management

负责所有旨在管理、维护和更新信息系统 IT 基础设施(包括信息系统的运行)的任务和活动的部门。

注 1: IT 基础设施管理包括旨在使信息系统正确技术运行的所有任务、职责和活动, 包括硬件、(系统)软件和数据集的管理。

注 2: IT 基础设施管理组织负责在生产环境中运行应用软件。

[来源: ISO/IEC 16350-2015, 4.23]

3.2275

IT 项目数据集 IT project dataset; information technology project dataset

依据预先界定的准则选出、收集起来, 并带有类别的数据记录组。

注: IT 项目数据集的分类准则可基于数据的拥有方和/或用户的信息要求。

[来源: GB/T 41866.1-2022, 3.17]

3.2276

IT 服务 IT service

以信息技术(IT)系统为工具, 通过助力用户或业务达成其期望实现的成果, 从而为个人用户或业务创造价值的服务。

参见: 信息技术服务。

注: 使用者包括顾客和服务提供者。

[来源: ISO/IEC 25002:2024, 3.11]

3. 2277

IT 服务适应性 IT service adaptability; information technology service adaptability

IT 服务能够自行配置或进行修改以满足新需要的程度。

[来源: ISO/IEC TS 25011:2017, 3.2.7]

3. 2278

IT 服务功能 IT service function; information technology service function

作为 IT 服务的一部分而执行的相关步骤的集合, 或 IT 系统提供的功能。

示例: 互联网银行服务的服务状态监测或数据备份。

注: ISO/IEC 拥有一套软件功能识别、分类和规模度量的标准方法, 这些方法为识别独特的 IT 服务功能提供了一致性, 包括: GB/T 42449 (IFPUG 方法)、GB/T 42452 (COSMIC 方法)、GB/T 42448 (FisMA 方法)、ISO/IEC 20968 (MarkII 方法)、GB/T 42588 (NESMA 方法)。

[来源: ISO/IEC TS 25025: 2021, 3.4]

3. 2279

IT 服务体系 IT service system

由 IT 服务、使用该服务的人员(用户)在给定(用户)环境中构成, 用以满足其服务需求的体系。

[来源: ISO/IEC 25002:2024, 3.12]

3. 2280

迭代 iteration

重复执行一系列步骤的过程。

[来源: ISO/IEC/IEEE 26515: 2011, 4.7]

一个或多个软件开发活动的系统重复。

注: 一次或多次迭代构成一个实例。

[来源: ISO/IEC/IEEE FDIS 24765 : 2025, 3.2470]

3. 2281

迭代待办事项 iteration backlog

为纳入当前迭代而选定的待办事项子集。

[来源: ISO/IEC 33202:2024, 3.19; ISO/IEC/IEEE FDIS 24765, 3.2471]

3. 2282

迭代计划 iteration plan

当前迭代的详细计划。

[来源: 项目管理知识体系指南 (PMBOK® 指南)-第七版]

3. 2283

迭代规划 iteration planning

用于澄清待办项细节、验收标准以及为达成下一代承诺所需工作量的会议。

[来源: 项目管理知识体系指南 (PMBOK® 指南)-第七版]

3. 2284

迭代评审 iteration review

在迭代结束时召开的会议, 用于展示迭代期间已完成的工作成果。

[来源: 项目管理知识体系指南 (PMBOK® 指南)-第七版]

3. 2285

迭代方法 iterative approach

以初始简化实现为核心, 逐步细化完善、扩充功能集合, 直至最终交付物完整的开发方法。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3.2286

迭代结构 **iterative construct**

一计算机程序语句序列，会在某一给定条件满足前或条件为真时反复执行。参见：循环体(loop body)、循环控制(loop control)、UNTIL 语句、WHILE 语句。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2628]

3.2287

迭代开发 **iterative development**

并行规划、开发和测试活动的重复使用。

[来源：ISO/IEC/IEEE 26515: 2018 , 3.11]

3.2288

迭代生存周期 **iterative life cycle**

一种项目生存周期，项目范围通常在项目生存周期的早期确定，但时间和成本估算会随着项目团队对产品了解的增加而定期修改。通过一系列重复周期迭代开发产品，而每个增量阶段都会逐步增加产品功能。

[来源：项目管理知识体系指南（PMBOK® 指南） 第五版]

3.2289

迭代过程 **iterative process**

对输出进行评估并将其用作下一个周期输入的重复过程。

[来源：IEEE 7002:2022, 3.1]

3.2290

作业 **job**

由计算机实现的用户定义的工作单元。参见：作业控制语言(job control language)、作业步(job step)、作业流(job stream)。

示例：计算机程序的编译、装入和执行。

[来源：GB/T 25000.23-2019, 4.3]

3.2291

作业控制语言 **job control language; JCL**

用于标识作业序列、向操作系统描述其要求并控制其执行的语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2488]

3.2292

作业功能 **job function**

一组被识别为用于工作组织、任务分配或评估单位的工程过程。

示例：设计、测试、配置管理。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2489]

3.2293

作业流 **job stream**

程序或作业设置序列，它使计算机能逐个自动处理而不需操作员的介入。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2491]

3.2294

JSON web 令牌 **JSON (JavaScript Object Notation) web token; JWT**

JSON web 令牌。

[来源：ISO/IEC/IEEE 26531:2023, 3.2]

3.2295

跳转 jump

从隐含的或声明的正在执行的计算机程序语句顺序离开。

3.2296

看板 kanban board

用于展示在研工作，协助识别瓶颈与过度分配情况，进而使团队能够优化工作流的可视化工具。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3.2297

KDM 实体 KDM entity

用于表示所关注系统中具有重要意义事物的元模型元素（以及相应的模型元素），关于该事物需要了解或掌握其信息。

[来源：ISO/IEC 19506:2012]

3.2298

KDM 实例 KDM instance

表示所关注系统的一个或多个视图的 KDM 模型元素的集合。

[来源：ISO/IEC 19506:2012]

3.2299

KDM 模型 KDM model

作为 KDM 视图容器的元模型元素（以及相应的模型元素）。

[来源：ISO/IEC 19506:2012]

3.2300

KDM 关系 KDM relationship

表示所关注系统元素之间某种语义关联的元模型元素（以及相应的模型元素）。

[来源：ISO/IEC 19506:2012]

3.2301

内核 kernel

集成了操作系统的基本功能的软件模块。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2503]

3.2302

内核常驻控制程序 kernel resident control program

封装了系统的一个或多个基本功能的软件模块。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2503]

3.2303

键迁移 key migration

在数据建模中，将父实体或通用实体的主键作为外键放置在其子实体或类别实体中。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.2505]

3.2304

关键实践 key practices

对关键过程域的有效实施和制度化贡献最大的基础设施和活动。

3.2305

关键过程域 key process area; KPA

一组相关的活动，当这些活动共同完成时，对建立过程能力至关重要的一组目标就能实现。

3. 2306

键式视图 key-style view

表示企业内部数据结构和语义的视图。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.2506]

3. 2307

关键字 keyword

在执行一个或多个测试用例时，用于描述一组特定行为的一个或多个词汇。

注 1：包括在测试、验证和特定行为建立的测试场景中 UI 交互启动测试方案。

注 2：关键字需至少使用一个动词命名。

注 3：复合关键字有时基于其他关键字构建。

注 4：关键字通常会带有参数，这样它们就能与不同的数据进行搭配使用。

[来源：ISO/IEC/IEEE 29119-5:2024, 3.5, 有修改]

3. 2308

关键字字典 keyword dictionary

关键字库 keyword library

由反映了使用的语言和抽象级别、将被用于编写测试用例的一组关键字构成的库。

[来源：GB/T 38634.5-2024, 3.1.4]

3. 2309

关键字执行代码 keyword execution code

由测试执行引擎执行关键字的实现。

[来源：GB/T 38634.5-2024, 3.1.7]

3. 2310

关键字测试用例 keyword test case

使用关键字及其关联参数的所需值（如适用）组成的序列来描述用例活动的测试用例。

[来源：GB/T 38634.5-2024, 3.1.8]

3. 2311

关键字驱动测试 keyword-driven testing

使用由关键字组成的测试用例进行的测试。

[来源：GB/T 38634.5-2024, 3.1.5]

3. 2312

关键字驱动测试框架 keyword-driven testing framework

由关键字驱动编辑器、分解器、数据序列化器、手动测试助手或桥接工具、数据和脚本库、关键字库等功能组件以及测试执行环境组成的测试框架。

[来源：GB/T 38634.5-2024, 3.1.6]

3. 2313

项目启动会议 kickoff meeting

项目伊始，由团队成员及其他关键干系人参与的会议，用于正式明确预期、达成共识并启动工作。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3. 2314

千次运算每秒 kilo-operations per second; KOPS

每秒千次运算。参见：百万次浮点运算每秒(MFLOPS)，百万次指令运算每秒(MIPS)。

注：计算机处理速度的一种度量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2520]

3.2315

知识域 knowledge area; KA

知识体系（BOK）中的子领域或相关主题分组。

[来源：ISO/IEC 24773-2:2024, 3.1]

3.2316

知识库 knowledge base; K-base

包含推理规则以及某一领域中人类经验与专业知识信息的数据库。

注：在自优化系统中，知识库还包含由以往问题的解决方案所形成的信息。

3.2317 [[来源：ISO/IEC 2382:2015, X.X]

标号 label

与网络图形或其对象之一相关的信息。

[来源：ISO/IEC 15909-2:2011, 4.1.5, 有修改]

3.2318

语言处理程序 language processor

一种软件工具，它完成这样一些功能，诸如处理指定的语言（如需求规格说明语言、设计语言或程序设计语言）所需的翻译、解释或其他任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2528]

3.2319

语言标准 language standard

一种描述语言特征的标准，此语言用于描述需求规格说明、设计或测试数据。

3.2320

最后责任时刻 last responsible moment

延迟决策，使团队有时间考量多种方案，直至继续拖延的成本将超过其收益的决策理念。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版]

3.2321

时延 latency

指令控制单元发出数据请求的时刻与数据传输开始时刻之间的时间间隔。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2532]

3.2322

潜在变量 latent variable

表示一维结构的变量。

[来源：ISO/IEC 33003:2015, 3.8]

3.2323

横向压缩 lateral compression

软件设计中的一种去模块化形式，它把两个或多个顺序执行的模块被合并为单个模块。参见：向下压缩(downward compression)，向上压缩(upward compression)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2534]

3.2324

层 layer

对软件系统进行功能划分后的结果，其中：。

—— 层组成了层次结构；
 —— 每个层级只有一个层；
 —— 在软件系统中，由可以直接进行数据交换的任意两层中软件提供的功能服务之间，有上下层等级依赖关系；

—— 在交换数据的软件系统中，在任意两层中的软件，只可阐释相同的数据部分。

[来源：GB/T 42452-2023, 3.15]

3.2325

布局 layout

源代码的物理组织形式，包括空格、分组、空行、对齐、缩进和括号的使用。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2536]

3.2326

引领式创业画布 lead startup canvas

一种单页模板，旨在高效、有效地与关键干系人传达商业计划。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版；ISO/IEC/IEEE FDIS 24765, 3.2544]

3.2327

前置时间 lead time

从客户提出需求到实际交付之间的时长。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版；IEEE_CS_SE_Vocabulary-2025]

3.2328

前置时间图 lead time chart

展示已完成工作项的平均前置时间随时间变化趋势的图表。

[来源：项目管理知识体系指南（PMBOK® 指南）-第七版；IEEE_CS_SE_Vocabulary-2025]

3.2329

前导判定 leading decision

在循环体之前执行的循环控制。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2548]

3.2330

左移 left-shift

在生存周期早期优先让相关利益相关方参与到质量活动、安全性、隐私、性能、验证和确认的活动中。

[来源：IEEE 2675-2021, 3.1]

3.2331

遗留硬件 legacy hardware

最初未采用原生信息结构所开发的硬件。

[来源：ISO/IEC 19770-6:2024, 3.1.5；IEEE_CS_SE_Vocabulary-2025]

3.2332

遗留软件 legacy software

最初创建时没有信息结构的软件。

[来源：ISO/IEC 19770-5:2015, 3.17]

3.2333

图例 legend

在架构描述语境中，对约定规则的非正式说明。

[来源: GB/T 45630-2025, 3. 19; IEEE_CS_SE_Vocabulary-2025]

3. 2334

经验教训 lessons learned

在项目过程中获得知识, 用于说明项目事件如何处理, 或未来宜如何处理, 以改进未来绩效。

[来源: 项目管理知识体系指南 (PMBOK®指南) 第七版]

3. 2335

经验教训登记册 lessons learned register

项目文件, 用于记录项目期间获得的知识, 以便在当前项目中使用, 并纳入经验教训知识库。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3. 2336

经验教训登记册 lessons learned register

经验教训知识库 lessons learned repository

用于记录在项目期间所获知识的项目文件, 以便在当前项目中使用并录入经验教训库。

[来源: ISO/IEC FDIS 24765, 3. 2558]

3. 2337

谅解函 letter of understanding; LOU

谅解函。

[来源: ISO/IEC/IEEE 41062:2024, 3. 2;; IEEE_CS_SE_Vocabulary-2025]

3. 2338

等级, 层级 level

视图覆盖范围与详细程度的指定。

注: 视图存在多个层级; 每个层级均应独立定义, 并通过所使用的建模构件加以区分。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 1. 100]

3. 2339

抽象层级 level of abstraction

系统描述中的特定详细级别的视图。

[来源: GB/T 45802-2025, 3. 1. 15]

3. 2340

文档等级 level of documentation

指明文档的范围、内容、格式以及质量。文档等级可根据项目成本、预期用途、作用范围、及其他因素进行选择。

3. 2341

支持型活动 level of effort; LOE

不产生明确最终产品的支持活动, 其特征通常是在由所支持活动决定的时间段内, 以均匀的速率开展工作。

示例: 客户联络、项目成本核算、项目管理。

注: 用于测量工作绩效的三种挣值管理活动类型之一。

[来源: ISO/IEC FDIS 24765, 3. 2561]

3. 2342

维修级别分析 Level of Repair Analysis; LORA

维修级别分析。

[来源: INCONE 系统工程手册 v4, APPENDIX B]

3. 2343

风险等级 level of risk

以后果和可能性的组合来描述的单个风险或组合风险的大小。

[来源：GB/T 23694-2024, 3. 3. 24]

3. 2344

服务等级 level of service

反映组织所产生的社会、政治、环境和经济成果的参数或参数组合。

[来源：ISO/IEC 19770-1:2017, 3. 31]

3. 2345

莱文斯坦距离 Levenshtein distance

两个字符序列之间的差异测度，基于将一个字符序列转换为另一个所需的最少单字符编辑（插入、删除或替换）操作次数。

[来源：ISO/IEC/IEEE 26531:2023, 3. 1. 21]

3. 2346

词典式决策规则 lexicography

对决策属性进行优先级排序的决策技术。

参见：优势法 (dominance)，满意法 (satisficing)。

[来源：ISO/IEC FDIS 24765, 3. 2565]

3. 2347

联络 liaison

对象集之间通过某种建立行为产生的关系。

[来源：ISO/IEC 10746-2:2009, 13. 2. 4]。

3. 2348

库 library

存放各类工作成果和有用信息项的场所，供人员阅读、借用或参考，并供机器访问和检索数据。

[来源：ISO/IEC/IEEE 42020:2019, 3. 10]

3. 2349

许可证 license

许可方与被许可方之间关于知识产权使用或转让的条款与条件的法律协议。

[来源：ISO/IEC FDIS 24765, 3. 2569]

3. 2350

许可证合规审计 license compliance audit

通过核对来自多个信息源的许可相关信息而进行的审计，例如将权益消耗量与权益权限进行比对。

[来源：ISO/IEC FDIS 24765, 3. 2570]

3. 2351

许可模式 license model

具有共同特征的许可证类别。

示例：场地许可证、原始设备制造商许可证、每计算机许可证。

[来源：ISO/IEC FDIS 24765, 3. 2571]

3. 2352

许可证发放标准 licensing standard

描述由官方或法定机构授予个人或组织从事或拥有特定事项的授权特征的标准。

[来源: ISO/IEC FDIS 24765, 3.2572]

3.2353

生存周期 life cycle

5. 系统、产品、服务、项目或其他人工实体从概念到退役的演变。

[来源: ISO/IEC/IEEE 12207:2017, 3.1.26; ISO/IEC/IEEE 15288:2023, 3.21; GB/T 22032-2021]

3.2354

生存周期评估 life cycle assessment; LCA

用于评估产品、过程或系统整体环境影响的工具。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.2355

生存周期成本 life cycle cost; LCC

产品开发、制造、测试、分销、运行、支持、培训及处置的总投资。

[来源: ISO/IEC FDIS 24765, 3.2575]

3.2356

生存周期管理 Life cycle management; LCM

生存周期管理。

[来源: INCONE 系统工程手册 v4, APPENDIX B]

3.2357

生存周期模型 life cycle model

与生存周期相关的过程和活动的框架, 可以组织成多个阶段, 也可作为交流和理解的通用参考。

[来源: GB/T 22032-2021, 4.1.24]

3.2358

生存周期过程 life cycle processes

一组相互关联或相互作用的活动, 其结果是开发或评估系统、软件或硬件产品。

注: 每项活动由若干任务组成。生存周期过程可相互重叠。出于验证与确认 (V&V) 的目的, 任何过程均需根据其验证与确认计划中规定的任务, 对其开发产品完成验证与确认后, 方可视为结束。

[来源: IEEE 1012-2024, 3.1]

3.2359

轻量级进程 lightweight process

单控制线程的进程, 一个任务。

[来源: ISO/IEC FDIS 24765, 3.2578]

3.2360

可能性 likelihood

事件发生的概率。

注: 指事件发生的机会, 无论是以客观的或主观的、定性或定量的方式来定义、度量或确定, 还是用一般词汇或数学术语来描述 (例如给定时间段内的概率或频率)。

[来源: ISO/IEC/IEEE 16085:2021, 3.2]

3.2361

限制 limit

对软件授权所授予权利或特权的约束。

[来源: ISO/IEC 19770-3:2016, 3.1.16]

3. 2362

受限项表 limited entry table

对所有条件和动作进行完整描述而未参照规则的判定表。

[来源: GB/T 15535-1995, 3.14]

3. 2363

代码行 line of code

编程语言中的基本语句、非注释、非空白的可交付源代码语句。

[来源: ISO/IEC FDIS 24765, 3.2582]

3. 2364

线性插值 linear interpolation

基于两个已知点之间直线上的值来近似函数在给定点的值。

[来源: ISO/IEC FDIS 24765, 3.2583]

3. 2365

连接, 链接 link

<超文本>从一个文档中某一部分到另一文档(或同一文档中其他部分)的引用。同义词: 超链接(hyperlink)。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.15]

<编译>计算机程序的一部分, 通常为单条指令或单个地址, 用于在程序的独立模块间传递控制权与参数。参见: 连接编辑程序。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.22]

3. 2366

连接加载程序 linking loader

将一或多个目标模块读入主存储器以备执行的计算机程序, 该程序通过解析各独立模块间的交叉引用生成单一加载模块, 并在某些情况下调整地址以反映代码被载入的存储位置。参见: 绝对加载程序(absolute loader)、重定位加载程序(relocating loader)、连接编辑程序(linkage editor)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2586]

3. 2367

[列]表, 清单 list

具有相同数据定义的数据项集合。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2587]

3. 2368

列表功能 list function

一个显示满足, 或者不一定满足特定选择标准的实体类型概要信息的在线功能。

[来源: GB/T 42588-2023, 3.1.45]

3. 2369

列表 listing

数据项、程序语句或其他信息的有序的显示或打印输出。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2589]

3. 2370

字面值 literal

源程序中某个项值的显式表示。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2590]

3. 2371

加载 load

将计算机指令或数据从外部存储器复制到内部存储器，或从内部存储器复制到寄存器。参见：加载程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2592]

3. 2372

加载映射表 load map

计算机生成的列表，用于标识内存驻留的全部(或部分)代码(或数据)的位置(或大小)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2593]

3. 2373

加载模块 load module

适合被加载到主存以供计算机执行的计算机程序或子程序；通常是连接编辑程序的输出。参见：目标模块(object module)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2594]

3. 2374

加载即执行 load-and-go

一种操作系统技术，在计算机程序的加载阶段与执行阶段之间没有停顿。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2596]

3. 2375

加载起点 loaded origin

计算机程序被加载到主存时的初始存储地址。参见：汇编起点(assembled origin)、偏移量(offset)、起始地址(starting address)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2597]

3. 2376

加载程序 loader

将程序或数据读入主存储器的程序。参见：引导程序(bootstrap)，连接编辑程序(linkage editor)。

注：类型包括绝对加载程序(absolute loader)、连接加载程序(linking loader)、重定位加载程序(relocating loader)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2598]

3. 2377

局域网 local area network; LAN

位于用户场所内、地理范围有限的计算机网络。

注：局域网内部通信不受外部法规约束；但跨局域网边界的通信可能受某种形式的法规管制。

[来源：ISO/IEC 2382:2015; IEEE_CS_SE_Vocabulary-2025]

3. 2378

局部符合逻辑 local coincidence logic; LCL

局部符合逻辑。

[来源：IEEE 1012-2024, 3.2; ISO/IEC/IEEE FDIS 24765, 3.2540]

3. 2379

局部微代码压缩 local compaction

在微程序设计中，指微代码不会被移动到其所在的单入口单出口顺序块的边界之外的压缩优化方式。参见：全局微代码压缩(global compaction)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2602]

3.2380

本地化定制 local customization

为适应本地使用进行修改的 FSM 方法, 与修改之前相比, 可能产生不同的功能规模。

[来源: ISO/IEC 14143-1:2007, 3.9]

3.2381

局部数据 local data

计算机程序中, 仅可被单个模块或一组嵌套模块访问的数据。

参见: 全局数据 (global data)。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.2604]

3.2382

本地化软件资产管理负责人 local SAM owner

组织内中被指定对特定部分的 SAM 负责的个人, 级别低于软件资产管理 (SAM) 负责人。

[来源: ISO/IEC 19770-5:2015, 3.21]

3.2383

局部变量 local variable

计算机程序中只能由单个模块或一组嵌套模块访问的变量。参见: 全局变量 (global variable)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2606]

3.2384

质量责任局部化 locality of quality responsibility

将特定质量相关需求或其分解的责任分配给特定过程实例。

[来源: ISO/IEC 30103:2015, 3.6]

3.2385

本地化 localization

为产品及其信息创建国家或特定区域版本, 提供给用户使用。

参见: 国际化 (internationalization)。

注: 本地化可与翻译过程分开执行。即使产品未进行本地化, 也可对文档进行本地化。

[来源: Sevocab 术语库]

3.2386

定位机制 location facility

允许客户端绑定对象向服务器端查询其是否接受携带对特定 (计算) 服务器对象调用的请求的一组服务原语。

注: 服务器端可以确认或拒绝该提议, 或者建议一个能够处理该请求的替代服务器端。

[来源: ISO/IEC 14752:2000, 3.3.9]

3.2387

空间位置 location in space

可以发生动作的任意大小的空间间隔。

[来源: ISO/IEC 10746-2:2009, 8.10]

3.2388

时间位置 location in time

可以发生动作的任意大小的时间间隔。

[来源: ISO/IEC 10746-2:2009, 8.11]

3. 2389

参考位置 location reference

索引或目录中标题或子标题后的指示符，显示该标题或子标题对应的文档部分。

[来源：ISO/IEC FDIS 24765, 3. 2611]

3. 2390

位置透明性 location transparency

在对界面标识和绑定时，屏蔽有关空间位置信息使用的分布透明性。

[来源：ISO/IEC 10746-3:2009, 4. 4. 1. 3]

3. 2391

锁 lock

独占编辑权限。

[来源：ISO/IEC FDIS 24765, 3. 2613]

3. 2392

锁定 lockout

一种计算机资源分配技术，对共享资源（尤其数据）进行保护，同一时刻仅允许一个设备或进程访问。

参见：死锁（deadlock）、信号量（semaphore）。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3. 2614]

3. 2393

日志 log

对过程或活动实施期间的特定事项进行记录、描述或说明的文件。前面常加修饰词，如问题、变更或假设等。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3. 2394

注销 log off; log out

结束会话。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2617]

3. 2395

登录 log on; log in

启动会话。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2618]

3. 2396

逻辑 logic

动作的逻辑限制（或约束条件）。

[来源：IEEE 1175.3-2004, 3. 18]

3. 2397

逻辑编程语言 logic programming language

用控制结构和受限谓词演算来表达程序的编程语言。

示例：PROLOG。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 2619]

3. 2398

逻辑内聚 logical cohesion

一种内聚类型,其中软件模块执行的任务在逻辑上具有相似的功能。参见:偶然内聚(coincidental cohesion),通信内聚(communicational cohesion),功能内聚(functional cohesion),过程内聚(procedural cohesion),顺序内聚(sequential cohesion),时间内聚(temporal cohesion)。

示例:处理不同类型的输入数据。

[来源:ISO/IEC/IEEE FDIS 24765, 3.2620]

3.2399

逻辑文件 logical file

从用户视角所看到的一个永久数据的逻辑组。逻辑文件可以是内部逻辑文件或者外部逻辑文件。

参见:数据功能。

[来源:GB/T 42588-2023, 1.838; ISO/IEC/IEEE 32430:2025, 3.1.22]

3.2400

逻辑接口 logical interface

两个或多个系统元素或软件系统元素之间的输入输出流,以及确定该流的功能。

[来源:ISO/IEC/IEEE 24748-6:2023, 3.1.5; IEEE_CS_SE_Vocabulary-2025]

3.2401

逻辑布局 logical layout

为输出产品定义的用户需要的一组数据元素类型的集合及其逻辑结构,区别于物理实现的考虑。逻辑布局不涉及到数据在屏幕、记录或者其他介质上的物理展现方式。

[来源:GB/T 42588-2023, 3.1.47]

3.2402

逻辑记录 logical record

在主程序单次迭代中处理的一组数据。

示例:数据库表中的一行。

注:它可以是单个物理记录(或多个记录)的一部分(或全部)。

[来源:ISO/IEC/IEEE FDIS 24765, 3.2623]

3.2403

逻辑关系 logical relationship dependency

依赖

两个活动之间,或一个活动与一个里程碑之间的依赖关系。

参见:紧前关系(precedence relationship)。

[来源:《项目管理知识体系指南(PMBOK® 指南)第六版》]

3.2404

逻辑源语句 logical source statement; LSS

与其呈现的物理格式(代码行)无关的软件指令。

参见:物理源语句。

[来源:ISO/IEC FDIS 24765, 3.2624]

3.2405

逻辑轨迹 logical trace

仅记录了分支或跳转指令的执行轨迹。

参见:执行轨迹(execution trace),回溯轨迹(retrospective trace),子程序轨迹(subroutine trace),符号轨迹(symbolic trace),变量轨迹(变量轨迹)。

需要补充索引号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2625]

3.2406

逻辑事务 logical transaction

分析的基本功能部件,在业务上对用户有意义的最小完整信息处理单元,通过真实世界与用户相关的一个事件或者信息请求来触发,由输入部件、处理部件和输出部件组成。它是自包含的,并能使被计数的应用程序处于一致状态。

[来源: GB/T 42566-2023 系统与软件工程 功能规模测量 MkII 功能点分析方法, 3.21]

3.2407

逻辑型 logical type

一种数据类型,它的成员只能取逻辑值(通常为 TRUE 和 FALSE)且只能由逻辑运算符(如 AND、OR 和 NOT)对其进行运算。

参见: 字符型(character type), 枚举型(enumeration type), 整型(integer type), 实型(real type)。

需要补充索引号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2627]

3.2408

循环 loop

一计算机程序语句序列,会在某一给定条件满足前或条件为真时反复执行。

参见: 循环体(loop body)、循环控制(loop control)、UNTIL 语句、WHILE 语句。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2628]

3.2409

循环断言 loop assertion

逻辑表达式,规定在程序循环中某一特定点每次被执行时必须满足的一个或多个条件。

参见: 输入断言(input assertion), 输出断言(output assertion), 归纳断言法(inductive assertion method)。

需要补充索引号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2629]

3.2410

循环体 loop body

循环中实现其主要功能的部分。

参见: 循环控制(loop control)。

需要补充索引号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2630]

3.2411

循环控制 loop control

循环中用于判定是否退出循环的部分。

参见: 循环体(loop body), 前导判定(leading decision), 尾随判定(trailing decision)。

需要补充索引号。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2631]

3.2412

循环控制变量 loop control variable

用于判定是否从循环退出的程序变量。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2632]

3.2413

损耗 **lossage**

故障或减损对存储数据产生的有害影响。

[来源: IEEE 982:2024, 3.1; IEEE_CS_SE_Vocabulary-2025]

3.2414

低层级 **low level**

具体的; 详细的。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2637]

3.2415

低层类 **lowclass**

若一个实例属于类 S 且不属于 S 的任何子类, 则 S 是该实例的低类。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2641]

3.2416

低层级设计 **low-level design**

在更通用设计的指导下, 于个体例程或有时在类级别进行设计的过程。

参见: 详细设计。

[来源: ISO/IEC FDIS 24765, 3.2638]

3.2417

低层级关键字 **low-level keyword**

只涵盖一个或极少的简单活动, 且组成中不含其他关键字的关键字。

[来源: GB/T 38634.5-2024]

3.2418

机器学习 **machine learning; ML**

利用计算技术使系统从数据或经验中学习的过程。

[来源: ISO/IEC TR 29119-11:2020, 3.1.43]

通过计算技术优化模型参数的过程, 使模型行为能够反映数据或经验。

[来源: IEEE 1012-2024, 3.1]

3.2419

机器可读 **machine readable**

以能自动输入至计算机的格式表示的数据。

[来源: IEEE 2675-2021, 3.1]

3.2420

机器翻译 **machine translation; MT**

使用计算机应用程序将文本从一种自然语言自动翻译成另一种自然语言。

[来源: GB/T 40036-2021, 3.1.1]

3.2421

机器可读的 **machine-readable**

可通过计算机自动生成并输出到计算机中的数据形式。

[来源: IEEE 2675-2021, 3.1]

3.2422

宏 **macro**

在软件工程中，一个预先定义好的计算机指令序列。

注：它在汇编或编译期间把该指令序列插入到程序中每一处出现相应宏指令的地方。

参见：宏指令、宏生成器和开式子例程。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.2660]

3.2423

宏定义 macro definition

在软件工程中，一个预先定义好的计算机指令序列，通常在汇编或编译期间把该指令序列插入到程序中每一处出现相应宏指令的地方。

参见：宏指令、宏生成器和开放式子程序

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2424

宏生成程序 macro generating program

一个例程，通常是汇编程序或编译程序的一部分，它将源程序中的每个宏指令替换为其代表的预定义指令序列。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2425

宏库 macro library

可由宏生存器使用的宏的集合。参见系统库（system library）。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2426

宏汇编程序 macroassembler

包括或执行宏生成器功能的汇编程序。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2427

宏生成器 macrogenerator

一个例程，常常是汇编程序或编译程序的一部分，在源程序中，它用宏指令表示的预定义指令序列，替换每条宏指令。

参见：宏指令、宏

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.2663]

3.2428

宏指令 macroinstruction

源代码指令，它将用同一源语言书写的预先定义的指令序列在汇编或编译期间代替。宏指令也可以为将要代替它的指令中的参数指定其值。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2429

宏处理程序 macroprocessor

在某些汇编程序和编辑程序中的例程或例程集，它用以支持宏的定义和使用。

3.2430

宏编程 macroprogramming

编写用宏和宏指令构成的计算机程序。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2431

主程序 main program

由计算机的操作系统调用的软件部件，它通常调用其他软件部件。参见：例程（routine）、子程序（subprogram）。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2432

主进度计划 main schedule

包含主要里程碑、可交付成果和工作分解结构要素（活动）的高层次进度计划。

同义词：顶级进度计划（top-level schedule），摘要进度计划（summary schedule），主进度计划（已废弃）（master schedule（deprecated））。

[来源：ISO/IEC/IEEE FDIS 24765; ISO/IEC/IEEE FDIS 24765, 3.2673]

3. 2433

维护知识库活动 maintain repository activity; MRa

对标杆管理知识库产品和已发布的标杆数据进行管理的活动。

注1：标杆管理知识库产品包含标杆管理数据、标杆管理模型和标杆管理结果。

注2：已发布的标杆数据是指已获批准发布的标杆管理结果。

[来源：GB/T 41866.4-2022, 4.1.3]

3. 2434

可维护性 maintainability

软件系统或部件能修改以排除故障、改进性能或其他属性或适应变更了的环境的容易程度。

参见：2.594 条 可扩展性（extendability）和 2.634 条 灵活性（flexibility）。

硬件系统或部件能从执行它所要求的功能的状态得到或重新存放回去的容易程度。

3. 2435

维护 maintenance

在交付以后，修改软件系统或部件以排除故障、改进性能或其他属性或适应变更了的环境的过程。

注：在可信性范畴内，维护是为使产品保持或恢复到能执行规定功能的状态，所采取的全部技术与管理活动的总和。

参见：软件维护、适应性维护、改正性维护、完善性维护、预防性维护

[来源：ISO/IEC 25051:2014, 4.1.9]

3. 2436

维护手册 maintenance manual

软件工程项目可交付文档，它为系统维护人员（而非用户）提供系统维护所需信息的文档。参见：操作手册（operator manual）、用户手册（user manual）。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2437

维护阶段 maintenance phase

软件生存周期中的一个阶段，在此阶段软件产品在规定的运行环境中进行使用、监控，以得到满意的性能，必要时要进行修改，以改正问题或对变化了的需求作出响应。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2438

维护计划 maintenance plan

维护软件产品所使用的说明管理方法和技术途径的文档。典型的维护计划内容包括：工具、资源、设施及日程。

3. 2439

维护程序 Maintenance procedure

包括一组维护操作的维护程序，其使用特定的维护技术，通过特定的维护促成因素执行。

[来源：INCOSE，附录 B]

3. 2440

维护计划 维护基础设施 maintenance program maintenance infrastructure

用于实施维护计划的组织结构、职责、程序、过程和资源。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2441

制造或购买分析 make or buy analysis

收集和整理有关产品需求的数据，根据可用的替代方案进行分析的过程，包括产品的采购或内部制造。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2688]

3. 2442

受到管理和控制 managed and controlled

标识和定义软件工作产品的过程，这些软件工作产品不是基线的一部分，因此不一定要置于配置管理之下，但为使项目有纪律地开展又必须受控。“受到管理和控制”是指在给定时间（过去或现在）使用的工作产品的版本均是可查知（即，版本控制），而且以受控的方式进行更改（即，更改控制）。

3. 2443

管理过程 management process

为确保软件工程过程按照组织的政策、目标和标准执行而进行的活动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2699]

3. 2444

管理储备 management reserve

为管理控制目的，在绩效测量基准之外预留项目预算或进度额度，用于应对项目范围未预见的工作。

注：管理储备是为项目范围内不可预见工作而预留的预算，不计入绩效测量基准。

[来源：项目管理知识体系指南（PMBOK®指南）—第七版]

3. 2445

管理评审 management review

由管理层或代表管理层执行的软件采购、供应、开发、运作或维护过程的系统性评估，旨在监控进展、确定计划和进度的状态、确认需求及其系统分配情况，或评估为实现适用性而采用的管理方法的有效性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2701]

3. 2446

管理技能 management skills

计划、组织、指导和控制个人或团队，以实现特定目标的能力。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2702]

3. 2447

管理系统 management system

一组相互关联或相互作用的要素，用于制定政策和目标，并实现这些目标。

注：管理系统可以涉及单一学科或多个学科。系统要素包括组织结构、角色与职责、规划和运营。管理系统的范围可以包括组织的全部内容、组织特定和已识别的职能、组织特定和已识别的部门，或跨多个组织的一个或多个

职能。

[来源：ISO/IEC 19770-1:2017, 3.33]

3.2448

管理独立性 managerial independence

软件质量保证（SQA）中的情况，其中 SQA 工作的责任由独立于开发组织和项目管理组织的机构承担。

[来源：IEEE 730-2014, 3.2]

3.2449

管理角色 managing role

正在执行管理操作的对象的视图。

[来源：ISO/IEC 10746-2:2009, 14.5]

3.2450

强制的 mandatory

用于指定完全映射的语法关键字。

参见：可选、完全。

注：强制性要求使用“shall”表示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2706]

3.2451

强制性依赖 mandatory dependency

合同上要求的或工作本质中固有的关系。

[来源：项目管理知识体系指南（PMBOK® 指南）第七版]

3.2452

强制性非标识关系 mandatory nonidentifying relationship

非标识关系，其中子实体的一个实例与父实体的一个实例相关。

参见：可选非标识关系、非标识关系 [关键样式]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2708]

3.2453

手动测试 manual testing

通过人工方式将信息输入测试项并验证结果来执行测试。

注：手动测试不使用工具、机器人或其他测试执行引擎。

[来源：ISO/IEC/IEEE 29119-5:2024, 3.15]

3.2454

制造 manufacture

在软件工程中，复制软件至磁盘、芯片或其他设备以发布至客户或用户的过程。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2455

制造阶段 manufacturing phase

软件生存周期中软件产品的基本版本适应一组特定的操作环境并分发给客户群的时间段。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2456

多分类代数 many-sorted algebra

由一组集合和一组以这些集合作为定义域和上定义域的函数组成的数学结构。

[来源: ISO/IEC 15909-1:2019, 3.14]

3.2457

多对多关系 many-to-many relationship

两个状态类（不一定不同）之间的关系，其中一个类的每个实例可以与第二个类的任意数量的实例关联（可能为零），并且第二个类的每个实例可以与第一个类的任意数量的实例相关联（可能为零）。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2713]

3.2458

映像程序 map program

一种软件工具，通常是编译程序或汇编程序的组成部分，它生成装入映像。

[来源: ISO/IEC IEEE FDIS 24765, X.X]

3.2459

映射 mapping

在两个事物之间分配对应关系，以有序对的形式表示。

2. 根据选定的软件生存周期模型（SLCM）建立活动序列。

参见：实例、调用、迭代、软件生存周期模型（SLCM）。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2716]

3.2460

映射完整性 mapping completeness

映射是完整（完全映射）还是不完整（部分映射）的标识。

参见：部分、全部。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2717]

3.2461

映射文件 mapping document

与标准和现有行业实践相关的文件。

[来源: ISO/IEC 19770-8:2020, 3.1]

3.2462

标记 marking

产品、标签或包装上用于识别其类型的符号、象形图、警告、标识或铭文，还可包括简短的文本信息。

[来源: IEC/IEEE 82079-1:2019, 3.24]

3.2463

（网络的）标记 marking (of a net)

网络所有位置的位置标记集。

[来源: ISO/IEC 15909-1:2019, 3.15]

3.2464

地点标记 marking of a place

与（“驻留在”）地点相关联的多组标记。

[来源: ISO/IEC 15909-1:2019, 3.16]

3.2465

标记语言 markup language

定义和描述不同类型电子文档结构的方法。

[来源: ISO/IEC/IEEE 26511:2018, 3.1.20]

3. 2466

掩码 mask

用来与一未知的数据项逻辑地组合的位或字符的模式以得到或抑制数据项的某些部分。例如，位串”00000011”当与一八位数据项逻辑与时，得到数据项的最低两位而在所有其他位的位置为0的结果。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2467

掩膜只读存储器 mask ROM

在制造过程中对电路进行编程的只读存储单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2725]

3. 2468

原料 material

组织在任务中所使用的物品的集合。

示例：设备、器械、工具、机械、装置及物资。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2726]

3. 2469

矩阵图 matrix diagrams

一种质量管理和控制工具，用于在矩阵中创建的组织结构内执行数据分析。矩阵图旨在显示构成矩阵的行和列之间存在的因素、原因和目标之间的关系强度。

[来源：项目管理知识体系指南（PMBOK®指南）第五版]

3. 2470

成熟性 maturity

系统、产品或组件在正常运行下满足可靠性需求的程度。

注：成熟度的概念可应用于质量特性，以表明它们在正常操作下满足所需需求的程度。

[来源：GB/T 25000.10-2016, 4.3.2.5.1]

3. 2471

成熟度等级 maturity level

在实现成熟软件过程的过程中，组织的不断进化但又相对稳定的软件过程的一种定义的状态。

[来源：ISO/IEC 33001:2015, 3.4.1]

3. 2472

成熟度模型 maturity model

源自一个或多个指定过程评估模型的模型，该模型识别与组织过程成熟度的指定规模相关的过程集。

[来源：ISO/IEC 33001:2015, 3.3.7]

3. 2473

成熟度提问单 maturity questionnaire

一组关于软件过程的提问，抽样检查能力成熟度模型的每个关键过程域中关键实践的实施情况。它是评价组织的软件过程能力或项目的软件过程性能的一种手段。

3. 2474

大中取大准则 maximax rule

在不确定性决策中，假定最有利的自然状态将会发生，从所有最优收益中选择收益最大的方案。

参见：赫维茨准则、小中取大准则、最小最大后悔值准则。

[来源: ISO/IEC/IEEE 24765:20125, 3. 2729]

3. 2475

小中取大准则 maximin rule

在不确定性决策中, 假定最不利的自然状态将会发生, 从所有最差收益中选择收益最优的方案。

注: 这是不确定性决策方法中最悲观的一种。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 2730]

3. 2476

均值 mean

一组数字的算术平均值。

注: 对于 n 个观测值 $x_1, x_2, \dots, x_n$, 其算术平均值 $\bar{x} = (x_1 + x_2 + \dots + x_n)/n$ 。

[来源: IEEE 982:2024, 3. 1]

3. 2477

平均执行时间 mean execution time

在评级区间内提交的 j 类任务的所有执行时间的平均值。

[来源: ISO/IEC 14756:1999, 4. 8]

3. 2478

平均执行时间评级值 mean execution time rating value

(对应第 j 类任务的) 平均执行时间参考值与实测平均执行时间的商。

[来源: ISO/IEC 14756:1999, 4. 9]

3. 2479

平均执行时间参考值 mean execution time reference value

模拟用户可接受的最大平均执行时间。

[来源: ISO/IEC 14756:1999, 4. 1]

3. 2480

平均失效时间间隔 mean time between failures; MTBF

在系统或部件中, 在连续的失效之间预料的或观察到的时间。

参见: 开机时间 (up time)。

[来源: ISO/IEC/IEEE 24765:2017, X. X]

3. 2481

平均失效时间间隔 (MTBF) mean time between failures (MTBF); MTBF

在系统或部件中, 在连续的失效之间预料的或观察到的时间。

[来源: ISO/IEC IEEE FDIS 24765, X. X]

3. 2482

平均故障间隔时间 mean time between failures, ; MTBF; MTBF

在系统或部件中, 在连续的失效之间预料的或观察到的时间。

参见: 开机时间 (up time)。

[来源: GB/T 11457-2006; ISO/IEC/IEEE 24765:2017]

3. 2483

平均检测时间 mean time to detection; MTTD

从故障、事件或异常发生到被检测到的平均时间。

[来源: ISO/TR 12489:2013, 3. 1. 38]

3.2484

平均修复时间 mean time to repair; MTTR

修复系统或部件使它恢复正常工作所要求的预料的或观察到的时间。

参见：关闭时间(down time)。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2485

平均修复时间 (MTTR) mean time to repair (MTTR); MTTR

修复系统或部件使它恢复正常工作所要求的预料的或观察到的时间。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3.2486

有意义的 meaningful

用户可识别并满足功能用户需求。

[来源：GB/T 42449-2023, 3.41]

3.2487

测度 (名) measure (noun)

作为测量结果被赋予值的变量。

注：该术语的复数“measures”被用来指代基本测度、派生测度和指标的统称。

[来源：ISO/IEC 14598-1:1999, 有修改]

3.2488

有效性测度 measure of effectiveness; MOE

待评价的运行目标在预期运行环境中、指定条件下的满足程度相关的运行测度。

参见：性能测度 (measure of performance)。

注：定义决策者在系统满足运行预期方面有效性所需的信息。

[来源：GB/T 45615-2025; ISO/IEC/IEEE 24748-4:2016, 4.7]

3.2489

性能测度 measure of performance; MOP

提供关键性能需求以满足有效性测度的工程参数。

参见：有效性测度 (measure of effectiveness)。

注：MOP 通常表征与系统运行相关的物理或功能属性。

[来源：ISO/IEC/IEEE 24748-4:2016, 4.8;GB/T 45615-2025]

3.2490

测量 measurement

一组操作，其目的是确定某个测度的值。

注1：改写 1993 年版计量学基本与通用术语国际词汇。

注2：测量可包括分配一个定性的类别，例如源程序的语言 (Ada, C, COBOL 等)。

[来源：ISO/IEC/IEEE 15939:2017, 3.17]

3.2491

测量分析师 measurement analyst

负责策划、执行、评估和改进测量工作的个人或组织。

[来源：ISO/IEC/IEEE 15939:2017, 3.18]

3.2492

测量经验库 measurement experience base

包含信息产品评估、测量过程以及测量过程中所获得的经验教训的数据存储。

[来源：ISO/IEC/IEEE 15939:2017, 3.29]

3.2493

测量函数 measurement function

为组合两个或多个基本测度而执行的算法或计算。

[来源：GB/T 25000.40-2018, 4.41]

3.2494

测量管理员 measurement librarian

负责管理测量数据存储的个人或组织。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2758]

3.2495

测量方法 measurement method

操作的逻辑序列，用以按规定标度量化的属性。

注：改写 1993 年版计量学基本与通用术语国际词汇。

[来源：GB/T 25000.40-2018, 4.42]

3.2496

测量模型 measurement model

潜变量与其（多项目）测量指标之间的隐式或显式关系。

[来源：ISO/IEC 33001:2015, 3.10]

3.2497

度量绩效域 measurement performance domain

绩效域，涉及评估项目绩效并采取适当措施以保持可接受绩效的相关活动和职能。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.2498

测量规程 measurement procedure

一组具体操作的描述，可按给定方法执行特定的测量。

注：改写 1993 年版计量学基本与通用术语国际词汇。

[来源：GB/T 25000.40-2018, 4.43]

3.2499

测量过程 measurement process

在一个完整项目或组织的测量机构中确立、策划、执行和评价软件测量的过程。

[来源：GB/T 25000.40-2018, 4.44]

3.2500

测量过程负责人 measurement process owner

负责测量过程的个人或组织。

[来源：ISO/IEC/IEEE 15939:2017, 3.24]

3.2501

测量储藏库 Measurement repository

一种能够以及时、完整、有效，并在必要时保密的方式，保障所有相关测量项 / 度量项可用、可传递的知识库（仓库）。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2502

度量来源 measurement source

用于质量评估的质量度量工件集。

[来源: ISO/IEC 25040:2024, 3. 5]

3. 2503

测量发起人 measurement sponsor

授权并支持建立测量过程的个人或组织。

[来源: ISO/IEC/IEEE 15939:2017, 3. 25]

3. 2504

测量标准 measurement standard

描述评价过程或产品特征的标准。

[来源: ISO/IEC/IEEE FDIS 24765 3. 2767]

3. 2505

测量用户 measurement user

使用信息产品的个人或组织。

[来源: ISO/IEC/IEEE 15939:2017, 3. 260]

3. 2506

测量仪器 measuring instrument

单独或与辅助设备结合用于进行测量的设备。

[来源: ISO/IEC TR 14143-3:2003, 3. 6]

3. 2507

机制回环 mechanism loopback

一个函数的输出回环作为同一图表中另一个函数的机制。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 2770]

3. 2508

成员产品 member product

应用 application

属于产品线的产品。

[来源: GB/T 41865-2022, 3. 15]

3. 2509

内存 memory

处理单元中的可寻址存储空间以及用于执行指令的所有其他内部存储。

3. 2510 [[来源: ISO/IEC 2382:2015, X. X]

内存容量 memory capacity

在给定的计算机内存中能保持的项的最大数，它通常用字或字节测量。参见：通道容量（channel capacity）和存储容量（storage capacity）。

[来源: ISO/IEC IEEE FDIS 24765, X. X]

3. 2511

内存映像 memory map

显示程序和数据存放在计算机的内存的何处的图。

3. 2512

菜单 menu

显示在屏幕画面上的，展示可用功能的列表，在这个列表上可以进行选择。

[来源：GB/T 42588-2023，3.1.49]

3.2513

菜单结构 `menu structure`

通过一系列相关的菜单和屏幕画面实现应用程序与用户之间的对话。

[来源：GB/T 42588-2023，3.1.50]

3.2514

合并 `merge`

合并对同一文件的不同更改。

注：许多系统采用乐观策略，即合并所有不存在冲突的代码行。

[来源：ISO/IEC/IEEE FDIS 24765 3.2780]

3.2515

从当前分支合并 `merge from current`

将当前分支的变更合并到稳定分支。

注：为了避免稳定分支出现破坏性变更，代码变更通常先被引入到当前（开发）分支，经测试后再合并回稳定分支。

[来源：ISO/IEC/IEEE FDIS 24765 3.2781]

3.2516

消息 `message`

从一个对象发送到另一个对象的通信。参见：请求（request）。条目注释：消息既包括履行职责的请求，也包括简单的信息性通信。

[来源：ISO/IEC IEEE FDIS 24765，X.X]

3.2517

消息摘要 5 `message digest 5; MD5`

消息摘要算法 5 `Message-Digest algorithm 5`

一种通过输入数据（可以是任意长度的消息）生成 128 位消息摘要，从而验证数据完整性的算法，该算法声称对于该特定数据来说是唯一的，就像指纹对于特定个人一样。

[来源：ISO/IEC 19770-5:2015，3.22]

3.2518

元 `meta`

概念的前缀，用于说明该概念的定义信息。

注：具体来说，用于指示对象在三个模型层中的位置。

[来源：ISO/IEC 15474-1:2002，4.2]

3.2519

元属性 `meta-attribute`

元实体或元关系特征的定义。

注：元属性的实例作为数据值出现在模型中。

[来源：ISO/IEC 15474-1:2002，4.2]

3.2520

元类 `metaclass`

一种其实例为类的类。其典型用途是构造元模型。

3.2521

元编译程序 `metacompiler`

用来构造编译程序的一部分或全部的翻译程序或解释程序。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2522

元数据 metadata

描述其他数据的数据。

[来源：GB/T 25000.12-2017, 4.13]

3. 2523

元实体 meta-entity

CDIF 模型中出现的数据对象类型的定义。

注：元实体表示一组零个或多个元属性，存储在一起以表示在模型中具有实例的事物、事件或概念。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2524

元语言 metalanguage

用于规定语言的某些方面或所有方面的语言。参见：分层的语言（stratified language）和非分层的语言（unstratified language）。例如，巴科斯-诺尔范式。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2525

元元属性 meta-meta-attribute

元元实体或元元关系的特征的定义。

注：元元属性的实例在元模型中是元数据的值。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2526

元元实体 meta-meta-entity

元实体、元关系、元属性或主题领域的行为和结构的定义。

注：即用于描述模型中信息的元对象定义的定义。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2527

元元模型 meta-metamodel

一种用于定义元模型的模型。元元模型与元模型之间的关系，类同于元模型与模型之间的关系。

3. 2528

元元关系 meta-meta-relationship

CDIF 元模型中数据对象的类型的定义。

注：具体地，元元关系表示元元实体实例之间关系的定义。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2529

元模型 metamodel

用于定义方法的概念、关系和规则。

[来源：ISO/IEC/IEEE 24641:2023, 3.1.12]

3. 2530

元模型元素 metamodel element

元模型中的一个元素，各类模型元素均由其实例化生成。

[来源：ISO/IEC/IEEE FDIS 24765 3.2795]

3. 2531

变体关系 metamorphic relation

描述从源测试用例到后续测试用例测试输入的变化如何影响（或不影响）从源测试用例到后续测试用例预期输出的变化。

[来源：ISO/IEC TR 29119-11:2020, 3.1.44]

3. 2532

元对象 meta-object

元实体、元关系和元属性的通用术语。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2533

元对象设施 meta-object facility

对象管理组针对任意类型系统的类型信息存储库的规范。

[来源：ISO/IEC 14769:2001, 3.3.1]

3. 2534

元对象设施 MOF meta-object facility MOF; MOF

对象管理组织针对任意类型系统的类型信息存储库的规范。

[来源：ISO/IEC 14769:2001, 3.3.1]

3. 2535

元关系 meta-relationship

CDIF 模型中出现的一种数据对象类型的定义。

注：具体而言，元关系表示元实体（meta-entities）之间关系的定义，且该关系在模型中存在实例。元关系还可定义一组零个或多个元属性（meta-attributes），这些属性共同存储以表示元实体之间关系的特征。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2536

方法 method

执行服务所需的代码。

[来源：ISO/IEC IEEE FDIS 24765, X.X]

3. 2537

方法工程师 method engineer

负责设计、构建、扩展和维护方法论的人员。

注：方法工程师通过基于元模型的生成过程创建方法论。

[来源：ISO/IEC 24744:2014, 3.10]

3. 2538

方法标准 method standard

描述用在产品工程中或执行服务的顺序进程或过程的特征的标准。

[来源：ISO/IEC/IEEE FDIS 24765 3.2800]

3. 2539

方法论 methodology

某学科从业者所采用的方法体系，涵盖实践、技术、程序及规则。

[来源：项目管理知识体系指南（PMBOK®指南）——第七版]

3. 2540

方法论元素 methodology element

方法论的简单组件。

注：方法论元素通常包括：在应用该方法论时，可或必须使用的任务、活动、技术、模型、文档、语言及/或符号的规范说明。方法论要素相互关联，构成抽象概念网络。典型的方法论要素包括需求捕获、方法代码编写（任务类）、需求工程、高层建模（活动类）、伪代码、依赖关系图（符号类）、类、属性（模型构件类）、类模型、类图、需求规格说明书（工作产物类）。

[来源：ISO/IEC 24744:2014, 3.6]

3. 2541

计量、度量[体制] metric

系统、部件或进程占有给定的属性的程度的数量的测量。参见：2.1300 条 质量度量（quality metric）。

定义的测量方法和测量标度。

注1：度量[体制]可以是内部的或外部的，可以是直接的或间接的。

注2：度量[体制]包括把定性数据进行分类的方法。

[来源：GB/T 11457-2006, 2.942]

3. 2542

微码汇编器 micro code assembler

将微程序从符号形式转换为二进制形式的计算机程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2806]

3. 2543

微控制器单元 microcontroller unit; MCU; single chip microcomputer

在单个芯片上具有应用控制功能的单元。

注：它包含处理器、RAM、ROM、时钟、寄存器和 I/O 控制单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2810]

3. 2544

微型发生器 microgenerator

宏生成程序。

通常作为汇编器或编译器的一部分，用于将源程序中的每条宏指令替换为该宏指令所代表的预定义指令序列的例程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2663]

3. 2545

微指令 microinstruction

在微程序编程中，规定执行机器语言指令所需的一个或多个基本操作的指令。

注：类型包括对角微指令、横向微指令、纵向微指令。

[来源：ISO/IEC 24765 FDIS, 3.2811]

3. 2546

微操作 microoperation

在微程序设计中，执行机器语言指令所需的基本操作之一。

[来源：ISO/IEC 24765 FDIS, 3.2812]

3. 2547

微程序 microprogram

指令序列，称为微指令，指定执行机器语言指令所需的基本操作。

[来源：IEEE 1012-2024]

3. 2548

可编微程序的计算机 microprogrammable computer

微程序能由用户创建或替换的微程序的计算机。

[来源: ISO/IEC 24765 FDIS, 3.2815]

3. 2549

微程序的计算机 microprogrammed computer

机器语言指令由微程序而不是硬连线逻辑执行的计算机。

[来源: ISO/IEC 24765 FDIS, 3.2816]

3. 2550

微程序设计 microprogramming

设计和实现计算机的控制逻辑的过程，它是通过标识为实现每条机器语言指令所需的基本操作并把这些操作表示为在称为控制存储的特殊存储器中的指令序列来实现的。参见：微码（microcode）、微操作（microoperation）和微程序（microprogram）。

注：此方法是执行每条机器语言指令所需控制信号硬连线的一种替代方案。相关技术包括位操纵、压缩、剩余控制、单级编码和两级编码。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3. 2551

中间件 middleware

操作系统和软件应用程序之间的软件层。

[来源: ISO/IEC 24765 FDIS, 3.2819]

3. 2552

可迁移性 migratability

指对象在被使用过程中，通过替换其某一参考点来改变配置的能力。

参见：可移植性。

[来源: ISO/IEC 10746-2:2009, 15.4.2]

3. 2553

迁移属性 migrated attribute

子实体的外键属性。

注：[键样式]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2821]

3. 2554

迁移 migration

将集群移动到不同的容器中。

[来源: ISO/IEC 10746-3:2009, 8.1.27]

3. 2555

迁移透明度 migration transparency

分布透明性，系统屏蔽对象感知其位置变更的能力。

注：迁移通常用于实现负载均衡和降低延迟。

[来源: ISO/IEC 10746-3:2009, 4.4.1.4]

3. 2556

里程碑清单 milestone list

里程碑是项目中的重要时点或事件，里程碑清单列出了所有项目里程碑，并指明每个里程碑是强制

性的（如合同要求的）还是选择性的（如根据历史信息确定的）。里程碑的持续时间为零，因为它们代表的是一个重要时间点或事件。

[来源：项目管理知识体系指南（PMBOK®指南）——第六版]

3. 2557

里程碑回顾 milestone review

对工作产品及其支持证据进行正式评审，以确定其是否可接受用于下一开发阶段或交付。

注：此类评审的要求通常在项目计划中明确规定。

[来源：ISO/IEC 20246:2017, 3.10]

3. 2558

里程碑进度计划 milestone schedule

一种展示带有计划日期的里程碑的进度表类型。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3. 2559

思维导图 Mind-Mapping

把从头脑风暴中获得的创意整合成一张图的技术，用以反映创意之间的共性与差异，激发新创意。

[来源：项目管理知识体系指南（PMBOK®指南）第六版]

3. 2560

极简主义 minimalism

所使用的信息包含关键信息，及为保证完整性所必需的最少其他信息。

[来源：IEC/IEEE 82079-1:2019, 3.25]

3. 2561

最小最大后悔值准则 minimax regret rule

在不确定性决策中，先计算各方案在不同自然状态下的后悔值，找出每一方案的最大后悔值，再从中选择后悔值最小的方案作为决策方案。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2831]

3. 2562

最低期望收益率 minimum attractive rate of return; MARR

组织考虑投资的最低回报率；商业决策分析中使用的利率。

参见：机会成本。

注：反映组织有信心通过典型活动实现的回报率。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2832]

3. 2563

最低吸引力收益率 minimum attractive rate of return; MARR

最低吸引力收益率。

组织愿意考虑投资的最低收益率；商业决策分析中使用的利率。

另见机会成本。

注：反映了在既定风险水平下的最低可接受收益率。

[来源：ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.2832]

(1) 最低吸引力收益率。

[来源：ISO/IEC/IEEE 24765:2017, X.X]。

3. 2564

最小延迟编程 minimum delay programming

一种编程技术，它选择计算机指令和数据的存储单元，以使访问时间最小。

[来源：ISO/IEC 24765 FDIS, 3.2833]

3.2565

最低限度任务 minimum tasks

为使指派了完整性等级的系统、软件或硬件得以完成验证与确认，所必须开展的验证与确认任务。

[来源：IEEE 1012-2024, 3.1]

3.2566

最小可行产品 minimum viable product; MVP

具备足够功能和需求以满足早期客户并为后续开发提供反馈的工作产品版本。

[来源：IEEE 7052-2023, 3.29]

3.2567

镜像站点 mirror site

主站点的副本，通常维护在不同的主机上，以提供冗余、更高的性能或本地访问。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2837]

3.2568

任务 mission

分配给一个资源、一组资源或某些人群的重要运营工作或职责。

注：资源是人力资源，也是包括系统与产品在内的技术资源。

[来源：GB/T 45615-2025]

3.2569

错误 mistake

产生不正确结果的人为动作。

注：容错系统，在人为动作（mistake）、它的显示（硬件和软件故障）、故障的结果（failure）和结果是不正确的（error）总数之间加以区分。

3.2570

混合条目表 mixed entry table

表头由若干行组成、且行内包含有限条目与扩展条目的判定表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2841]

3.2571

混合方式 mixed mode

包含两个或多个不同数据类型的表达式。例如， $Y := X + N$ ，其中，Y 和 X 是浮点变量，而 N 是整数变量。同义词：2.961 条 混合类型（mixed type）。

3.2572

移动设备 mobile device

便携式计算设备，通常具有无线互联网连接和显示屏，支持触摸、手写笔或键盘输入，并可能具备听觉输入和输出功能。

注：由于移动设备的尺寸以及输入和输出可用功能的限制，其设计受到特殊可用性需求的约束。

[来源：ISO/IEC/IEEE 26513:2017, 3.26]

3.2573

移动性规约 mobility schema

对对象移动性施加约束的规范。

[来源：ISO/IEC 10746-3:2009, 16.4.1.1]

3. 2574

模拟对象 mock object

在真实对象可用之前，用于辅助测试的临时虚拟对象。

[来源：ISO/IEC/IEEE FDIS 24765，3.2850]

3. 2575

原型 mock-up

一次性产品。

注：可保留以供验证或培训，并作为记录。

[来源：ISO/IEC/IEEE FDIS 24765，3.2851]

3. 2576

模式 mode

在设计时预见的情况下，系统、其参与者或其组件的预期行为定义。

[来源：ISO/IEC/IEEE 24641:2023，3.1.14]。

2. 系统中使用的设备或数据的类别。

[IEEE 7014-2024，3.1]。

3. 通用操作活动或系统状态，其持续时间作为可靠性、可用性、可支持性或可恢复性的变量。

[IEEE 982:2024，3.1]。

示例：在线模式、离线模式和维护模式；传感器类型、传感器数据或情感特征。

注：模式的主要特征在于该模式下系统的预期功能内容。模式可以反映各种概念，如任务的各个阶段、系统在特定条件下的特定所需功能，或系统使用的特定条件。

[来源：ISO/IEC/IEEE FDIS 24765，3.2852]

3. 2577

模型 model

a) 用来定义一套方法的概念、关系和规则的规约。

注 1：模型使用形式理论，而形式理论基于数学或者科学的原理与概念。模型用已确立的元模型生成。元模型通常用于促进开发准确、完整、一致且可理解的模型。

注 2：模型用于构造或表示实体的架构视图。描述模型和分析模型是模型的两种类别。模型通过符合 ISO/IEC/IEEE 42010 的模型类别进行管理。

注 3：参考模型用于刻画一般实例，该一般实例用作具体条件或情况创建特殊实例模型的基础。参考模型用于促进和加强架构与架构元素的一致性。

注 4：根据具体情况，模型是架构模型、架构实体模型、概念模型或者参考模型。

注 5：物理模型是一种具体表示，它和数学与逻辑模型相区别，后两种模型都是系统的更抽象表示。抽象模型进一步分为描述模型（类似于逻辑模型）和分析模型（类似于数学模型）。

注 6：（建模和仿真）模型是对现实世界过程、概念或系统的选定方面的近似、表示或理想化，这些选定方面包括结构、行为、运行或其他特征。模型以其他模型作为组件。

注 7：给出了 MBSSE 中模型和关系分类的例子（见附录 C）。

[来源：ISO/IEC 24744:2007;GB/T 25000.21-2019;ISO/IEC 24744:2007]

[来源：b) 实体或实体集合的抽象表征，提供了在所关注的条件或情况下，描绘、理解或预测实体或实体集合属性或特征的能力。]

[来源：ISO/IEC/IEEE 42020:2019，3.13，有修改;GB/T 45615-2025]]

3. 2578

模型基线 model baseline

模型配置项及其相关版本和变体的不可变集合。

[来源：ISO/IEC/IEEE 24641:2023, 3.1.17]

3.2579

模型配置项 model configuration item

模型的逻辑组成部分，以受控方式维护，具有可追踪的修订历史。

注：MCI 可以定义在不同的粒度级别，从一组模型元素到整个模型。

[来源：ISO/IEC/IEEE 24641:2023, 3.1.18]

3.2580

模型驱动体系结构 model driven architecture; MDA

它是解决基于不同中间件系统的集成问题。它将不同的系统都结构化成平台独立模型（PIM-platform independent model）和平台特定模型（PSM-platform specific model），并针对不同实现技术与平台制订多个映射规则，然后通过这些映射规则及辅助工具将 PIM 转换成 PSM，再将 PSM 不断求精直至形成最后代码。

3.2581

模型元素 model element

表示单个组件、动作、状态、消息、属性、关系的原子（基本）项，或其他描述系统的组成、特性或行为的其他项。

[来源：GB/T 45615-2025]

3.2582

模型元素库 model element library

在任何项目中均可使用的不可修改模型元素的集合或目录，封装在单个制品中。

[来源：ISO/IEC/IEEE 24641:2023, 3.1.20]

3.2583

模型术语表 model glossary

模型视图中出现的所有已定义概念的名称和定义的集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2858]

3.2584

模型层级 model hierarchy

与 IDEF0 模型层次图结构节点相对应的图表。

[来源：IEEE 1320.1-1998 (R2004), 2.1.74]

3.2585

模型种类 model kind

通过关键特性和建模约定区分的模型类别。

示例：功能模型、活动模型、结构模型、用例模型、地缘政治模型、分析模型和经济模型。

[来源：GB/T 45630-2025, 3.15]

3.2586

模型层 model layers

在定义 CDIF 标准系列时所使用的不同定义（或抽象）层次。

注：CDIF 中的四个模型层分别是用户数据、模型、元模型和元元模型。任何给定的模型层都能准确且完整地定义该层以下一层出现的所有实例。例如，元元模型提供了一组用于构建和理解元模型的定义；元模型则提供了一组用于构建和理解模型的定义。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2587

型号名称 model name

一个独特且具有描述性的名称，用于将一个 IDEF0 模型与其相关的其他 IDEF0 模型区分开来。

注：IDEF0 模型的模型名称和模型名称缩写与模型的用途声明和视角声明一起放在 A-0 上下文图中。

[来源：IEEE 1320.1-1998 (R2004)，2.1.75]

3. 2588

模型名称缩写 model name abbreviation

用于构建图表引用关系、模型名称的唯一简写形式。

[来源：ISO/IEC/IEEE FDIS 24765，3.2861]

3. 2589

模型注释 model note

图表中的文本或图形组成部分，用于记录无法通过图表内的方框、箭头等元素另行表达的事实信息。

[来源：ISO/IEC/IEEE FDIS 24765，3.2862]

3. 2590

模型注释编号 model note number

置于小方框内的整数编号，用于在特定图表内唯一标识某条模型注释。

[来源：ISO/IEC/IEEE FDIS 24765，3.2863]

3. 2591

模型页面 model page

IDEF0 模型的逻辑组成部分，可呈现在单张纸上。

注：模型页面包括图表、文本、前端对象（FEO）和术语表页面。

[来源：IEEE 1320.1-1998 (R2004)，2.1.79]

3. 2592

模型模式 model pattern

在系统或软件设计中，能够用于解决给定语境下常见问题的常规或可复用的模型或模型部件。

[来源：ISO/IEC/IEEE 24641:2023，3.1.21；GB/T 45615-2025]

3. 2593

模型存储库 model repository

在不同抽象层次存储不同模型，并促进不同层次的利益相关者和从业者之间理解与协作的手段。

[来源：ISO/IEC/IEEE 24641:2023，3.1.22；GB/T 45615-2025]

3. 2594

基于模型的系统与软件工程 model-based systems and software engineering, ; MBSSE; MBSSE

对建模的形式化应用，以支持系统与软件工程。[来源：ISO/IEC/IEEE 24641:2023，3.1.16；GB/T 45615-2025]。

3. 2595

基于模型的系统工程 model-based systems engineering; MBSE

建模方法的形式化应用，以支持系统从概念设计阶段一直持续到开发阶段和后续生存周期阶段的需求、设计、分析、验证和确认活动。

[来源：GB/T 45803-2025, 3.1.21]

3. 2596

建模 modeling

<动词>对过程、设备或概念中的某些要素进行表示的活动。

〈名词〉创建系统、解决方案或可交付成果的简化表示形式，例如原型、图表或故事板。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2867]

3.2597

建模工具 modeling tool

为软件产品或信息系统提供建模支持（即表示、描述）的工具。

[来源：ISO/IEC 15474-1:2002, 4.2]

3.2598

可修改性 modifiability

产品在不引入缺陷、不降低现有产品质量的前提下，能够被有效且高效地进行修改的能力。

参见：可分析性、可维护性、模块化。

注：可修改性是可变更性与稳定性的结合。

[来源：ISO/IEC 25010:2023, 3.7.4]

3.2599

可修改的 modifiable

具备相应结构与风格，使得在保留原有结构的前提下，能够完整、一致、正确地实施变更。

[来源 1: ISO/IEC/IEEE 15289:2019, 5.15]。

3.2600

修改请求 modification request

用于识别和描述对产品或服务的拟议变更的信息项。

另见变更请求、变更请求。

注：维护请求可分为修正或增强，并可分为纠正性维护、预防性维护、适应性维护、附加性维护或完善性维护。维护请求可基于利益相关者的请求、事件、投诉、故障或问题报告以及根本原因分析。维护请求还可分为计划内维护、计划外维护和紧急维护。

[来源：ISO/IEC/IEEE 14764:2021, 3.1.8]

3.2601

修改条件/决策覆盖测试 modified condition/decision coverage testing; MC/DC testing

基于结构的测试用例设计技术基于证明决策中的单个布尔条件可以独立影响决策的果。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.36]

3.2602

修改后的源语句 modified source statement

经过修改、变更的源语句。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2873]

3.2603

商用现货软件产品 modified-off-the-shelf; MOTS

已完成开发、可用直接使用或经修改后使用的软件产品，由供方、需方或第三方提供。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2874]

3.2604

模块分解 modular decomposition

将系统分解成组件以方便设计和开发的过程；模块化编程的一个元素。参见：内聚（cohesion）、耦合（coupling）、解模化（demodularization）、因式分解（factoring）、功能分解（functional decomposition）、层次分解（hierarchical decomposition）、包装（packaging）。

[来源：ISO/IEC 24765 FDIS, 3.2877]

3.2605

模块化编程 modular programming

一种软件开发技术，其中软件被开发为一系列模块的集合。

[来源：ISO/IEC 24765 FDIS, 3.2878]

3.2606

模块性 modularity

软件属性，为高度独立组件提供结构。参见：内聚性(cohesion)、耦合性(coupling)、可修改性(modifiability)。

注：模块化意味着产品由具有内聚内容的独立模块或组件构成，且与其他模块或组件的耦合程度最小。

[来源：ISO/IEC 24765 FDIS, 3.2879]

3.2607

模块化 modularization

见：2.973 条 模块分解(modular decomposition)。

3.2608

模块 module

程序中能逻辑地分开的部分。

注：术语“模块(module)”、“部件(component)”和“单位(unit)”常常是可互换的或以不同方法定义为另一个的子元素，取决于上下文。这些术语之间的关系尚未标准化。

3.2609

模块数据 module data

可以被声明它的模块中的任何例程访问但不能被其他模块中的例程访问的数据。

[来源：ISO/IEC 24765 FDIS, 3.2881]

3.2610

模块数据 module data

实例数据 instance data

类数据 class data

可由声明该数据所在模块内的任意例程访问，但不能被其他模块中的例程访问的数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2881]

3.2611

模块强度 module strength

单个软件模块所执行的任务之间相互关联的方式和程度。参见：耦合(coupling)。

注：类型包括巧合性、交际性、功能性、逻辑性、程序性、顺序性和时间性。

[来源：ISO/IEC TR 19759:2016]

3.2612

监控程序，监控器 monitor

与系统或组件同时运行并监督、记录、分析或验证系统或组件运行的软件工具或硬件设备。

[来源：ISO/IEC 24765 FDIS, 3.2885]

3.2613

监控项目工作 monitor and control project work

跟踪、审查和报告整体项目进展，以实现项目管理计划中确定的绩效目标的过程。

[来源：项目管理知识体系指南(PMBOK®指南)第六版]

3. 2614

监控风险 monitor and control risks

[过程] 在整个项目过程中实施风险应对计划、跟踪已识别风险、监控剩余风险、识别新风险以及评估风险的过程。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3. 2615

监督沟通 monitor communications

确保满足项目及其相关方的信息需求的过程。

[来源：项目管理知识体系指南（PMBOK®指南）第六版]

3. 2616

监督风险 monitor risks

在整个项目期间，监督商定的风险应对计划的实施、跟踪已识别风险、识别和分析新风险，以及评估风险管理有效性的过程。

[来源：项目管理知识体系指南（PMBOK®）指南）第六版]

3. 2617

监督相关方参与 monitor stakeholder engagement

监督项目相关方关系，并通过修订参与策略和计划来引导相关方合理参与项目的过程。

[来源：项目管理知识体系指南（PMBOK® 指南）第六版]

3. 2618

可监控性 monitorability

服务提供监控参数和工具以监控服务性能的程度。

服务提供监控参数与监控工具、支持对服务性能进行监控的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2886]

3. 2619

监控 monitoring

确定系统、过程或活动的状态。

[来源：ISO/IEC 19770-1:2017, 3.35]。

3. 2620

监控过程组 monitoring and controlling process group

用于跟踪、审查和调控项目进度与绩效，识别需要对计划进行变更的领域，并启动相应变更的一组过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2888]

3. 2621

单体执行器 monolithic executor

由单一制品构成的执行器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2889]

3. 2622

蒙特卡洛模拟 Monte Carlo simulation

通过对计算机模型进行多次迭代，识别风险与不确定性的潜在影响，以构建决策或行动方案可能产生的一系列结果的概率分布的方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2890]

3.2623

情绪图 mood chart

用于追踪情绪或反应的可视化图表，以识别需要改进的领域。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.2624

机器可读护照 MRP; machine-readable passport; MRP

机器可读护照。

[来源：ISO/IEC TR 29119-13:2022, 3.2]

3.2625

蜕变关系 MR; metamorphic relation; MR

基于测试项的所要求行为，描述测试用例的测试输入变化如何影响预期输出。

[来源：ISO/IEC/IEEE 29119-4:2021 3.34]

3.2626

平均软件故障间隔时间 MTBSF; mean time between software failures; MTBSF

平均软件故障间隔时间。

[来源：IEEE 982:2024, 3.2]

3.2627

平均支持间隔时间 MTBS; mean time between support; MTBS

平均支持间隔时间。

[来源：IEEE 982:2024, 3.2]

3.2628

故障平均恢复时间 MTIR; mean time to impairment recovery; MTIR

故障平均恢复时间。

[来源：IEEE 982:2024, 3.2]

3.2629

平均无故障时间 MTTF; mean time to failure; MTTF

平均无故障时间。

[来源：IEEE 982:2024, 3.2]

3.2630

平均时间类指标（统称） MTX; mean time to x; MTX

平均到达 X 状态所需时间。

[来源：IEEE 982:2024, 3.2]

3.2631

多属性决策 multi-attribute decision; MADM

不只考虑单一准则的决策。

示例：同时基于价格、交付日期和质量做出的决策。

[来源：ISO/IEC/IEEE 4765:2025, 3.2917]

3.2632

多组件系统 multi-component system

由多个软件组件组成的被测系统。

[来源：ISO/IEC 29881:2010, A.10]

3. 2633

多核 multi-core

集成两个或两个以上微处理器单元的芯片。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2919]

3. 2634

多核处理器 multi-core processor

内部包含两个及以上处理单元的单片集成电路芯片。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2920]

3. 2635

多准则决策分析 multi-criteria decision analysis; MCDA

一种采用多项准则,对一组备选方案的整体绩效进行透明化评估与比对的技术。其总体目标是形成备选方案的优先次序。该分析需构建由备选方案与评价准则构成的矩阵,经排序、聚合计算后,得出各方案的综合得分。此类技术也称为多属性决策或多目标决策。

[来源: ISO/IEC 31010:2019, B.29]

3. 2636

多维结构 multidimensional construct

由若干一维结构组成的结构。

[来源: ISO/IEC 33003:2015, 3.11]

3. 2637

多学科仿真 multidisciplinary collaborative simulation

建模仿真人员在一个协同、互操作的环境中方便、快捷和友好地采用各自领域的专业分析工具,对构成系统的各子系统进行建模与仿真分析,或从不同技术视图进行功能、性能的单点分析,并透明地支持它们参与整个系统的联合仿真,协作完成对系统仿真的一种复杂系统仿真分析方法。

注:虚拟样机系统通常需要进行多学科仿真工作。

[来源: GB/T 36328-2018 信息技术 软件资产管理 标识规范]

3. 2638

多级安全 multilevel security

一种操作方式,当至少有某些用户对系统中包括的全部数据既不清楚也不需要知道时,它允许处于各种安全级上的数据并行地在计算机系统中存储和处理。

3. 2639

多模态生物特征识别系统 multi-modal biometric system

基于多种生物特征的生物特征系统。

[来源: ISO/IEC TR 29119-13:2022, 3.1.31]

3. 2640

多[重]继承 multiple inheritance

子类型继承其所有超类型(及其超类型的超类型)的所有元属性和元关系的情况。

[来源: ISO/IEC 15474-2:2002, 6.2.6]

3. 2641

多实例方法 multiple instance approach

单独计算同一功能的每种实现/交付方法的情况。

参见: 单实例方法(single instance approach)。

[来源: ISO/IEC/IEEE 3240:2025, 3.1.23]

3.2642

多读写者 multiple readers and writers

允许多个读取者同时访问共享数据存储库的算法；但是，写入者必须具有互斥访问权限才能更新数据存储库。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2930]

3.2643

多准则决策 multiple-criteria decision making; MCDM

对具有多项评价准则的可选方案进行偏好决策（如评估、优先级排序与遴选）。

注：仅含一个备选方案的多准则决策等同于综合度量指标的构建。

[来源：ISO/IEC 33003:2015, 3.9]

3.2644

多命中判定表 multiple-hit decision table

存在至少一组条件可被多条规则同时满足的判定表。

[来源：ISO 5806:1984, 3.3]

3.2645

多点估算 multipoint estimating

当单个行动估算存在不确定性时，通过应用乐观、悲观和最可能估算值的平均值或加权平均值来估算成本或工期的方法。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.2646

多重集基数 multiset cardinality; cardinality of a multiset

多重集中每个成员的重数之和。

[来源：ISO/IEC 15909-1:2019, 3.19]

3.2647

多重集 multiset

包 bag

允许元素重复出现的元素集合。

[来源：ISO/IEC 15909-1:2019, 3.17]

3.2648

多值属性 multi-valued property

具有多值映射的属性。

参见：单值属性（single-valued property）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2924]

3.2649

可变类 mutable class

实例集不固定的类；其实例会随着时间的推移创建和消亡。

参见：不可变类（immutable class）、状态类（state class）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2941]

3.2650

互斥 mutual exclusion

在同一时刻只允许一个任务访问共享数据。

注：可以通过二进制信号量或使用监视器来强制实现。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2651

排斥的 *mutually exclusive*

相互排斥

从中可选择至多一者的备选项。

参见: 相互包容。

[来源: ISO/IEC 26580:2021, 3.9]

3.2652

互斥聚类 *mutually exclusive clustering*

一种任务结构化准则, 在该准则中, 一组对象被组合成一个任务, 因为在任何时候只能执行一个对象。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2653

相互包容 *mutually inclusive*

从中选择了零个或多个的备选方案。

参见: 相互排斥。

[来源: ISO/IEC 26580:2021, 3.10]

3.2654

N 2 图 *N 2 diagram, N2 diagram*

一种用于列表化、定义、分析和描述系统组件之间功能接口和交互的系统工程或软件工程工具。

注1: N 2 图是一种矩阵结构, 用图形方式显示给定系统或结构中功能与组件之间的双向相互关系。

同义词: N2 图 (N2 diagram)。

[来源: ISO/IEC/IEEE 24765e:2015]

3.2655

名称解析 *name resolution*

在给定初始名称和初始命名上下文的前提下, 查找名称与初始名称指定的实体之间的关联关系的过程。

[来源: ISO/IEC 10746-2:2009, 12.8]

3.2656

命名空间 *name space*

可用作名称的一组术语。

[来源: ISO/IEC 10746-2:2009, 12.3]

3.2657

具名常量 *named constant*

具名常数

指代程序执行过程中不会改变的数值或字符串值的标识符。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2658

具名约束 *named constraint*

特定于特定模型的约束, 而不是某些建模构造中固有的约束。

注: 例如基数约束。命名约束是显式命名的, 其含义以自然语言表述, 其实现以规范语言编写。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.122]

3. 2659

命名操作 naming action

将名称空间中的术语与给定实体相关联的操作。

[来源：ISO/IEC 10746-2:2009, 12.5]

3. 2660

命名上下文 naming context

一组名称和一组实体之间的关系。

[来源：ISO/IEC 10746-2:2009, 12.5]

3. 2661

命名域 naming domain

命名上下文的子集，这样所有命名操作都由域的控制对象（名称权限对象）执行。

[[来源：ISO/IEC 10746-2:2009, 12.6]

3. 2662

命名图 naming graph

有向图，其中每个顶点表示一个命名上下文，每条边表示一个出现在源命名上下文中的名称和目标命名上下文之间的关联。

[来源：ISO/IEC 10746-2:2009, 12.5]

3. 2663

狭义人工智能 narrow artificial intelligence

聚焦于单一明确任务以解决特定问题的人工智能。

同义词：狭义人工智能（narrow AI）、弱人工智能（weak artificial intelligence）、弱人工智能（weak AI）。

[来源：ISO/IEC TR 29119-11:2020, 3.1.47]

3. 2664

n 元关系 N-ary relationship

元数（度） $n > 2$ 的关系。

注：具有两个以上参与实体的关系。单个实体可以在单个关系中参与多次。

[来源：ISO/IEC 15474-1:2002, 4.2]

3. 2665

自然单位 natural unit

用于量化软件产品使用频率的、针对特定应用的绝对或相对时间的替代单位。

示例：传感器中断、运行次数、用户菜单选择、输出页数、事务、电话呼叫、短信、作业、查询或 API 调用。

[来源：IEEE 982:2024, 3.1]

3. 2666

导航 navigation

通过在网站或电子文档中的不同位置之间移动来访问界面信息的过程。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.17]

3. 2667

导航辅助 navigational aids

帮助用户在计算机应用程序中导航的软件功能。

示例：快捷键，可以更快地在对话框间移动。

[来源：ISO/IEC 20968:2002, 10; GB/T 42566-2023]

3.2668

保密协议 NDA; non-disclosure agreement; NDA

保密协议。

[来源: ISO/IEC/IEEE 41062:2024, 3.2]

3.2669

非发展项目 NDI; NDI

3.2670

需求陈述 need statement

将一个或多个生存周期概念正式转化为对某实体执行特定功能或具备某种质量的共识期望的结果。

[来源: INCOSE SE Handbook, 第五版.]

3.2671

协商 negotiation

通过相关方之间的磋商解决争议。

[来源: ISO/IEC/IEEE FDIS 24765, 3.2976]

3.2672

嵌套 nest

将一种计算机程序结构嵌入到另一种同类结构中的行为。

示例: 将 WHILE 循环嵌入另一个 WHILE 循环或将 IF 测试嵌入 WHILE 循环。

[来源: ISO/IEC/IEEE 24765:2017, 3.2978]

3.2673

网络开放端 network open end

没有任何前置活动或后续活动, 从而导致进度网络路径中断的进度活动。

注: 网络开放端通常是由于缺失逻辑关系引起的。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2674

网络路径 network path

在项目进度网络图中通过逻辑关系连接的任何连续进度活动序列。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版)]

3.2675

建立工作关系网 networking

与可以协助实现目标和职责的人建立关系。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2676

神经网络 neural network

2. 由通过权重可调的加权链接相连的基本处理元素所构成的网络; 其中每个元素通过对其输入值应用非线性函数产生一个值, 并将其传输给其他元素或作为输出值呈现。

示例 1: 阈值函数, Sigmoid 函数, 多项式函数。

示例 2: 阈值函数, Sigmoid 函数, 多项式函数。

示例 3: 同义词: 人工神经网络 (artificial neural network)。

注: 尽管有些神经网络旨在模拟神经系统中神经元的工作机制, 但大多数神经网络在人工智能中被用作联结主义模型的实现。

[来源: ISO/IEC TR 29119-11:2020, 3.1.48]

3.2677

神经元覆盖率 neuron coverage

在一组测试中，神经网络内被激活神经元的数量占神经元总数的比例（通常以百分比表示）。

注：如果神经元的激活值超过零，则认为该神经元被激活。

[来源：ISO/IEC TR 29119-11:2020, 3.1.49]

3.2678

新源语句 new source statements

新增和修改的源语句总和。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2679

非功能规模度量 NFSM; non-functional size measurement; NFSM

非功能规模度量。

[来源：ISO/IEC/IEEE 32430:2025, 3.2]

3.2680

节点字母 node letter

作为节点号的第一个字符的字母。

[来源：ISO/IEC/IEEE 24765m:2024, 2.1.82]

3.2681

节点号 node number

明确标识函数在模型层次结构中位置的表达式。

注：节点号由节点字母、包含表示该功能的框图的图号，以及该框的框号拼接构成。

[来源：ISO/IEC/IEEE 24765m:2024, 2.1.83]

3.2682

名义尺度 nominal scale

测量值为类别形式的尺度。

示例：按照缺陷类型进行分类，并不意味着这些类别之间存在顺序关系。

[来源：ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary]

3.2683

非补偿性决策技术 noncompensatory decision technique

一种多属性决策技术，它对所有属性赋予同等权重，不允许用一个属性上的较差表现去换取另一个属性上的较好表现。

参见：补偿性决策技术。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2684

非补偿性模型 non-compensatory model

不允许各准则按其权重彼此补偿的多准则决策模型。

注：极强的正向或负向的项会对总体综合值产生不成比例的影响，尽管权重保持不变。根据评价策略、综合度量的目的或测量尺度的不同，存在各种非补偿性模型。

[来源：ISO/IEC 33003:2015, 3.12]

3.2685

不符合 nonconformity

不合格

未满足需求。

同义词：不符合（non-conformity）。

[来源：ISO/IEC 19770-1:2017, 3.36]

3.2686

非交付项 non-deliverable item

合同中不需要交付但可用于产品开发的硬件或软件产品。同义词：非交付项(nondeliverable item)。

[来源：ISO/IEC 12207:2008, 4.20]

3.2687

非交付源语句 nondelivered source statement

为支持最终产品而开发、但不交付给客户的源语句。

同义词：非交付源语句（non-delivered source statement）。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2688

非确定性系统 non-deterministic system

给定一组特定的输入和初始状态时，并不总是会产生相同的一组输出和最终状态的系统。

同义词：非确定性系统（nondeterministic system）。

[来源：ISO/IEC TR 29119-11:2020, 3.1.50]

3.2689

未新开发源语句 nondeveloped source statement

被复用或删除的现有源语句。

同义词：非开发源语句（non-developed source statement）。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2690

非研制的 nondevelopmental

在当前获取或开发过程中实际使用前已经开发完成的。

注：此类项目可能需要进行微小修改才能满足其当前预期用途的要求。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2691

无量纲标度 nondimensional scaling

一种决策技术，其中属性值被转换为一个共同的尺度，在该尺度上可以将它们相加从而为每个备选项形成一个综合得分。

参见：加法加权，层次分析法，补偿性决策技术。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2692

非功能性需求 non-functional requirement; NFR

针对软件密集型系统或软件产品的需求，包括其应如何开发和维护，以及其在运行中应如何表现，但不包括任何软件的功能性用户需求。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.25]

3.2693

软件的非功能性需求 non-functional requirements for software

由软件实现的非功能性需求（NFR）部分。

参见：软件非功能性需求（NFR for software）。

[来源: ISO/IEC/IEEE 32430:2025, 3.1.26]

3.2694

非功能规模 non-functional size

依据既定的规则集, 对非功能需求 (NFR) 进行量化而得出的软件规模。

[来源: ISO/IEC/IEEE 32430:2025, 3.1.24]

3.2695

非人力资源成本 non-human resource cost

为达成软件项目目标所需要付出的人力成本之外的其他资源代价总和。

[来源: GB/T 36964-2018]

3.2696

非标识关系 nonidentifying relationship

父实体主键中包含的部分或全部属性不参与子实体主键的特定 (非多对多) 关系。

同义词: 非标识关系 (non-identifying relationship)。

参见: 标识关系, 强制非标识关系, 可选非标识关系 [键模式]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.123]

3.2697

非内在关系 nonintrinsic relationship

部分、多值或可变的关系。

参见: 内在关系。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.125]

3.2698

非键属性 nonkey attribute

不是实体的主键或复合主键组成部分的属性。

注: [键模式]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.125]

3.2699

非主要实体 non-primary entity

通过第三范式分析得出、且并非构建目标应用系统所针对的主要实体类型的数据实体类型。

参见: 系统实体。

注: 非主要实体只包含有少数属性, 例如代码、描述。

[来源: ISO/IEC 20968:2002, 10; GB/T 42566-2023]

3.2700

非过程语言 nonprocedural language

用户无需说明计算机必须按给定顺序执行的特定指令即可说明要实现的目标的语言。参见: 过程式语言、声明式语言、交互式语言、基于规则的语言。

[来源: ISO/IEC/IEEE 24765:2017, 3.2625]

3.2701

非过程编程语言 nonprocedural programming language

用于表达问题参数而不是解决方案步骤的计算机编程语言。

同义词: 非过程编程语言 (non-procedural programming language)。

参见: 过程编程语言。

示例: 报告编写器或排序规范语言。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2702

不可编程终端 nonprogrammable terminal

不具备独立数据处理能力的用户终端。

参见: 不可编程终端 (non-programmable terminal)。

[来源: ISO/IEC/IEEE 24765 FDIS, 3.3022; ISO/IEC/IEEE FDIS 24765]

3.2703

不可否认性 non-repudiation

2. 产品具备的一种能力, 证明操作或事件已发生, 以确保该事件或操作后续无法被否认。

[来源: ISO/IEC 25010:2023, 3.6.3]

3.2704

非技术需求 nontechnical requirement

影响产品和服务采购或开发、但并非产品或服务本身属性的需求。

示例: 待交付产品或服务的数量、交付的商用现货 (COTS) 非开发项的数据权利、交付日期、带退出准则的里程碑、与培训相关的工作约束、现场保障条件以及部署进度计划。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3023]

3.2705

非终结符 non-terminal symbol

语法层级定义的组成部分, 在层级结构中可进一步分解。

[来源: ISO/IEC 15474-1:2002, 4.2]

3.2706

非时间关键型计算密集任务 non-time-critical computationally intensive task

消耗空闲 CPU 周期的低优先级计算密集型任务。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.2707

非易失性存储器 non-volatile memory

无论通电与否均可保存数据的存储单元。

[来源: ISO/IEC/IEEE 24765c:2014]

3.2708

规范 norm

具有常规性、典型性或标准性的事物。

[来源: IEEE 7005 2021, 3.1]

3.2709

规范化 normalization

数据库设计人员将数据结构转换为一组不存在重复组的关系的过程。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3026]

3.2710

修订通知 notice of revision; NOR

在配置管理中用的一种格式, 用以对图、表进行修订提出建议, 以及在批准后通知用户图或表已经或将相应地修订。

参见: 配置控制 (configuration control)、工程变更 (engineering change)、规格说明变更通知 (specification change notice)。

[来源: GB/T 11457:2006; ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE 24765 FDIS, 3.3024]

3.2711

正常运行响应 NRR; normal run response; NRR

正常运行响应。

[来源: IEEE 982:2024, 3.2]

3.2712

正常运行终止 NRT; normal run termination; NRT

正常运行终止。

[来源: IEEE 982:2024, 3.2]

3.2713

N平方图 N-squared diagram; N2 diagram

用于定义目标系统内部运行关系或外部接口的图形化表示。

[来源: INCOSE 第五版. APPENDIX C]

3.2714

正常支持响应 NSR; normal support response; NSR

正常支持响应。

[来源: IEEE 982:2024, 3.2]

3.2715

非终止性故障 NTF; non-terminal failure; NTF

非终止性故障。

[来源: IEEE 982:2024, 3.2]

3.2716

数据元素类型的数量 number of data element types

基本过程（EP）的输入/输出/查询中所有数据元素类型（DETs）的总和，加上边界内读取或维护的内部数据元素。

[来源: ISO/IEC/IEEE 32430:2025, 3.1.27]。

3.2717

数字的 numeric

用于描述由数字构成的数据，以及使用这些数据的功能单元。

[来源: ISO/IEC 2382:2015, ?]

3.2718

网络视频录像机 NVR; network video recorder; NVR

网络视频录像机。

[来源: IEEE 1012-2024, 3.2]

3.2719

开放认证 OAUTH; open authentication; OAUTH

开放认证。

[来源: ISO/IEC/IEEE 23026:2023, 3.2]

3.2720

目标, 对象 object

对数据及操纵该数据的服务的封装。

[来源: ISO/IEC 19500-1:2012, 5.3.1]

3. 2721

对象图 object diagram

一种由在某一时刻的各对象及其关系所组成的图。对象图为可视类图或协作图的特例。参见：2. 212 条 类图 (class diagram) 和 2. 231 条 协作图 (collaboration diagram)。

3. 2722

对象流状态 object flow state

在某一活动图中的一种状态，它表示一个对象从一个状态中的动作输出到另一状态中的动作输入的传递的状态。

3. 2723

对象标识符 object identifier

对象（实例）身份的具体表示。

注：对象标识符 (OID) 用于展示具有身份的实例示例、形式化身份概念，并在编程语言或数据库系统中支持此概念。

[来源：IEEE 1320.2-1998 (R2004)，3.1.128]

3. 2724

对象实现 object implementation

提供创建对象所需信息并允许该对象参与提供适当服务集的定义。

[来源：ISO/IEC 19500-1:2012，5.4.2]

3. 2725

目标语言 object language

一种语言，它表示从机器辅助转换过程的输出。参见：源语言。

示例 1：由汇编器或编译器输出的语言。

示例 2：[来源 ISO/IEC/IEEE 26511:2018，3.1.32]。

3. 2726

对象生存线 object lifeline

在时序图中，一条表示对象经历时期的线。参见：2.1437 条 时序图 (sequence diagram)。

3. 2727

目标程序 object program

汇编器或编译器输出的计算机程序。

同义词：目标程序。

参见：源程序对象模块。

[来源：ISO/IEC/IEEE 24765:2017，3.2646]

3. 2728

对象引用 object reference

存储明确访问对象引用的值。

[来源：ISO/IEC 19500-2:2012，3.2.24]

3. 2729

对象集 object set

对象类型集合中所有对象类型的所有可能实例化组成的集合的子集。

注：对象集是对象类型集合中各成员之并集的子集；对象集的集合包含空集及对象类型集合各成员之并集本身。对象集通过箭头建模。

[来源：IEEE 1320.1-1998 (R2004)，2.1.86]

3. 2730

对象类型 object type

成员为对象引用的类型。

注：IDEF0 对象类型通常类似于 IDEF1X 实体或 IDEF1 实体类。

[来源：ISO/IEC 19500-1:2012, 5.3.4]

3. 2731

对象类型集 object type set

一个或多个对象类型的命名集合。

注：对象类型集合可包含本身已分组为对象类型集合的对象类型。对象类型集合通过箭头标签进行标识。

[来源：IEEE 1320.1-1998 (R2004), 2.1.88]

3. 2732

目标函数 objective function

将决策变量与方案的成本或收益相关联的公式。参见：成本函数、收益函数。

[来源：ISO/IEC/IEEE 24765:2017, 3.2655]

3. 2733

面向对象的设计 object-oriented design

系统或部件用对象和它们之间连接的术语表示的软件开发技术。参见：以数据结构为中心的设计 (data structure-centered design)、输入-处理-输出 (input-process-output)、模块分解 (modular decomposition)、快速原型 (rapid prototyping)、逐步细化 (stepwise refinement)、结构设计 (structured design)、事务分析 (transaction analysis) 和转换分析 (transform analysis)。

[来源：ISO/IEC/IEEE 24765:2017, 3.2651]

3. 2734

办公自动化 office automation; OA

利用信息处理系统实现办公活动的集成。

注：该术语特别包括文本、图像和声音的处理与通信。

3. 2735 [[来源：ISO/IEC 2382:2015, X.X]

偏移量 offset

与相对地址相加以确定要访问的存储单元地址的数值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3081]

3. 2736

开箱即用 off-the-shelf; OTS

已开发并可用的产品或系统。

[来源：ISO/IEC/IEEE 41062:2024, 3.1.12]

3. 2737

现货产品 off-the-shelf product

由供方、需方或第三方提供的、已经开发出来的、可得到、可使用的、现成的或需要加以修改的产品。

3. 2738

片上振荡器 on-chip oscillator

微型计算机上产生周期性电子信号的电子电路，常用于设备时钟。

[来源：ISO/IEC/IEEE 24765d:2015]

3.2739

按需调度 on-demand scheduling

根据对客户的感知价值，从待办工作中抽取任务，并在资源可用时进行分配的调度方法。

参见：晚绑定。

[来源：Software Extension to the PMBOK® 指南 Fifth Edition]

3.2740

一次性密码/一次性编程 one-time password/one-time programming; OTP

一种仅能被写入一次的 ROM 数据记录方法。

[来源：ISO/IEC/IEEE 24765c:2014; ISO/IEC/IEEE FDIS 24765, 3.3094]

3.2741

一次性可编程 one-time programming; OTP

一次性可编程只读存储器的数据写入方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3094]

3.2742

在线文档 online documentation

用户通过使用软件获取的信息条目。

注：可以是上下文相关的。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3097]

3.2743

联机帮助 online help

用户在使用软件时在屏幕上阅读的关于软件的信息。

注意：联机帮助可以以多种形式显示（上下文帮助、屏幕提示和示例）。

[来源：ISO/IEC/IEEE 26513:2017, 3.28]

3.2744

屏幕文档 on-screen documentation

旨在供用户在使用软件时在计算机屏幕上阅读的文档。

印刷文档、嵌入式文档。

示例：屏幕上的弹出式帮助和帮助文本。

[来源：ISO/IEC/IEEE 26512:2018, 3.15]

3.2745

用户的屏幕信息 onscreen information for users

旨在供用户在使用软件时在屏幕上阅读的用户信息。

参见：为用户嵌入的信息、为用户打印的信息。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.36]

3.2746

面向对象的系统工程方法 OOSEM; object-oriented systems engineering method; OOSEM

面向对象的系统工程方法。

[来源：ISO/IEC/IEEE 24641:2023. 3.2]

3.2747

开放分布式处理 open distributed processing; ODP

设计为符合 ODP 标准的分布式处理。

同义词：ODP。

[来源: ISO/IEC 10746-2:2009, 3.2.4]

3.2748

开放软件基金会 open software foundation; OSF

开放软件基金会。

[来源: ISO/IEC 10746-1:1998; ISO/IEC/IEEE FDIS 24765, 3.3172]

3.2749

开源软件 open source software; OSS

开源软件。

参见: FOSS。

[来源: IEEE 2675-2021, 3.2; ISO/IEC/IEEE FDIS 24765, 3.3175]

3.2750

开放系统 open system

一种按标准建立起来的计算机系统。开放系统的主要特征是: 兼容性 (compatibility) 见: 2.249 条 兼容性 (compatibility); 可移植性 (portability) 见: 2.1149 条 可移植性 (portability); 互操作性 (interoperability) 见: 2.807 条 互操作性 (interoperability); 可扩展性 (scalability) 见: 2.1411 条 可扩展性 (scalability)。开放系统可以使用户摆脱特定厂商的控制, 并有利于提高软件厂商的效益。

3.2751

开放系统环境 open system environment; OSE

开放系统环境。

[来源: ISO/IEC 10746-1:1998; ISO/IEC/IEEE FDIS 24765, 3.3171]

3.2752

开放系统互连 open systems interconnection; OSI

开放系统互连。

[来源: ISO/IEC 10746-1:1998; ISO/IEC/IEEE FDIS 24765, 3.3173]

3.2753

操作数 operand

运算对象、常量或函数, 即将要对其执行运算的对象。

示例: 在表达式 $A = B + 3$ 中, B 和 3 均为运算对象。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3108]

3.2754

(软件) 运行环境 operating environment (software)

在指定计算机系统上并发运行的软件集合。

[来源: ISO/IEC 19761:2011 软件工程 — COSMIC: 一种功能规模测量方法, 2.20]

3.2755

运行模式 operating mode

系统的运行类型 示例: 高速模式、低功耗模式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3110]

3.2756

操作系统 operating system; OS

一组软件、固件和硬件元素, 用于控制计算机程序的执行, 并在计算机系统中提供计算机资源分配、作业控制、输入/输出控制和文件管理等服务。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3111]

3.2757

操作 operation

运行

运算

在计算机算术操作中,由一个操作符在一个或多个操作数上规定的动作。例如,在表达式 $A=B+3$ 中,加 B 至 3 以得到 A 的过程。

在程序设计中,定义由计算机系统执行的动作、例如,相加、比较、分支等。

注 1: 与算术操作不同,操作可能不包含操作符或操作数、例如暂停 (halt) 操作。

注 2: 在目标环境中执行目标函数的计算机系统的运行过程。

注 3: 在面向对象中,操作是对对象行为的指正。

[来源: GB/T 11457-2006, X.X]

3.2758

运营和维护成本 operation and maintenance costs

与使用资产相关的成本以及将其保持在可用状态的成本。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3113]

3.2759

运行和维护阶段 operation and maintenance phase

软件生存周期中,软件产品在其运行环境中投入使用、监控其运行性能,并根据需要进行修改以纠正问题或响应不断变化的需求的时间段。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3114]

3.2760

操作码 operation code

指定一种计算机操作的字符或字符集。

示例: 代码 BNZ 用于表示“非零则转移”操作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3115]

3.2761

运行约束 Operation constraints

由运行策略所产生的对系统的约束,包括成本、进度和技术方面的约束。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2762

运行使能系统需求 Operation enabling system requirements

支持目标系统运行所需的各类系统要求。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2763

操作异常 operation exception

程序遇到无效操作码时引发的异常。参见: 寻址异常、数据异常、溢出异常、保护异常、下溢异常。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3116]

3.2764

操作字段 operation field

计算机指令中指定要执行的操作的字段。参见: 地址字段。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3117]

3.2765

操作接口 operation interface

所有交互均为操作的接口。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3118]

3.2766

操作记录 Operation record

与操作相关的永久、可读形式的数据、信息或知识。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.2767

运行报告 Operation report

为向相关方通报运行活动的状态、结果与成效而编制的报告。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.2768

运行策略 Operation strategy

执行系统运行所需的方法、进度计划、资源及特定考量因素。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.2769

运行概念 operational concept; OpsCon

组织对特定系统或一组相关的特定新的、现有的或修改的系统的一个或一系列运行行为的设想或意图的文字和图形表述。

参见：运营构想

注 1：运行概念旨在从用户和操作者角度，对组织的作业环境中使用一个或多个特定系统或与一组相关的系统，给出运行的整体描述。

注 2：运行概念是关于系统的，而运营构想通常与组织相关。

[来源：GB/T 45615-2025; ISO/IEC/IEEE 15288:2023, 3.23]

3.2770

操作约束 operational constraint

产品在遇到操作风险时，将其操作限制在安全参数或状态范围内的能力。

[来源：ISO/IEC 25010:2023, 3.9.1]

3.2771

运行环境 operational environment

用于支持系统持续运行的物理环境、设置和条件。

注：包括直接因素（开发、技术、业务、运营、组织）以及在某些情况下可能对系统可信性产生重大影响的间接因素（政治、伦理、经济、法律、监管、生态和社会）。

[来源：IEEE 982:2024, 3.1]

3.2772

运行产品 operational product

在实际运行条件下发挥功能的产品。

[来源：ISO/IEC TR 14759:1999 软件工程 模型与原型 软件模型与原型模型分类及其应用, 3.2 e)]

3.2773

运行可靠性 operational reliability

在实际使用环境中,系统或软件子系统的可靠性。运行可靠性可能与规定环境或测试环境中的可靠性有很大的不同。

3.2774

运行场景 operational scenario

对想象中的事件或活动序列的描述,包括产品或服务与其环境和用户(3.1.35)的交互,以及在具有最终用途意义时其产品或服务组件之间的交互。

注:运行场景用于评估需求(3.1.19)和系统设计,以及验证和确认系统。

[来源:GB/T 45802-2025 系统与软件工程 生存周期过程 需求工程]

3.2775

运行软件 operational software

预定在交付给顾客,并部署在预定环境下的系统中使用和运行的软件。

3.2776

运行测试与评估 operational test and evaluation; OTE

运行试验在目标使用环境与目标用户群体中,对完整生物特征系统进行评估。运行评估与技术评估或场景评估的区别在于,不再为可重复性测试控制用户群体、环境与系统设计,而是按实际使用场景变化。

[来源:ISO/IEC 19795-6:2012, 3.1]

3.2777

运营 operations

持续开展可产出相同产品或提供重复性服务的活动 示例:生产、制造、会计核算。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3126]

3.2778

操作员手册 operator manual

提供启动和操作系统或部件必需的信息的文档。典型地描述准备、操作、监控和恢复过程。

注:当在操作计算机系统(安装磁带等等)的人与为了一定的目的使用计算机系统的人之间有区分时,操作员手册和用户手册有区别。参见:2.476条 诊断手册(diagnostic manual)、2.766条 安装手册(installation manual)、2.1250条 程序员手册(programmer manual)、2.1631条 支持手册(support manual)和2.1813条 用户手册(user manual)。

[来源:GB/T 11457-2006, 2.1068]

3.2779

操作手册 operator manual; operations manual

提供启动和操作系统或组件所需信息的文件。

参见: 诊断手册(diagnostic manual)、安装手册(installation manual)、程序员手册(programmer manual)、支持手册(support manual)、用户手册(user manual)。

注:通常描述的是准备、运行、监控和恢复的流程。当需要区分操作计算机系统的人员与按既定用途使用系统的人员时,操作员手册与用户手册是不同的。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3128]

3.2780

操作员/维护人员培训材料 Operator/maintainer training materials

培训能力和培训文档。

[来源:INCONE 系统工程手册 v4, 附录 E]

3. 2781

对象过程法 OPM; OPM

对象过程法。

[来源: ISO/IEC/IEEE 24641:2023, 3. 2]

3. 2782

机会 opportunity

不确定性对目标达成的正面影响。

[来源: GB/T 42560-2023, 3. 1. 9]

3. 2783

机会成本 opportunity cost

将资金投入某项活动而产生的隐性成本, 该资金因此无法再用于其他投资。

参见: 最低期望收益率 (minimum attractive rate of return (MARR))。

注: 进行一项投资意味着同一笔资金无法再投入到可获得最低期望收益率 (MARR) 的其他领域。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3131]

3. 2784

系统操作概念 OpsCon

对组织关于特定系统或一组相关的新建、现有或已修改的特定系统的一项操作或一系列操作的假设或意图进行的文字与图形表述。

[来源: ISO/IEC/IEEE 15288:2023, 3. 23]

3. 2785

光盘 optical disc; OD

以凹坑形式存储二进制数据的磁盘, 这些凹坑会中断激光的反射。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3134]

3. 2786

乐观工期 optimistic duration

在考虑已知会影响执行情况的变量的前提下, 对活动最短持续时间所做的估算。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3135]

3. 2787

优化分析 optimization analysis

在特定情境下, 对相互制约的各组成部分进行权衡, 以实现最佳性能。

注: 示例, 运行速度更快的算法通常会占用更多内存。优化就是在更快的运行时间与额外内存开销之间进行价值权衡。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3136]

3. 2788

优化过程 optimizing process

基于对过程中固有变异的常见原因的理解而进行改进的量化管理过程。

注: 优化过程的重点在于, 通过渐进式改进和创新式改进, 持续提升过程性能的范围。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3137]

3. 2789

可选择的 optional

用于指定部分映射的语法关键字。

参见: 强制性的、部分的。

注：可选建议使用“should”表述。

[来源：ISO/IEC 24765 FDIS, 3.3138]

3.2790

可选属性 optional attribute

可为实例取空值的属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3139]

3.2791

可选非标识关系 optional nonidentifying relationship

子实体实例可不与父实体实例相关联而存在的非标识关系。

参见：强制非标识关系、非标识关系[来源：关键样式]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3140]

3.2792

可选要求 optional requirement

为了符合规范性文件所允许的特定选择而必须遵守的要求。

注1：可选要求可以是：

a) 两个或更多的可选择的要求中的一个；

b) 仅在适用时必须符合的，而在不适用时则可不予考虑的附加要求。

[来源：ISO/IEC 14143-2:2011, 3.6]

3.2793

可选任务 optional task

为满足特定需求，可在最低限度测试任务基础上增加的任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3142]

3.2794

顺序标度 ordinal scale

测量值进行排序的标度。

参见：等距标度、名义标度、比率标度。

注：例如，将缺陷按严重程度分级，即为一种排序。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.2795

组织基础设施 Organization infrastructure

支持组织的资源和服务。用于硬件制造、软件开发、系统实施和集成、验证、确认等的组织级设施、人员和资源。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.2796

组织基础设施需求 Organization infrastructure needs

组织对基础设施产品或服务的具体要求，包括对外部利益相关者的承诺。

[来源：INCONE 系统工程手册 v4, 附录 E]

3.2797

组织经验教训 Organization lessons learned

组织相关的经验教训，指对已实施纠正措施的评估或观察结果，该纠正措施有助于提升绩效或增强能力。吸取的教训也来自对积极发现的评估或观察，此类积极发现未必需要纠正措施，仅需持续维持即可。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2798

组织组合方向和约束 Organization portfolio direction and constraints

组织业务目标、资金支出和约束、正在进行的研发 (R&D)、市场趋势等, 包括成本、进度和解决方案约束。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2799

组织战略规划 Organization strategic plan

整体组织战略, 包括业务使命或愿景以及战略目标和目的。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2800

组织裁剪战略 Organization tailoring strategy

将新的或更新的外部标准纳入组织的标准生存周期过程集所需的方法、进度计划、资源和具体考虑。

[来源: INCOME 系统工程手册 v4, 附录 E]

3.2801

组织分解结构 organizational breakdown structure; OBS

项目组织的层级化表示, 用于说明项目活动与将执行这些活动的组织单元之间的关系。

[来源: 《项目管理知识体系指南 (PMBOK® 指南)》第七版]

3.2802

组织管理 organizational management

指导和控制组织的任务与职责。

[来源: ISO/IEC 29110-4-2:2021, 3.1.1]

3.2803

组织管理概况 organizational management profile

面向微型实体 (VSE) 制定的概况, 用于提供额外的组织管理指导。

[来源: ISO/IEC 29110-1-2:2024, 3.55]

3.2804

组织成熟度 organizational maturity

组织在多大程度上明确且一致地部署了记录、管理、测量、控制和持续改进的流程。

注: 组织成熟度可以通过评估来衡量。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3152]

3.2805

组织计划 organizational plan

规定实现组织目标的程序的文件化信息。

[来源: ISO/IEC 19770-1:2017, 3.40]

3.2806

组织方针、程序和资产 Organizational policies, procedures, and assets

与组织标准生存周期过程集相关的条目, 包括指南和报告机制。组织过程指南以组织方针、程序和资产的形式出现, 用于实施系统生存周期过程并为满足单个项目需求进行适配 (例如, 模板、检查清单、表格), 包括为组织内的所有 SE 流程定义职责、责任和权限。

[来源: INCOME 系统工程手册 v4, 附录 E]

3. 2807

组织政策 **organizational policy**

通常由最高管理层制定并为组织采纳，用于影响和确定各项决策的指导原则。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3155]

3. 2808

组织过程资产 **organizational process assets**

用以支持组织过程的部署、应用、管理以及改进的制品和数据等。

注：典型地，组织过程资产包括过程架构、模板、标准、数据、风险和机会库等。

[来源：GB/T 42560-2023, 3.1.10]

3. 2809

组织过程成熟度 **organizational process maturity**

组织单元在既定范围内持续实施过程的程度，这些过程有助于实现其（当前或预期的）业务需求。

[来源：ISO/IEC 33001:2015, 3.4.2]

3. 2810

组织过程绩效测量数据 **Organizational process performance measures data**

为已识别的测量需求所提供的数据。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2811

组织过程绩效度量需要 **Organizational process performance measures needs**

确定用于度量组织实现其目标的程度的过程绩效指标。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2812

组织概况 **organizational profile**

过程概要集。

注：这些概要符合与基础、中级和高级概要相对应的组织成熟度等级。

[来源：ISO/IEC TR 29110-3-1:2015, 3.3]

3. 2813

组织级项目管理成熟度 **organizational project management maturity**

组织以可预测、可控制且可靠的方式实现预期战略成果的能力等级。

[来源：项目管理知识体系指南（PMBOK®指南） 第五版]

3. 2814

组织风险 **organizational risk**

阻碍商业价值和产品线目标实现的风险。

[来源：ISO/IEC 26556:2018, 3.1]

3. 2815

组织测试规程 **organizational test practice**

规定组织内部开展测试工作的推荐思路与实施方法，用以明确测试具体执行要求的文件。

条目注释 1：组织测试规程应与组织测试方针保持一致。组织可针对差异化业务场景制定多份测试规程文件，例如分别面向移动应用、安全关键系统编制专用规程。在未单独制定测试方针的情况下，组织测试规程可融入方针相关内容。

[来源：ISO/IEC/IEEE 29119-2:2021, 3.8]

3. 2816

组织级测试过程 **organizational test process**

开发和管理组织级测试规格说明的测试过程。

[来源: GB/T 38634.1-2020, 3.23]

3. 2817

组织级测试规格说明 **organizational test specification**

为一个组织的测试提供信息的文档, 该信息并不针对具体项目。

示例: 组织级测试规格说明最常见的例子是组织级测试方针和组织级测试策略。

[来源: GB/T 38634.1-2020, 3.24]

3. 2818

组织级测试策略 **organizational test strategy**

为组织内所有项目执行测试提供一般要求的文档; 提供有关如何执行测试的细节。

注1: 与组织级测试方针保持一致。

注2: 一个组织可以有多个组织级测试策略以应对不同背景的项目。

[来源: GB/T 38634.1-2020, 第1部分]

3. 2819

原始属性 **origin attribute**

软件按自研开发类或非自研开发类进行分类。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3166]

3. 2820

原始设备制造商 **original equipment manufacturer; OEM**

原始设备制造商。

[来源: ISO/IEC 19770-3:2016, 3.2; INCOSE 系统工程手册 第四版, Appendix B]

3. 2821

原始来源声明 **original source statement**

从外部产品获得的来源声明。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3168]

3. 2822

孤立页面 **orphan page**

网站中没有被本站任何其他页面链接的页面。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.18]

3. 2823

正交可变性模型 **orthogonal variability model; OVM**

正交可变性模型。

[来源: ISO/IEC 26559:2017, 4]

3. 2824

渗透式沟通 **osmotic communication**

通过非正式聆听和非言语暗示等非直接交流的方式获取信息。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3. 2825

结果 **outcome**

状态的显著变化或满足特定约束条件。

示例：要求或目标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3181]

3.2826

外层基数 outer cardinality

从关系中其他参与方的视角，一个参与数据对象所允许的实例数量。

参见：内部基数。

[来源：ISO/IEC 15476-4:2005, 6.6.1]

3.2827

输出 output

由程序产生的产品、成果或服务。

参见：过程结果 (process outcome)。

[来源：ISO/IEC/IEEE 24774:2021, 3.6]

3.2828

(过渡段的)输出弧 output arc (of a transition)

从变迁指向库所的有向弧。

[来源：ISO/IEC 15909-1:2019, 3.22]

3.2829

输出断言 output assertion

指定程序输出为满足程序正确性所需满足的一个或多个条件的逻辑表达式。参见：输入断言、循环断言、归纳断言法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3186]

3.2830

(过渡的)输出位置 output place (of a transition)

通过输出弧与变迁相连的库所。

[来源：ISO/IEC 15909-1:2019, 3.23]

3.2831

输出原语 output primitive

包含源语句、功能点和文档的原语。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3188]

3.2832

输出产品 output product

信息可以采用的，由应用程序发布的一种物理形式。典型的输出产品包括记录、输出文件或者发送给其他应用程序的消息。

[来源：GB/T 42588-2023, 3.1.51]

3.2833

外部视角 outsider's viewpoint

潜在采购方的视角：该采购方既不拥有现有系统，也不拥有其拟替换系统，其评估方案为按残值购置现有系统或购置候选替换系统。

注：便于在系统替换决策中对沉没成本与残值进行合理核算。

[来源：ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3189]

3.2834

外包 outsource

通过协议安排，由外部组织执行本组织的部分职能或流程。

[来源：ISO/IEC 19770-1:2017, 3.41]

3.2835

项目总体风险 overall project risk

所有不确定性来源对项目整体的综合影响，包括单个风险之间的相互作用，以及利益相关方对项目结果波动的暴露程度。

[来源：GB/T 20032-2024, 3.10]

3.2836

过拟合 overfitting

一种机器学习模型的生成状态：模型与训练数据过度契合，导致模型难以对新数据实现泛化。

[来源：ISO/IEC TR 29119-11:2020, 3.1.51]

3.2837

溢出异常 overflow exception

当算术运算的结果超出了指定的存储单元能接受的大小时发生的异常。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3192]

3.2838

开销操作 overhead operation

建立或重新建立一组初始条件以促进计算机程序执行的计算机操作。

3.2839

开销时间 overhead time

计算机系统执行不直接影响任何用户任务进度的任务所花费的时间。

示例：为计费目的而统计计算机资源使用情况所花费的时间。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3193]

3.2840

覆盖 overlay

存储分配技术，在需要时从辅助存储器加载计算机程序段到主存储器，覆盖当前未使用的其他段。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3194]

3.2841

覆盖监督程序 overlay supervisor

控制覆盖顺序和定位的例程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3195]

3.2842

重载 overload

根据程序执行期间任何给定时间与其关联的数据类型，为运算符、标识符或字面量赋予不止一种含义。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3196]

3.2843

重写 override

子类中的属性在保留相同含义的同时重新指定同名继承属性的实现能力。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3197]

3.2844

重写属性 overriding property

子类中的属性，其含义和标识与其某个超类中同名属性的含义和标识相同，但实现方式不同。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3198]

3.2845

所属属性 owned attribute

未迁移到实体中的实体的属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3200]

3.2846

FSM 方法的拥有者 owner of the FSM method

拥有 FSM 方法知识产权的个人或组织。

[来源：ISO/IEC TR 14143-3:2003, 3.7]

3.2847

包 package

〈软件〉由相关数据类型、数据对象和子程序组成的可单独编译的软件组件。

示例：具有一组文件的软件包，可用于在计算设备上安装软件，并可通过 CD 或电子方式分发。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3208]

[来源：〈软件与系统工程〉分组元素的命名空间。]

[来源：ISO/IEC 30130:2016]

3.2848

包图 package diagram

包图由包和包之间的联系构成。包是一种对模型元素进行成组组织的通用机制，它把语义上相近的可能一起变更的模型元素组织在同一个包里，便于理解复杂的系统，控制系统结构各部分间的接缝。包图是维护 and 控制系统总体结构的重要建模工具。

[来源：GB/T 11457-2006, 2.1090]

3.2849

填充 padding

用虚拟字符、字或记录填充固定长度数据块的技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3210]

3.2850

页 page

〈存储分配〉在存储分配中作为单元处理的固定长度数据段或计算机程序段。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3211]

〈虚拟存储系统〉在虚拟存储系统中，具有虚拟地址，并作为整体在主存储器和辅助存储器之间传输的固定长度的数据或计算机程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3211]

〈视频显示〉视频显示终端上的一屏信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3211]

〈网络图〉用于将大型网络图分割成较小部分的结构化机制，这些较小部分也是待打印网络的单元。

[来源：ISO/IEC 15909-2:2011, 4.1.8]

3.2851

页亏 page breakage

当最后一页数据或计算机程序没有填满分配给它的整个存储块时未使用的主存储部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3212]

3. 2852

页帧 page frame

大小为一页，用于保存页的主存储块。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3213]

3. 2853

页面参考 page reference

用于唯一标识一个模型页面的表达式。

注：页面引用包含对关联图表的引用、页面类型，以及区分同一图表下相同类型不同页面所需的序号信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3214]

3. 2854

页交换 page swapping

主存和辅存之间页的交换。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3215]

3. 2855

页表 page table

标识页面在存储中的位置，并给出这些页的重要属性的表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3216]

3. 2856

页面类型信函 page type letter

页面引用中，用于表示某一特定类型模型页面的大写字母。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3217]

3. 2857

0 页 page zero

在存储器分配的调页方法中，页序列中的第一页。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3218]

3. 2858

逐页审阅 page-by-page reviewing

评审人员按先后顺序对工作产品进行评审的技术。

[来源：ISO/IEC 20246:2017, 3.11]

3. 2859

调页程序 pager

启动和控制主存和辅存之间页传输的例程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3220]

3. 2860

分页，调页 paging

块分配 block allocation

存储分配技术，其中程序或数据被分成固定长度的块，称为页，主存储器被分成相同长度的块，称为页帧。页存储在页帧中，不一定连续或按逻辑顺序。并且页根据需要在主存储和辅助存储之间传输。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3221]

3. 2861

结对编程 pair programming

一种典型的实践方式，即一名开发人员编写源代码或配置系统，而另一名开发人员观察其工作，查

找错误并确定完成的工作是否准确且符合要求。

注：两人会定期互换角色，由另一名开发人员编写代码，原编写人员则进行观察。

[来源：ISO/IEC 29110-1-2:2024, , 3.56]

3.2862

结对评审 pair review

由作者以外两名具备相应资质的人员共同开展的，对工作产品的非正式评审。

[来源：ISO/IEC 20246:2017, 3.12]

3.2863

并行 parallel

使用不同设施对整体中的各个部分（例如字符的位）同时进行传输、产生或处理。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3226]

3.2864

并行类 parallel classes

一对不同的类，它们不是互斥的，有一个共同的通用基类，且它们都不是对方的通用基类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3227]

3.2865

并行构造 parallel construct

由两个或多个可同时存在的过程组成的程序构造。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3228]

3.2866

并行运行操作 parallel run operation

给定的信息处理系统和其预期的替代系统，两个系统的运行具有相同的应用程序和源数据，用于比较和置信。

[来源：ISO/IEC 2382:2015, X.X]

3.2867

参数 parameter

在特定应用中赋予值的变量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3230]

3.2868

参数化集合类 parameterized collection class

仅限存储指定类型（类）实例的集合类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3232]

3.2869

参数化高级网络图 parameterized high-level net graph

定义中包含参数的高级网络图。

[来源：ISO/IEC 15909-1:2004, 2.1.19]

3.2870

参数值对 parameter-value pair

将测试项参数与为该参数分配的取值相结合，在组合测试设计技术中作为测试覆盖项使用。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.38]

[来源：同义词：P-V 对]

3. 2871

参数图 parametric diagram

参数图 (par) 表示对系统属性值的约束, 以支持详细的工程分析。这些约束可能包括性能、可靠性和质量属性等。

[来源: INCOSE 系统工程手册 第四版, 9.1.9]

3. 2872

参数估计 parametric estimating

一种估算技术, 其中利用算法根据历史数据和项目参数来计算成本或工期。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3233]

3. 2873

问题登记表 parking lot

延期、未解决问题或关注点登记表。

[来源: ISO/IEC/IEEE FDIS 24765, X.X]

3. 2874

停车场图 parking lot diagram

由尚未开展或尚未完成的任务、用户故事组成的展示性清单。

注: 该清单可按功能分组, 并标注各项的预估优先级及计划开始、完成或处置日期。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3240]

3. 2875

解析 parse

通过将语言单元分解为更基本的子单元并建立子单元之间的关系来确定语言单元的句法结构。

示例: 将块分解为语句, 将语句分解为表达式, 将表达式分解为运算符和操作数。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3241]

3. 2876

解析器 parser

解析计算机程序或其他文本的软件工具, 通常作为汇编、编译、解释或分析的第一步。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3242]

3. 2877

部分的 partial

不完全映射。

注: 即部分实例未映射到任何关联实例。属性可声明为部分的, 表示其可无值; 关系语法中可将参与者属性声明为可选的; 若操作对部分实例无意义 (即无返回结果或不产生响应), 则该操作声明为部分的。

参见: 完全映射、映射完备性、可选的。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3243]

3. 2878

部分簇 partial cluster

父类存在实例, 但该实例不属于其任何子类的子类簇。

参见: 完全簇、父类。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3244]

3. 2879

参与者属性 participant property

一种状态类的属性, 用于体现该类对其实例所参与的关系的认知。

注：当两个状态类之间存在关系时，每个类均包含该关系对应的参与者属性。参与者属性是从一个状态类到关联状态类（不一定是不同类）的映射。每个参与者属性的名称，为关联中另一类所扮演的角色名，或直接为关系另一端类的类名（只要类名不产生歧义即可）。参与者属性的取值为关联实例的标识。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3246]

3.2880

参与式设计 participatory design

一种系统设计过程，旨在通过集体行动中反思的方式，开展调研、理解、反思、确立、演进，并支持多方系统利益相关者与系统开发者之间的相互学习。

注：参与式设计实践的参与者通常承担用户与设计者两大核心角色：设计者力求理解用户场景与需求的实际状况，用户则力求清晰表达自身期望目标，并明确达成目标的合适技术手段。

[来源：IEEE 7000:2021, 3.1]

3.2881

通过 / 失败准则 pass/fail criteria

用于确定软件项或软件特性是否通过或未通过测试的决策规则。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3252]

3.2882

被动互连 passive interconnection

描述两个交互对象之间共享的一个或多个现象的共同解释的互操作性协议。

[来源：IEEE 1175.2-2006]

3.2883

被动对象 passive object

无控制线程的对象。

注：此类对象的操作可由并发对象（即任务）直接或间接调用。

[来源：ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3255]

3.2884

补丁 patch

对源程序或者目标程序的修改。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3256]

3.2885

路径 path

〈软件工程〉在执行计算机程序时执行的一系列指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3257]

2. 〈文件访问〉指定文件存储位置的目录和子目录名称的层次序列。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3257]

3. 〈软件测试〉测试项中可执行语句的序列。

[来源：GB/T 38634.4-2020]

3.2886

路径分析 path analysis

分析计算机程序以识别程序中所有可能的路径，检测不完整的路径，或发现不在任何路径上的程序部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3258]

3. 2887

路径条件 path condition

为执行特定程序路径而必须满足的一组条件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3259]

3. 2888

路径表达式 path expression

表示为执行特定程序路径而必须满足的输入条件的逻辑表达式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3260]

3. 2889

相依耦合 pathological coupling

一个软件模块影响或依赖于另一个软件模块。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3262]

3. 2890

模式敏感故障 pattern-sensitive fault

导致对某种特定数据模式响应失败的故障。

[来源: GB/T 11457-2006, 2.1124]

3. 2891

支付矩阵 payoff matrix

在不确定性决策中, 将一组备选方案的合意度与一组未来状态相关联的表格。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3264; ISO/IEC TR 7052:2023,

3.32]

3. 2892

同行评审 peer review

对由具备相应资质的人员所完成的相同工作内容的工作成果进行的审查。

参见: 伙伴检查、检查、结构化审查、工作成果。

注: 通常在工作成果的开发过程中进行, 以识别并消除缺陷这个其目的是提高工作成果的质量, 并通过尽早修复缺陷来降低成本。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3276]

3. 2893

对等软件 peer software

指与另一软件处于同一层次, 并与其进行数据交换的软件片段。

[来源: ISO/IEC 19761:2011, 2.21]

3. 2894

百分位数 percentile

统计学术语, 如果将一组数据从小到大排序, 并计算相应的累计百分位, 则某一百分位所对应数据的值就称为这一百分位的百分位数。可表示为: 一组 n 个观测值按数值大小排列, 处于 $p\%$ 位置的值称第 p 百分位数。

[来源: GB/T 36964-2018, 1.82]

3. 2895

感知参考点 perceptual reference point

指系统与物理世界之间存在交互作用的参考点。

[来源: ISO/IEC 10746-2:2009, 15.3.2; ISO/IEC/IEEE 24765:2017]

3. 2896

完善性维护 perfective maintenance

对软件产品进行修改，以向用户提供增强功能、改进用户信息，并进行记录，以提升软件性能、维护性或其他软件属性。参见：第 317 条 适应性维护 (adaptive maintenance)，第 1063 条 改正性维护 (corrective maintenance)。

[来源：ISO/IEC/IEEE 14764:2021, 3.1.9]

3. 2897

完善性支持 perfective support

对系统进行在线修改以提供增强功能。

参见：附加支持、自适应支持。

[来源：IEEE 982:2024, , 3.1]

3. 2898

实施整体变更控制 perform integrated change control

对所有变更请求进行审查、批准变更以及管理变更以实现可交付成果的交付这一过程，包括组织过程资产、项目文件以及项目管理计划，并传达其处理结果。

[来源：ISO/IEC/IEEE 24765: 2017, 3.2860]

3. 2899

实施定性风险分析 perform qualitative risk analysis

通过评估并综合风险发生的可能性及其影响程度，来确定其优先级，以便进一步进行分析或采取行动的过程。

[来源：ISO/IEC/IEEE 24765: 2017, 3.2861]

3. 2900

实施质量保证 perform quality assurance

对质量要求进行审核，并依据质量控制测量结果来确保采用适当的质量标准和操作定义。

[来源：ISO/IEC/IEEE 24765:2017, 3.2862]

3. 2901

实施质量控制 perform quality control

监控并记录执行质量活动的结果的过程，以评估绩效并提出必要的改进措施。

[来源：IEEE_CS_SE_Vocabulary-2025]

3. 2902

进行定量分析 perform quantitative analysis

对已识别风险对整体项目目标的影响进行数值分析的过程。

[来源：IEEE_CS_SE_Vocabulary-2025]

3. 2903

进行定量风险分析 perform quantitative risk analysis

对已识别风险对整体项目目标影响的数值分析过程。

[来源：ISO/IEC/IEEE 24765:2017, 3.2863]

3. 2904

性能 performance

表征与流程、功能、活动或任务执行相关的物理或功能属性的定量度量。

注：性能属性包括数量（规模 / 总量）、质量（优劣程度）、及时性（响应性、频率）和就绪性（生效时间、适用条件）。性能可关联活动、流程、产品（含服务）、系统或组织的管理。

[来源: INCOSE SE Handbook v5; ISO/IEC/IEEE FDIS 24765, 3.3282]

3.2905

性能分析 performance analysis

对在一个给定硬件配置上运行的实时系统（或软件设计）进行的定量分析，该系统还受到给定外部工作负载的影响。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3283]

3.2906

性能基线 performance baseline

在不执行干扰注入的情况下对系统正常执行性能工作负载的结果。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3284]

3.2907

性能度量基线 performance measurement baseline; PMB

用于进行比较的综合范围、进度和成本基准，用于管理、衡量和控制项目执行。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3287]

3.2908

性能报表 performance reports

绩效报告

[输出/输入] 文件和演示文稿，提供有组织 and 总结的工作绩效信息、挣值管理参数和计算，以及项目工作进度和状态分析。

[来源: PMBOK Guide, 7th Edition]

3.2909

性能需求 performance requirement

可衡量的标准，用于确定功能的质量属性或衡量功能需求的完成程度。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3289]

3.2910

绩效评估 performance reviews

一种用于对项目中正在进行的工作的实际执行情况与基准进行测量、比较和分析的技术。

[来源: ISO/IEC/IEEE 24765:2017, 3.2871]

3.2911

性能规格 performance specification

用于标识系统或组件性能特征的信息项。

注：通常为需求规格说明的一部分这些特征通常包括速度、准确性和内存使用量。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3290]

3.2912

执行过程 performed process

完成生产工作产品所需工作的过程。

注：满足了该过程领域特定的目标。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3292]

3.2913

周期性任务 periodic task

由定时器事件按固定时间间隔激活执行的任务。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE FDIS 24765, 3.3295]

3. 2914

外围设备 peripheral equipment

由某一特定计算机控制并可与其进行通信的设备。

[来源: ISO/IEC 2382:2015, X. X]

3. 2915

永久性 permanence

故障对已完成事务所引发的对象状态变更产生影响的程度。

[来源: ISO/IEC 10746-3:2009, 13.7.1.5]

3. 2916

权限 permission

允许特定行为发生的许可。

[来源: ISO/IEC 10746-2:2009, 11.2.5]

3. 2917

永久许可证 perpetual license

永久授予软件权利的许可证。

注: 永久许可的替代方案是一个期限或基于订阅的许可证。

[来源: ISO/IEC 19770-3:2016, 3.1.18]

3. 2918

持久性 persistence

对象在契约上下文或时期发生变更时仍持续存在的属性。

[来源: ISO/IEC 10746-2:2009, 11.3.1]

3. 2919

持久性模式 persistence schema

使用特定处理、存储和通信功能的约束规范。

[来源: ISO/IEC 10746-3:2009, 16.5.1.1]

3. 2920

持久性对象 persistent object

在创建它的进程或线程不复存在时依然存在的对象。

3. 2921

持久存储器 persistent storage

允许一个功能过程存储的数据超出其生存周期; 和/或允许功能过程提取其它功能程序所存储的数据, 相同功能过程早期存储的数据, 或者其它过程存储的数据。

注 1: 因为持久存储器在边界的软件一端, 因此不被认为是被测软件的功能用户。

注 2: 本文中的“其它过程”可能是只读内存。

[来源: GB/T 42452-2023]

3. 2922

持久化统一资源标识符 persistent URI

一种无需修改文档中的链接, 即便目标对象的位置发生变更, 仍可访问到该对象的引用。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.19]

3. 2923

功能点耗时率 person hours per functional size unit

每功能点所消耗的人时数。

[来源: GB/T 36964—2018, 1.83]

3.2924

角色 **persona**

用户类型的表征, 包含对用户特征与行为的简明概括, 这些特征与行为对设计最具参考价值或能体现特定用户需求。

注: 人物角色通常包含行为模式、目标、技能、态度和环境, 并辅以若干虚构的个人细节以塑造真实可信的角色形象。人物角色能够代表更大群体在目标、期望和个人特征方面的需求, 有助于指导系统设计决策及确定设计目标。

[来源: ISO/IEC 29110-5-1-2:2025, 3.17]

3.2925

个人计算机 **personal computer; PC**

主要供个人独立使用的微型计算机。

[来源: ISO/IEC 2382:2015, X.X]

3.2926

个人计算机 (PC) **personal computer (PC)**

主要用于个人独立使用的微型计算机。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3306]

3.2927

个人数据 **personal data**

符合以下任一条件的信息: (a) 可用于识别该信息所关联的个人身份信息主体; 或 (b) 已直接或间接关联至个人身份信息主体, 或可被关联。

示例: 数据因关联标识符 (如姓名、识别码、位置数据) 或因涉及该自然人身体、生理、遗传、心理、经济、文化或社会身份的特定因素而具有个人属性。

注: 在某些司法管辖区, 此为法律术语; 其他司法管辖区则使用个人信息等替代表述。个人身份信息的范围可能因数据来源地或处理地适用的法律法规而异。

[来源: ISO/IEC TS 25052-1:2022, 3.3.10]

3.2928

个人防护装备 **personal protective equipment; PPE**

为个人佩戴或手持而设计, 以防范一种或多种健康和安全风险的设备或器具。

[来源: IEC/IEEE 82079-1:2019, 3.26]

3.2929

个人信息 **personally identifiable information; PII**

满足以下条件的信息: (a) 可用于识别该信息所关联的个人身份信息主体 (PII 主体), 或 (b) 直接或间接与 PII 主体相关联或可关联。

[来源: ISO/IEC TS 25052-1:2022, 3.3.10]

3.2930

基于视角的阅读 **perspective-based reading**

一种基于角色的评审形式: 采用检查清单, 并通过创建可交付成果原型, 来检查工作产品的完整性及其他质量特性。

[来源: ISO/IEC 20246:2017, 3.15]

3.2931

悲观持续时间 **pessimistic duration**

对最长活动持续时间的估计，该估计考虑了可能影响性能的已知变量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3315]

3.2932

Petri 网 Petri net; PN

由库所和变迁这两个集合及其关联关系与函数构成的代数结构，以发明者 Carl Adam Petri 命名。

注：Petri 网通常以图的形式表示，包含用弧连接的两类节点（称为库所和变迁），以及指示动态属性的标记（称为令牌）。

[来源：ISO/IEC 15909-1:2019, 3.25]

3.2933

阶段 phase

生存周期中的一个时间段，在此期间开展相关活动，以实现该阶段的目标。

参见：阶段（stage）。

[来源：ISO/IEC/IEEE 42020:2019, 3.15]

3.2934

相位门 phase gate

在某一阶段结束时进行审查，此时会决定是否继续进入下一阶段、是否进行修改继续进行、或者是否终止项目或计划。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3321]

3.2935

物理架构 physical architecture

物理架构描述了物理系统元素之间的关系，这些元素包括硬件、软件、数据、人员和程序。

[INCOSE 系统工程手册 第四版, 9.4.2.4]。

3.2936

物理配置审计 physical configuration audit; PCA

验证已建立的某个配置项遵循定义它的技术文档的审核行为。

参见：功能配置审核（functional configuration audit）。

[来源：IEEE 828-2012, 2.1 ;GB/T 11457-2006]

3.2937

物理配置审计 (PCA) physical configuration audit (PCA)

进行审计以核实一个配置项在实际构建时是否符合定义该配置项的技术文档。

注：已修改，“确保”一词改为“帮助确保”对于软件而言，软件物理配置审计（PCA）的目的在于帮助确保设计和参考文档与实际建成的软件产品保持一致。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3323]

3.2938

物理接口 physical interface

两个或多个系统元素之间的物理连接。

注：物理接口可使用固体、液体、气体、真空、热、电磁场或其他方式。

[来源：ISO/IEC/IEEE 24748-6:2023, 3.1.6]

3.2939

物理需求 physical requirement

该要求规定了系统或系统组件的一个具体且可衡量的特性。

参见：设计要求、功能要求、实施要求、接口要求、性能要求。

示例：材料、形状、尺寸、重量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3325]

3.2940

物理源语句 `physical source statement`; PSS

被视为一行代码的源语句。

参见：逻辑源语句。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3326]

3.2941

流水线 `pipeline`

〈软件过程〉将软件开发和运维过程按照合理的内在逻辑和关系拆分为若干专注于特定任务和目标的子活动，进而选择并组合子活动以实现特定的业务目标。流水线的自动化和持续流动是开发运维一体化的关键。

[来源：GB/T 42560-2023, 3.12]

〈软件设计〉软件或硬件设计技术，其中一个过程的输出作为第二个过程的输入，第二个过程的输出作为第三个过程的输入，依此类推，通常在单个周期时间内实现并行处理。

[来源：IEEE 2675-2021, 3.1]

3.2942

节点 `place`

Petri 网的节点，通常用椭圆表示。

[来源：ISO/IEC 15909-1:2019, 3.28]

3.2943

计划 `plan`

为达成既定目标而给出系统性行动方案的信息项，包含各项具体活动的执行时间、执行方式与执行主体。

[来源：ISO/IEC/IEEE 15289:2019, 5.16]

3.2944

计划沟通 `plan communications`

计划沟通管理 `plan communications management`

确定项目利益相关者信息需求以及制定沟通方式的过程。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.2945

计划成本管理 `plan cost management`

该过程用于制定项目成本的规划、管理、支出和控制方面的政策、程序及文件。

[来源：ISO/IEC/IEEE 24765: 2017, 3.2915]

3.2946

计划采购 `plan procurement`

计划采购管理 `plan procurement management`

记录项目采购决策的过程、明确处理方式以及确定潜在供应商的步骤。

[来源：IEEE_CS_SE_Vocabulary-2025]

3.2947

计划质量 `plan quality`

计划质量管理 `plan quality management`

确定项目及产品的质量要求和/或标准的过程，并记录项目将如何证明其符合性。

[来源：IEEE_CS_SE_Vocabulary-2025]

3. 2948

规划资源管理 plan resource management

定义如何估算、获取、管理和利用物理资源和团队资源的过程。

[来源：IEEE_CS_SE_Vocabulary-2021]

3. 2949

计划风险管理 plan risk management

确定如何开展项目风险管理活动的这一过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3340]

3. 2950

计划风险应对 plan risk responses

制定方案和措施的过程，旨在增加项目目标的机会并减少对其构成威胁的因素。

[来源：ISO/IEC/IEEE 24765:2017, 3.2920]

3. 2951

计划进度管理 plan schedule management

制定项目进度计划的规划、开发、管理、执行和控制所涉及的政策、程序及文件的过程。

[来源：ISO/IEC/IEEE 24765:2017, 3.2921]

3. 2952

计划范围管理 plan scope management

制定项目范围管理计划的过程，该计划将详细说明项目范围的定义、验证和控制方式。

[来源：ISO/IEC/IEEE 24765: 2017, 3.2922]

3. 2953

规划干系人参与 plan stakeholder engagement

根据项目利益相关者的需求、期望、利益和对项目的潜在影响，制定让项目利益相关者参与的方法的过程。

[来源：IEEE_CS_SE_Vocabulary-2021]

3. 2954

规划干系人管理 plan stakeholder management

在项目整个生存周期中，根据对利益相关者的需求、兴趣以及其对项目成功可能产生的影响的分析，制定出适当的管理策略，以有效与他们进行沟通。

[来源：ISO/IEC/IEEE 24765: 2017, 3.2923]

3. 2955

计划标准 plan standard

描述实现既定目标或在特定资源范围内开展工作所采用方案的特征的标准。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3341]

3. 2956

计划过程 planned process

这种过程是通过描述和计划两方面来记录的。

注：相关的过程描述和计划是相互协调的，该计划可以包括标准、要求、目标、资源和任务分配。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3342]

3. 2957

计划值 **planned value; PV**

分配给计划工作的授权预算。

[来源：项目管理知识体系指南(PMBOK® 指南) — 第七版]]

3. 2958

计划价值 **planned value; PV; BCWS**

计划工作的预算成本 **budgeted cost of work scheduled**

分配给预定工作的授权预算。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3343]

3. 2959

规划 **planning**

与计划规范有关的活动。

[来源：ISO/IEC/IEEE 24748-5:2017, 3.9]

3. 2960

规划范围 **planning horizon**

用于比较不同方案成本的固定时间段。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3345]

3. 2961

计划绩效域 **planning performance domain**

聚焦项目可交付成果与结果交付所需的初始、持续及动态组织协调活动和职能的绩效领域。

[来源：项目管理知识体系指南(PMBOK® 指南) — 第七版]]

3. 2962

计划扑克 **planning poker**

基于团队的估算方法，通过共识设定开发和测试每个用户故事所需工作量的相对估算值。

注 1：该方法使用通常基于斐波那契数列（即 2、3、5、8、13）的卡片。

注 2：该方法通过团队成员匿名出牌（数字卡）达成共识，避免锚定偏差。

[来源：ISO/IEC 29110-1-2:2024, 3.57]

[来源：ISO/IEC/IEEE FDIS 24765, 3.3347]

3. 2963

规划过程组 **planning process group**

这些过程旨在确定项目的范围、明确目标，并确定为实现该项目旨在达成的目标而必须采取的行动步骤。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3348]

3. 2964

在特定场所对不同系统进行的组装 **plant**

在特定场所对不同系统进行的组装。

[来源：IEC/IEEE 82079-1:2019; 3.27]

3. 2965

塑料有引线芯片载体 **plastic leaded chip carrier; PLCC**

矩形芯片单元，由塑料制成，用于低功耗设备，通常配备表面贴装或 J 形（J 引线）连接器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3350]

3. 2966

塑料引线芯片载体 plastic leaded chip carrier (PLCC); PLCC

矩形芯片单元，由塑料制成，用于低热设备，通常带有表面贴装或 J 形（J 型引线）连接器。

[来源：ISO/IEC/IEEE 24765c:2014]

3. 2967

平台 platform

操作系统和硬件的组合，构成程序运行的操作系统环境。

[来源：ISO/IEC/IEEE 26513:2017, 3.30]

3. 2968

平台即服务 platform as a service; PaaS

提供完整的 IT 资源环境，如服务提供商支持的编程语言、库、服务和工具。

注：服务提供商对提供给客户的服务控制程度可能因服务提供商而异。在 DevOps 中，平台即服务（PaaS）作为 DevOps 流水线的一部分实现了自动化。

[来源：IEEE 2675-2021, 3.1]

[来源：另见基础设施即服务、软件即服务。]

3. 2969

平台即服务 (PaaS) platform as a service (PaaS); PaaS

(1) 提供完整的 IT 资源环境，如服务提供商支持的编程语言、库、服务和工具。

注意：对提供给客户的服务的控制级别可能因服务提供商而异。在 DevOps 中，PaaS 作为 DevOps 管道的一部分被自动化。

参见：IaaS、SaaS。

[来源：IEEE 2675-2021, 3.1]

3. 2970

PNML 高级网络文档 PNML high-level net document

PNML 高层网文档

PNML 包含一个或多个网图的文档，其中所有网图都符合要求。

包含一个或多个网图、且所有网图均符合高层 Petri 网规范的 PNML 文档。

[来源：ISO/IEC 15909-2:2011, 4.1.11]

3. 2971

PNML 对称网文档 PNML symmetric net document

包含一个或多个网图，且所有网图均符合对称网规范的 PNML 文档。

[来源：ISO/IEC 15909-2:2011]

3. 2972

要点设计 point design

选择一种满足要求的设计而不检查其他可能更有效的设计。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3371]

3. 2973

控制与观察点 point of control and observation; PCO

控制与观察点。

[来源：ISO/IEC 10746-1:1998; ISO/IEC/IEEE FDIS 24765, 3.3270]

3. 2974

观点 point.

垂直距离测量

垂直距离计量单位；1 毫米约等于 28 点（1 英寸等于 72 点）。

[来源：ISO/IEC/IE[EE 24765a:2011；ISO/IEC/IEEE FDIS 24765]]

3. 2975

指针 **pointer**

规定另一数据项位置的一种数据项。

示例：指定要处理的下一个员工记录地址的数据项。

[来源：ISO/IEC/IEEE FDIS 24765, 3372)

3. 2976

政策与程序 **policies and procedures; P&P**

政策与程序。

[来源：ISO/IEC/IEEE 24765k:2022；ISO/IEC/IEEE FDIS 24765, 3. 3204]

[来源：同义词：P&P]

3. 2977

策略设置行为 **policy-setting behaviour**

规范中定义的行为，通过该行为可以更改策略。

[来源：ISO/IEC 10746-2:2009, 11. 2. 12]

3. 2978

PESTEL 分析 **political, economic, social, technological, environmental, and legal; PESTEL**

政治、经济、社会、技术、环境与法律。

[来源：ISO/IEC/IEEE 24765f:2016]

3. 2979

弹出窗口 **pop-up**

用户操作调用时显示的嵌入式上下文相关信息。

[来源：ISO/IEC/IEEE 24765g:2018；ISO/IEC/IEEE FDIS 24765]

3. 2980

端口 **port**

原子模型的一类输入输出数据或事件的通道。

注：它由一个或多个端口项组成，端口在原子模型内部具有唯一标识。

[来源：GB/T 36457-2018]

3. 2981

可移植性 **portability**

可运输性 **transportability**

程序在各种类型的数据处理系统上执行的能力，无需将程序转换为不同的语言并且只需很少或无需修改。

3. 2982 [[来源：ISO/IEC 2382:2015, X. X]

项目组合管理 **portfolio management; PM**

集中管理一个或多个项目组合以实现战略目标。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3386]

3. 2983

项目组合管理计划 **Portfolio management plan**

定义项目组合所需的方法、时间表、资源和特定注意事项。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2984

投资组合管理记录 Portfolio management record

与投资组合管理相关的永久、可读形式的数据、信息或知识。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2985

投资组合管理报告 Portfolio management report

为相关方准备的账户，用于传达项目组合管理活动的状态、结果和结果。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 2986

文件夹 portfolio.

将项目、计划、子项目组合和运营作为一个整体进行管理，以实现战略目标。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3. 2987

项目后评估 post project evaluation

在项目已经完成并运行一段时间后，对项目的目的、执行过程、效益、作用和影响进行系统、客观分析和总结的一种技术经济活动。

[来源：GB/T 25000.2-2018]

3. 2988

后同步码断点 postamble breakpoint

直至从给定的程序或例程退出时才启动的断点。同义词：2.1152 条 后同步断点（postamble breakpoint）。与 2.1268 条 序断点（prolog breakpoint）相对。

参见：2.220 条 代码断点（code breakpoint）、2.391 条 数据断点（data breakpoint）和 2.516 条 动态断点（dynamic breakpoint）、2.1249 条 可编程断点（programmable breakpoint）、2.1580 条 静态断点（static breakpoint）。

3. 2989

关闭后的活动 post-closure activities

软件系统被其客户正式接受后发生的活动。

注：这些活动包括但不限于经验教训审查和归档项目材料。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3388]

3. 2990

编译后时间 post-compile time

组件编译后的链接时间和加载时间的统称。

[来源：ISO/IEC 26557:2016, 3.9]

3. 2991

后置条件 postcondition

一种在某一操作完成之时必须为真的约束。

[来源：GB/T 11457-2006, 2.1150]

3. 2992

开发后审查 post-development review

在系统运转使用已经达到稳定状态之后，对该系统效果的一种评论。

[来源：GB/T 11457—2006, 2.1155]

3.2993

事后转储 postmortem dump

在计算机程序异常终止后产生的转储。

参见：2.196 条 变更转储 (change dump)、2.519 条 动态转输 (dynamic dump)、2.928 条 内存转储 (memory dump)、2.1425 条 选择的转储 (selective dump)、2.1466 条 快照转储 (snapshot dump)、2.1582 条 静态的转储 (static dump)。

[来源：GB/T 11457—2006, 2.1153]

3.2994

探测后 post-probe

准备应对挑战的行动计划的可选阶段。

[来源：ISO/IEC 26561:2019, 3.3]

3.2995

后处理器 postprocessor

在主要过程完成以后执行某些最后的处理步骤的计算机程序。

示例：为了输出，重新格式数据的例程。与 2.1165 条 预处理器 (preprocessor) 相对。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3393]

3.2996

测试后代代 post-tested iteration

单入口、单出口循环，在此循环中，控制在循环体后执行。同义词：2.1151 条 后测试迭代 (post-tested iteration)。与 2.215 条 闭合循环 (closed loop) 和 2.1847 条 “当” 语句 (WHILE) 相对。

参见：2.1756 条 尾随判定 (trailing decision)。图 17 UNTIL 结构。

[来源：GB/T 11457—2006, 2.1151]

3.2997

实践，惯例 practice

用于描述对于软件开发过程的约束的统一的的需求。

参见：约定 (conventions) 和标准 (standards)。

[来源：GB/T 11457—2006]

3.2998

做法 practices

用于描述对于软件开发过程的约束的统一的的需求。

参见：2.347 条 约定 (conventions) 和 2.1562 条 标准 (standards)。

[来源：GB/T 11457—2006, 2.1156]

3.2999

专业人员 practitioner; PT

在一个或多个过程域内执行活动的人员或团队。

[来源：ISO/IEC 29110-1-2:2024, I523.58]

3.3000

预防原则 precautionary principle

预防原则是指，在新产品或新过程的最终影响尚不明确或存在争议时，应抵制其引入，直到相关风险得到适当缓解的原则。

注：预防原则意味着在有能力的情况下有义务防止伤害，即使并非所有证据都已齐。该原则已在多项国际条约和国

际法中得到编纂。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.30]

3.3001

紧前关系绘图法 precedence diagramming method; PDM; AON

单代号网络图 activity-on-node

一种用于构建进度模型的技术，其中活动由节点表示，并通过一个或多个逻辑关系以图形方式链接，以显示活动执行的顺序。

[来源：ISO/IEC/IEEE 24765:2017, 3.2984]

3.3002

优先关系 precedence relationship

逻辑关系优先图法中使用的术语。然而，在当前的使用中，无论使用何种图表方法，优先关系、逻辑关系和依赖关系都被广泛互换使用。

3.3003

预编译程序 precompiler

一种计算机程序，它对源代码（其中有一部分可能是编译程序无法接受的）进行预处理，以产生与之等效但可为编译程序接受的代码。

参见：预处理器（preprocessor）。

示例：把结构化 FORTRAN 转换为 GB 3507 FORTRAN 的预处理程序。

3.3004

前置条件 precondition

在发起属性请求之前必须为真的条件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3409]

3.3005

断言 predicate

求值为 TRUE 或 FALSE 的逻辑表达式，通常用于指示代码中的执行路径。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.40]

3.3006

谓词数据使用 predicate data use; p-use

与决策语句的谓词部分的决策结果相关的数据使用。 [来源：ISO/IEC/IEEE 29119-4:2015]。

3.3007

预测 prediction.

为给定输入生成预测目标值的机器学习函数。

[来源：ISO/IEC TR 29119-11:2020, 3.1.56]

3.3008

预测方法 predictive approach

一种开发方法，即在项目生存周期的早期阶段确定项目范围、时间和成本。

[来源：项目管理知识体系指南(PMBOK® 指南) — 第七版]]

3.3009

预测生存周期 predictive life cycle

项目生存周期的一种形式，其中项目范围以及交付该范围所需的时间和成本尽可能在生存周期的早期确定。

[来源：ISO/IEC/IEEE 24765:2017, 3.2992]

3.3010

预测指标 predictive metric

在开发过程中应用的指标，用于预测软件质量因素的值。

[来源：ISO/IEC/IEEE 24765:2017, 3.2993]

3.3011

预测指标值 predictive metric value

与系统开发期间要满足的质量因素相关的数值目标。

注：这是一个中间要求，是最终目标的早期指标系统性能。例如，设计或代码错误可能是最终系统可靠性的早期预测指标。

[来源：ISO/IEC/IEEE 24765:2017, 3.2994]

3.3012

初步设计 preliminary design

对于系统或部件分析各种设计方案和定义软件体系结构、部件、接口和提出时间和规模方面的估计的过程。

参见：详细设计。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3415]

3.3013

概要设计评审 preliminary design review; PDR

对一个或多个配置项的选定设计方法进行审查，以评估其进度、技术充分性和风险解决情况；确定每个设计与配置项要求的兼容性；评估所选制造方法和工艺的明确程度，并评估其相关技术风险；确定配置项与其他设备、设施、软件和人员之间的物理和功能接口的存在性和兼容性；并在适用的情况下，评估初步操作和支持文件。

参见：关键设计评审、系统设计评审。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3416]

3.3014

初步接口定义 Preliminary interface definition

内部接口(组成系统的系统元素之间)和外部接口(系统的系统元素与感兴趣的系统外部元素之间)的初步逻辑和物理方面。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3015

初步的生存周期概念 Preliminary life cycle concepts

以生存周期概念文档的形式初步阐明与业务需求一致的各种生存周期概念，感兴趣的系统基于、评估和选择这些生存周期概念文档。该体系结构基于这些概念，它们对于为正确解释系统需求提供上下文至关重要。典型概念包括：

- 采办概念
- 部署概念
- OpsCon
- 支持概念
- 退役概念。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3016

初步 MOE 数据 Preliminary MOE data

为确定的测量需求提供的初步数据。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3017

教育部的初步需求 Preliminary MOE needs

初步确定 MOE (Roedler 和 Jones, 2006 年)，它定义了决策者在系统有效性方面的信息需求，以

满足运营预期。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3018

初步 TPM 数据 Preliminary TPM data

为确定的测量需求提供的初步数据。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3019

初步的 TPM 需求 Preliminary TPM needs

TPM 的初步识别 (Roedler 和 Jones, 2006 年)，它测量系统元素的属性以确定系统或系统元素满足或预期满足技术要求或目标的程度。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3020

初步验证标准 Preliminary validation criteria

初步验证标准（要评估的措施）、执行验证活动的人员以及相关系统的验证环境。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.3021

准备时间 preparation time

任务提交前经过的时间。

注：开始准备时间的事件取决于以下任务的任务模式定义。

[来源：ISO/IEC 14756:1999]

3.3022

预探测 pre-probe

阶段了解组织的基本背景，例如当前结构、术语、产品成熟度水平、实施和文档。

用于了解组织基本背景的阶段，包括现有架构、术语体系、产品成熟度等级、实施情况及文档体系。

[来源：ISO/IEC 26561:2019，3.4]

3.3023

预处理 pre-processing

(1) 机器学习 (ML) 工作流的一部分，将原始数据转换为可供 ML 算法使用以创建 ML 模型的状态。

注：预处理可以包括分析、归一化、过滤、重新格式化、插补、去除异常值和重复项，以及评估数据集的完整性。

[来源：ISO/IEC TR 29119-11:2020，3.1.57]

3.3024

预处理器 preprocessor

在主要过程之前，执行某些处理步骤的计算机程序或例程。

示例：预编译程序或为了处理重新格式代码和数据的例程。与 2.1154 条 后处理器 (postprocessor) 相对。

[来源：GB/T 11457-2006，2.1165]

3.3025

建立规则的运作 prescription

建立规则的动作。

[来源：ISO/IEC 15414:2015，6.6.3]

3.3026

可检索查看 presentable

可检索查看。

[来源：ISO/IEC/IEEE 15289:2019，5.18]

3.3027

演示设备 presentation device

用于向系统的预期用户呈现数据的设备。

[来源：ISO/IEC 25024:2015]

3.3028

预存储 prestore

在程序或例程进入之前，存储计算机程序或例程要求的数据。

3.3029

预测试迭代 pretested iteration; WHILE

一种单入口、单出口的循环控制语句，其中循环控制在循环体之前执行。

参见：XXX 条 先导判定 (leading desision)。

同义词：XXX 条 预测试迭代 (pre-tested iteration)。XXX 条 闭循环 (closed loop) 和 XXX 条 “直至” 语句 (UNTIL) 相对。

[来源：GB/T 11457-2006 2.1847]

3.3030

预打印 prettyprinting

使用缩排、空行和其他可视的线索以显示程序的逻辑结构。

3.3031

预防措施 preventive action

为消除潜在不符合项或其他不良潜在情况的原因而采取的措施。

参见纠正措施。

注：预防措施是为了防止发生并保持资产的功能而采取的行动，而纠正措施是为了防止再次发生而采取的行动。预防措施通常在资产功能可用且可操作时进行，或在功能失效发生之前进行。预防措施包括补充消耗品，其中消耗是功能要求的一部分。

[来源：ISO/IEC 19770-1:2017，3.45]

3.3032

预防性维护 preventive maintenance

在问题发生之前，为防止问题发生所进行的维护。

3.3033

预防性支持 preventive support

在潜在故障或漏洞显现为故障之前，采取在线措施进行纠正，以补充或更新资源，或恢复初始运行条件，从而防止软件老化故障。

[来源：IEEE 982:2024, 3.1]

3.3034

以前开发的软件 previously developed software

在准备计划的项目之前或独立于项目制作的软件，包括从外部来源获得或购买的软件。

[来源：ISO/IEC/IEEE FDIS 24765，3.3428]

3.3035

主要数据 primary data

组织所持有的、描述实体的数据，这些实体对企业而言既独立又基础，企业开展业务交易时需要对

其进行引用。

注：主数据（master data）这一术语已不推荐使用。

[来源：ISO/IEC 25024:2015, 4.25]

3.3036

主授权标签 primary entitlement

包含基本信息的授权标签。

[来源：GB/T 36329-2018]

3.3037

主授权模式 primary entitlement schema

权利模式（Ent），封装了权利的基本信息。

注：主要实体具有 Initial、Consolidation、AllocationReceived 或 TransferReceived 的 <entType>。这些是基础 Ents，允许将 Ent 初始填充到 Ent 库中（Consolidation 除外，如果需要，它可用于替换以前的几个 Ents）。其他 Ents（称为补充）。

[来源：ISO/IEC 19770-3:2016, 3.1.19]

3.3038

主实体类型 primary entity-type

在 MkII 功能点分析中涉及的、一种起主要作用的实体类型，所包含的属性是应用程序要专门设计来处理 and/或存储的。

[来源：ISO/IEC 20968:2002, 10]

3.3039

主信息结构 primary information structure

补充信息结构可能链接到的信息结构。

[来源：ISO/IEC 19770-5:2015, 3.24]

3.3040

主要意图 primary intent

最重要的意图。

[来源：ISO/IEC 20926:2009, 3.43]

3.3041

主键 primary key

被选为实体唯一标识符的候选键。

[来源：IEEE 1320.2-1998(R2004), 3.1.149]

3.3042

主要用户 primary user

与系统交互以达到主要目标的用户。

[来源：GB/T 25000.10-2016, 3.11]

3.3043

原始的 primitive

收集数据的最低级别。

示例：错误、故障、故障、时间、时间间隔、日期、非注释源代码语句的数量、边和节点。

注：基元是直接可测量或可计数的，或者可以为特定度量赋予常数值或条件。

[来源：IEEE_CS_SEVocabulary-2025; ISO/IEC/IEEE FDIS 24765]

3.3044

原始类型 primitive type

一种数据类型，它的每一个成员由单个不可分解的数据项构成。同义词：2.1170 条 原语类型 (primitive type)。与 2.268 条 复合类型 (composite type) 相对。

3.3045

印制电路板 printed circuit board; PCB

电子元件之间的连接配置，通常使用蚀刻在非导电基板（电路板）上的金属线路。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3438]

3.3046

印刷电路板 (PCB) printed circuit board (PCB)

电子元件之间的连接配置，通常使用蚀刻在非导电基板（板）上的金属路径。

[来源：ISO/IEC/IEEE FDIS 24765]

3.3047

印制电路板组件 printed circuit board assembly; PCBA

带有嵌入式元器件的印制电路板。

[来源：ISO/IEC/IEEE 24765d:2015]

3.3048

印刷电路板组装 (PCBA) printed circuit board assembly (PCBA)

带有嵌入式元器件的印刷电路板。

[来源：ISO/IEC/IEEE FDIS 24765]

3.3049

印刷文件 printed documentation

以印刷形式提供或以电子形式提供给客户或用户打印的文档。

参见：嵌入式文档。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3440]

3.3050

为用户打印的信息 printed information for users

以印刷形式提供或以电子形式提供给客户或用户打印的用户信息。

参见：用户的嵌入式信息，用户的屏幕信息。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.38]

3.3051

优先级矩阵 prioritization matrices

一种质量管理规划工具，用于识别关键问题并评估合适的备选方案以定义一组实施优先级。

[来源：ISO/IEC/IEEE 24765:2017, 3.3017]

3.3052

优先级方案 prioritization scheme

用于对项目组合、项目集或项目组成部分，以及需求、风险、特性或其他产品信息进行优先级排序的方法。

[来源：项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3053

优先级中断 priority interrupt

允许比当前正在执行的进程有更高优先级的进程执行的一种中断。

[来源：GB/T 11457—2006, 2.1172]

3.3054

优先级反转 priority inversion

由于低优先级任务阻塞，导致任务执行延迟的情况。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3445]

3.3055

隐私功能 privacy capabilities

以公平合法的方式处理个人数据的能力。

[来源：IEEE 7002:2022, 3.1]

3.3056

隐私控制 privacy control

为信息系统或组织规定的监控，旨在实现隐私目标[来源：IEEE 7002:2022]。

3.3057

隐私框架 privacy framework

帮助组织满足其隐私要求的结构，包括政策和程序、角色和职责、培训和治理功能。

[来源：IEEE 7002:2022]

3.3058

隐私原则 privacy principles

指导系统或组织的个人数据操作的一组值。

[来源：IEEE 7002:2022, 3.1]

3.3059

隐私要求 privacy requirement

处理个人数据以满足利益相关者期望的隐私结果和义务的规范。

[来源：IEEE 7002:2022, 3.1]

3.3060

隐私风险 privacy risk

个人或组织因暴露、使用、盗窃或更改个人数据而遇到问题的可能性和影响的组合。

[来源：IEEE 7002:2022, 3.1]

3.3061

私有 private

2. 仅在一个单独例程或模块内已知。

参见：受保护的（protected）、公开的（public）、隐藏的（hidden）。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3062

私有类型 private type

一种数据类型，它的结构和可能的值被定义但不透露给类型的用户。

参见：2.751 条 信息隐藏（information hiding）。

[来源：GB/T 11457—2006, 2.1173]

3.3063

特权指令 privileged instruction

只有管理程序能执行的计算机指令。

3. 3064

特权状态 privileged state

在计算机系统的操作中，管理程序执行的状态。此状态，通常比应用程序的执行状态有更高的优先权。同义词：2.585 条 执行状态 (execution state)、2.917 条 主状态 (master state) 和 2.1175 条 特权状态 (privileged state)。与 2.1178 条 问题状态 (problem state) 相对。

3. 3065

主动的方法 proactive approach

根据组织预测开发创新产品线或产品变体的方法，这些预测预测了规定的产品需求。

[来源：ISO/IEC 26553:2018, 3.15]

3. 3066

概率估算 probabilistic estimating

一种用于生成一系列估算值以及该范围内相关概率的方法。

[来源：Guide to the Project Management Body of Knowledge (PMBOK® 指南) — 第七版]

3. 3067

概率系统 probabilistic system

其行为用概率描述的系统，因此无法完美预测其输出。

[来源：ISO/IEC TR 29119-11:2020, 3.1.58]

3. 3068

概率 probability

事件可能发生的程度。

在数学上，是赋予随机事件、介于 0 到 1 之间的实数，与事件在长期试验中发生的相对频率相关，或与某一事件将会发生的置信程度相关。

注：置信程度越高，概率越接近 1。在描述风险时可使用频率而非概率。对概率的置信程度可划分为等级或层级，例如：罕见 / 不太可能 / 中等 / 可能 / 几乎肯定，或难以置信 / 极不可能 / 渺茫 / 偶尔 / 很可能 / 频繁。

[来源：ISO/IEC/IEEE 24765j:2021; ISO/IEC/IEEE 24765 FDIS, 3.3459]

3. 3069

概率和影响矩阵 probability and impact matrix

绘制每个风险发生概率及其对项目目标影响的网格。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3460]

3. 3070

探查 probe

一种用于调查组织在产品线工程和管理上准备程度或取得成功能力的诊断过程。

参见：产品线技术探查 (product line technical probe)，技术探查 (technical probe)。

[来源：ISO/IEC 26561:2019, 3.7]

3. 3071

问题 problem

需要调查和改正的困难、不确定性或其他的认识到的非预期的事件、事件集、条件或状况。

注：当风险指标（一种客观度量）超过预定阈值（问题触发器）时，风险因素即转变为问题。在创建问题记录时，根本原因通常尚不明确。问题可能涉及服务、产品或流程（步骤），或组织的其他要素。

[来源：ISO/IEC/IEEE 15288:2023 3.26]

3. 3072

问题定义 problem definition

问题描述 problem description

问题陈述，可以包括对解决问题的数据、方法、程序和算法的描述。

3. 3073 [[来源：ISO/IEC 2382:2015, X. X]

问题报告 problem report; PR

用于识别和描述产品中检测到的问题的文档。

注：问题报告要么直接提交以表示故障，要么在针对修改请求执行影响分析后发现故障时建立。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3464]

3. 3074

问题状态 problem state

从属状态 slave state

用户状态 user state

在计算机系统的运行中，除了监控程序之外的程序可以执行的状态。主管状态。

3. 3075

面向问题的语言 problem-oriented language

为解决给定问题类别而设计的编程语言；

示例：列表处理语言、信息检索语言、模拟语言。

[来源：ISO/IEC/IEEE FDIS 24765, 2025]

3. 3076

过程内聚 procedural cohesion

一种内聚性类型，其中软件模块执行的任务都服务于给定的程序流程，如迭代或决策过程。另见偶然内聚性、通信内聚性、功能内聚性、逻辑内聚性、顺序内聚性、时间内聚性。

[来源：ISO/IEC/IEEE FDIS 24765, X. X]

3. 3077

过程语言 procedural language

面向过程的语言 procedure-oriented language

一种编程语言，用户在其中编写一组特定的指令，计算机按给定的顺序执行。参见非过程式语言、代数语言、算法语言、列表处理语言、逻辑编程语言。

注：所有广泛使用的编程语言均属于此类。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3468]

3. 3078

过程编程语言 procedural programming language

用于表达计算机要执行的操作序列的计算机编程语言。

参见：非过程式编程语言。

示例：COBOL。

[来源：ISO/IEC/IEEE FDIS 24765, X. X]

3. 3079

过程/规程 Procedure

指定如何执行任务的一系列有序步骤。

注：过程/规程定义了组织中既定且批准的业务执行方式或模式。它详细说明了为实现技术或管理目标或成果而允许或推荐的方法。当过程/规程作为输出指定时，所产生的交付物通常会说明做什么、由谁做以及按什么顺序做。

它比过程（process）的规范层次更详细。过程/规程可以有文档记录，也可以没有。

[来源：ISO/IEC/IEEE 26514:2022 3.1.39]

3.3080

规程测试 procedure testing

功能性测试的一种类型。用于评价与测试项交互或使用其输出的规程指令是否满足用户需求及其使用目的。

[来源：GB/T 38634.1-2020, 3.29]

3.3081

过程 process

将输入转化为输出的相互关联或相互作用的一组活动。

[来源：ISO/IEC/IEEE 15288:2023 《系统和软件工程——系统生存周期过程》，3.27]

3.3082

过程行动计划 process action plan

通常源于评估所制定的计划，用于记录如何针对评估中发现的薄弱环节实施具体改进措施。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3473]

3.3083

过程行动组 process action team

负责为组织制定和实施过程行动计划中记录的过程改进活动的团队。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3474]

3.3084

过程分析 process analysis

过程分析遵循过程改进计划中概述的步骤以确定需要的改进。

[来源：ISO/IEC/IEEE 24765:2017, 3.3040]

3.3085

过程架构师 process architect

主要负责创建和维护软件生存周期过程（SLCP）的个人或团体。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3475]

3.3086

过程架构 process architecture

标准过程中过程元素之间的顺序、接口、相互依赖关系和其他关系。

注：过程体系结构还描述了过程元素与外部过程（如合同管理）之间的接口、相互依赖关系和其他关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3476]

3.3087

过程区域 process area

一个领域内的一组相关实践，当集体实施时，可以满足一系列被认为对改进该领域很重要的目标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3477]

3.3088

过程评估 process assessment

根据过程评估模型对组织单位过程进行严格评估。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3478]

3.3089

过程评估模型 process assessment model; PAM

基于一个或多个过程参考模型形成的适用于评估特定过程质量特性的模型。

参见：过程参考模型（PRM）。

注：针对特定过程质量特性的过程评估模型，可以在标题中包含该特性的名称；例如，针对过程能力的过程评估模型可以称为“过程能力评估模型”。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3479]

3.3090

过程资产库 process asset library

对组织中定义、实施和管理过程的人员有用的信息集。

示例：与过程相关的文档，例如政策、定义过程、检查表、经验教训文档、模板、标准、程序、计划和培训材料。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3480]

3.3091

过程属性 process attribute; PA

过程质量特性的可测量属性。

同义词：过程质量属性。

[来源：ISO/IEC 33001:2015, 3.4.3]

3.3092

过程属性结果 process attribute outcome

达到指定过程属性的可观察结果。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3482]

3.3093

过程属性评级 process attribute rating

被评估过程的过程属性达成程度的判断。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3483]

3.3094

过程能力 process capability

通过遵循一个过程可以实现的预期结果范围。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3484]

3.3095

过程能力基线 process capability baseline

对典型环境下遵循一个特定过程通常所能达到的预期结果范围的一种文档化的特征描述。一般在组织层上建立过程能力基线。

3.3096

过程能力确定 process capability determination

在一个组织内部，对选定的软件过程对照某个目标能力所进行系统性的评估和分析，其目的是识别出所使用过程的强项、弱项及相关风险以满足某个特定要求。

3.3097

过程能力等级 process capability level

在过程能力的有序测量尺度上表征过程。

注：每个级别都建立在下面级别的能力之上。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3485]

3.3098

过程能力评级 process capability rating

一种对某个已评估过程的过程属性评定所达到的过程能力等级的表示。

3.3099

流程类别 process category

涉及活动的方面的一组过程。

注：过程范畴涉及活动的5种一般方面：顾客—供方、工程、支持、管理和组织。

3.3100

过程组件 process component

CORBA 组件，具有持久状态（对客户端不可见）、持久标识和行为，并且可以是事务性的。

[来源：ISO/IEC/IEEE 24765:2017, 3.3052]

3.3101

过程环境 process context

记录在评估输入中的一组因素，这些因素会影响对过程属性评级的判断、理解与可比性。

[来源：GB/T 11457-2006, 2.1194]

3.3102

过程决策程序图 process decision program chart; PDPC

过程决策程序图用于理解目标与达成该目标所需步骤之间的关系。

[来源：ISO/IEC/IEEE 24765:2017, 3.3053]

3.3103

过程定义 process definition

识别过程的目的、成果以及为特定目标执行的一系列包含活动、约束和资源的步骤。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3486]

3.3104

过程描述 process description

为实现给定目的而执行的一组活动的文件化表达。

注：过程描述提供了过程主要组成部分的操作定义。描述以完整、精确和可验证的方式指定过程的目的、结果、活动和任务、要求、设计、行为或其他特征。它还可以参考确定是否满足这些规定的程序。过程描述可适用于项目或组织级别。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3487]

3.3105

工艺开发 process development

定义和描述过程的行动。它可包括策划、构造、设计、实施和确认。

3.3106

过程维度 process dimension

过程评估模型中与相关过程参考模型中定义的过程明确相关的一组过程元素。

注：过程维度的元素包括过程、过程目的陈述、过程结果和过程绩效指标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3488]

3.3107

处理证据 process evidence

表征过程是否满足期望的过程可信度的证据。

注：该证据的数据类型分为布尔型、数值型、百分比型和等级型。

[来源：GB/T 37970-2019, 3.1.11]

3.3108

过程组 process group

相关过程的集合。

2. 促进组织所用过程的定义、维护和改进的专家团队。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3489]

3.3109

过程改进 process improvement

根据业务需求和其他相关方的需求，为提高组织过程质量而采取的行动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3490]

3.3110

流程改进措施 process improvement action

某项计划的和执行的行动以改进所有或部分软件过程。

注：（一项）过程改进行动能有助于实现一个以上的过程目标。

3.3111

过程改进目标 process improvement objective

建立的一组目标特征，用于以特定的、可衡量的方式改进现有过程，无论是在结果产品或服务的特性（如质量、性能和符合标准），还是在过程的执行方式（如消除冗余过程步骤、合并过程步骤、缩短周期时间）方面。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3491]

3.3112

过程改进计划 process improvement plan

项目管理计划的附属计划。它详细说明了分析过程以识别提高其价值的活动的步骤。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3492]

3.3113

流程改进计划 process improvement programme

与特定改进目标实现相关的所有战略、方针、目标、职责和活动。

注：一项过程改进计划可以横跨一个以上的过程改进的完整的周期。

3.3114

流程改进项目 process improvement project

任何为实现某个特定改进所形成的过程改进计划的子集合。

3.3115

过程改进支持要素 process improvement support element

组织表达对过程改进项目或计划支持的方式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3493]

3.3116

过程基础设施 process infrastructure

软件生存周期过程的内部结构，包括生存周期阶段、文档、基线、评审和产品。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3494]

3.3117

过程实例 process instance

过程的单一特定和可识别执行。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3495]

3.3118

过程管理 process management

为开发产品或执行服务所开展工作的指导、控制和协调。

示例：质量保证。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3496]

3.3119

过程成熟度 process maturity

组织单位在规定范围内持续实施有助于实现其业务需求（当前或预计）的过程的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3497]

3.3120

过程测量 process measurement

为测量过程及其生成的产品而采用的一组定义、方法和活动，其目的是为了描述和了解该过程。

3.3121

过程测量框架 process measurement framework

用于表征已实施过程的过程质量特征的模式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3498]

3.3122

过程指标 process metric

用于衡量开发、实施和维护软件系统所采用的方法、技术和工具特性的指标。

[来源：ISO/IEC/IEEE 24765:2017, 3.3066]

3.3123

过程结果 process outcome

成功实现过程目标的可观察结果。

参见：输出。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3499]

3.3124

过程所有者 process owner

负责定义和维护过程的人员（或团队）。

注：在组织层面，过程所有者是负责描述标准过程的人员（或团队）；在项目级别，过程所有者是负责描述已定义过程的人员（或团队）。因此，一个过程可以有多个责任级别不同的所有者。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3500]

3.3125

过程性能 process performance

过程执行实现其目的的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3501]

3.3126

过程绩效基线 process performance baseline

对执行一个过程所达到实际结果的一种文档化的特征描述。它用作将实际过程性能与期望的过程能力进行比较的基准。一般在项目层上建立过程性能基线，尽管最初的过程性能基线通常由组织的过程能力基线导出。

参见：2.1188 条 过程能力基线（process capability baseline）。

3.3127

过程性能指标 process performance indicator

支持判断特定过程的过程性能的评估指标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3502]

3.3128

过程轮廓 process profile

某个已评估过程的过程属性鉴定的集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3503]

3.3129

过程目的 process purpose

执行过程的高级目标和有效实施过程的可能结果。

注：过程的实施旨在为利益相关者提供切实的利益。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3504]

3.3130

过程质量 process quality

在特定环境使用时，过程满足明确和隐含的利益相关者需求的能力。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3505]

3.3131

过程质量属性 process quality attribute

过程质量特性的可测量特性。

[来源：ISO/IEC 33001:2015, 3.4.3]

3.3132

过程质量特性 process quality characteristic

过程质量的可测量方面。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3506]

3.3133

过程质量确定 process quality determination

根据目标过程概况对选定过程进行系统评估和分析。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3507]

3.3134

过程质量维度 process quality dimension

过程评估模型中与特定过程质量特性的过程测量框架明确相关的一组元素。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3508]

3.3135

过程质量指标 process quality indicator

支持判断特定过程的过程质量特性的评价指标。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3509]

3.3136

过程质量等级 process quality level

从被评估过程的过程属性评级得出的过程质量特征的成就等级。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3510]

3.3137

过程参考模型 process reference model; PRM

定义了过程的应用领域、目的和结果，以及过程间关系的模型。

参见：过程评价模型。

[来源：ISO/IEC/IEEE 12207:2026 3.1.48]

3.3138

过程风险确定 process risk determination

针对目标过程概况对选定过程进行系统评估和分析，目的是识别过程相关风险以满足特定的规定要求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3512]

3.3139

过程标准 process standard

处理用于制造或实现产品的一系列动作或操作的标准。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3513]

3.3140

过程剪裁 process tailoring

为特定目的制作、更改或改编过程描述。

示例：项目从组织的标准过程集中裁剪已定义的过程以满足项目的目标、约束和环境。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3514]

3.3141

过程可信度 process trustworthiness

软件过程能够生产满足期望产品的信心度。

[来源：GB/T 37970-2019, 3.1.10]

3.3142

过程视图 process view

描述如何通过采用现有过程的活动和任务来实现特定目的和结果集。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3515]

3.3143

过程证据要素 process-evidence element

使用于测量过程证明的测量元。

[来源：GB/T 37970-2019, 3.1.12]

3.3144

处理逻辑 processing logic

用户为完成基本过程（如验证、算法或运算，以及读取或维护数据功能）而特别要求的需求。

[来源：GB/T 42449-2023]

3.3145

处理器 processor

在计算机中，解释和执行指令的功能单元。

注：处理器至少由指令控制单元和算术逻辑单元组成。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3518]

3.3146

过程相关风险 process-related risk

由于过程在执行、管理或部署方面的弱点而产生的风险。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3516]

3.3147

采购审核 procurement audit

审查合同和承包过程的完整性、准确性和有效性。

[来源：ISO/IEC/IEEE 24765:2017, 3.3085]

3.3148

采购文件 procurement documents

投标和报价活动中使用的文件，包括买方的投标邀请书、谈判邀请书、信息征询书、报价请求书、建议征求书和卖方的答复。

[来源：ISO/IEC/IEEE 24765:2017, 3.3086]

3.3149

采购管理计划 procurement management plan

项目或项目管理计划的组成部分，描述项目团队如何从执行组织外部获取商品和服务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3519]

3.3150

采购绩效审查 procurement performance review

对照合同，对卖方在交付项目范围和质量、成本控制以及进度遵守情况方面的进展进行的结构化审查。

[来源：ISO/IEC/IEEE 24765:2017, 3.3088]

3.3151

采购工作说明书 procurement statement of work

足够详细地描述采购项目，让潜在卖家确定他们是否有能力提供产品、服务或结果。

[来源：ISO/IEC/IEEE 24765:2017, 3.3089]

3.3152

生产者对象 producer object

生产者对象是开放式分布式处理（ODP）参考模型中定义的一种对象，它作为一个流（flow）（如序号2109）的起点，是信息的来源（source）。它通过交互将信息传递给对应的消费者对象（consumer object），共同完成信息的传递。[来源：ISO/IEC 10746-2:2009, 13.4.3]。

3.3153

产品分析 product analysis

通过手动或自动方式评估产品以确定产品是否具有某些特性的过程。2. 对于将产品作为可交付成果的项目，它是一种定义范围的工具，通常意味着询问有关产品的问题并形成答案来描述用途、特性和将要制造的产品的其他相关方面。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3522-product analysis]

3.3154

产品资产实例 product asset instance

特定于成员产品的共享资产的实例化，由产品线工程（PLE）工厂配置器自动生成，对应于该成员产品的特征清单。

注：产品资产实例类似于应用资产，但前提条件是它是由产品线工程（PLE）工厂配置器生成的。

[来源：ISO/IEC 26580:2021]

3.3155

产品权威 product authority

对产品的性能和质量负总责的人员。

3.3156

产品待办列表 product backlog

产品待办事项列表，通常以用户故事的形式呈现，包含了给定产品所有待办的需求。

[来源：ISO/IEC 29110-1-2:2024 3.69]

按照优先级排序的产品需求列表。

参见：迭代待办事项列表。

[来源：ISO/IEC 33202:2024 3.22]

3.3157

产品基线 product baseline

产品配置基线 product configuration baseline

生产配置 production configuration

在特定时间点对详细设计的描述，用于生产、部署、配置以及运营和支持。

参见：分配基线、开发配置、功能基线、产品配置标识。

示例：产品基线规定了所有必要的物理形态、适配性或功能特性，以及指定用于生产验收测试和生产测试要求的的选择性功能特性。

[来源：ISO/IEC/IEEE 24748-7:2019, 3.1]

3.3158

产品基线 product baseline

product configuration baseline

production configuration

在特定时间点对详细设计的描述，用于生产、现场使用、部署、运行以及操作与支持。

示例：产品基线规定了所有必要的物理形式、配合度或功能特性以及选定的功能。

参见：指派基线、开发配置、功能基线、产品配置标识

[来源：ISO/IEC/IEEE 24748-7:2019, 3.1]

3.3159

产品分解结构 Product breakdown structure; PBS

产品分解为其组件的分解结构。

注：PBS 以层级顶部的完整产品开始，其下方是主要组件，这些组件也可以各自进一步分解为组件。

[来源：ISO/IEC TR 7052:2023, 3.32]

3.3160

产品认证 product certification

一个系统、部件或计算机程序符合其规定的需求，对操作使用是可接受的一种书面保证。

示例：一计算机系统是安全的允许在定义的环境中操作的书面的认可。

3.3161

产品配置标识 product configuration identification

在产品基线确立后，用以定义和描述配置项在其生产、部署、运行、维护及保障阶段具体形态的、经正式批准或附带条件批准的技术文档集合。

注：这些文档记录了配置项所有必需的物理特性（形式、配合、功能），明确为生产验收测试选定的功能特性，并规定相应的测试要求。它与分配的配置标识、功能配置标识共同构成了技术状态管理的三大标识体系。

[来源：综合 GB/T 11457-2006（产品配置标识）、ISO/IEC/IEEE 24765 标准及 INCOSE 手册对配置识别过程的描述。]

3.3162

产品描述 product description

产品描述（product description）是即用型软件产品（RUSP）等产品的一项核心技术文档，其主要目的是向潜在用户或采办方说明产品（尤其是软件）的属性、功能和价值主张，帮助他们在购买前评估该产品是否满足其自身需求，以支持其采购决策。它明确区别于规范（specification），后者的目的是为设计和实现提供技术约束。

[来源：ISO/IEC 25051:2014]

3.3163

产品工程 product engineering

为定义、设计、构建或组装产品而执行的技术过程。

[来源：ISO/IEC/IEEE 24765: FDIS]

3.3164

产品标识 product identification

软件产品的名称、版本、变体及日期信息。

[来源：ISO/IEC 25051:2014, 4.1.12]

3.3165

个人信息 Product Improve; PI

产品改进

(1) 个人信息。

[来源：IEEE 7002:2022, IEEE, 3.1]

3.3166

产品实例 product instances

包含特征清单组合和产品资产实例的集合。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.3167

产品库 product library

投产库

一种软件库，其中含有已被批准供当前运行使用的软件。

[来源：GB/T 11457-2006, 2.1217]

3.3168

产品生存周期 product life cycle

代表产品从概念到交付、成长、成熟直至退市（淘汰）演进的一系列阶段。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3532-product life cycle]。

3.3169

产品线 product line

一组产品或服务，具有相同的明确定义和管理的通用和可变的特征，且依赖相同的领域架构满足特定市场的通用和可变需求。

[来源：GB/T 41865-2022]

3.3170

产品线采用计划 product line adoption plan

描述从当前状态到产品线工程所需的过程、组织结构和产品构建方法变更的计划。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3534]

3.3171

产品线采用场景 product line adoption scenario

提供与产品线采用相关的具体行动序列的场景。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3535]

3.3172

产品线架构 product line architecture

同领域架构。

[来源：GB/T 41865-2022, XXXX]

3.3173

产品线配置 product line configuration

产品线快照，包含该产品线中所有领域资产与成员产品的全部修订版本集合。

[来源：ISO/IEC 26563:2022, 3.4]

3.3174

产品线配置增量 product line configuration delta

两个版本产品线配置之间的差异。

[来源：ISO/IEC 26563:2022 3.5]

3.3175

产品线配置管理 product line configuration management

指导和控制产品线配置的协调活动。

注：产品线配置管理包括领域工程、应用工程以及组织性与技术性管理的配置管理活动。

[来源：ISO/IEC 26563:2022, 3.7]

3.3176

产品线工程工厂 product line engineering factory

技术、组织和业务基础设施及流程，以支持产品线工程（PLE）工厂配置器根据成员产品的功能清单，从共享资产超集中生成产品资产实例。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3541]

3.3177

产品线工程工厂配置器 product line engineering factory configurator

一种自动化机制，通过处理特定成员产品的特征清单，并根据该特征清单中的特征选择来运用共享资产（即变体点），从而为该成员产品生成资产。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3542]

3.3178

产品线工程工厂开发环境 product line engineering factory development environment

一套工具集，用于在功能目录、功能清单组合、共享资产超集以及产品线层次结构中创建、组织、组装和维护一系列元素。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3543]

3.3179

产品线制度化 product line institutionalization

让相关管理人员和员工将产品线工程视为工作文化的一部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3544]

3.3180

产品线平台 product line platform

由产品线体系结构、配置管理计划和领域资产构成的集合，使应用工程能够有效地重用和生产一组衍生产品。

[来源：ISO/IEC 26550:2015, 3.18]

3.3181

产品线产品系列 product line product family

一组共享明确定义和管理的共同且可变特征的产品或服务，它们依赖于相同的领域架构来满足特定市场的共同且可变的需求。

[来源：GB/T 41865-2022, X. X]

3.3182

产品线参考模型 product line reference model

领域工程和应用工程生存周期过程及其角色、关系的抽象表示，以及在产品线工程和管理过程中所生产、管理和使用的资产。

[来源：ISO/IEC 26550:2015, 3.19]

3.3183

产品线范围界定 product line scoping

定义产品线内将生产的成员产品和产品间主要共同与可变特性的过程，从经济角度分析产品，并控制和安排产品线及其产品的开发、生产和上市。

[来源：ISO/IEC 26550:2015, 3.20]

3.3184

产品线测试策略 product line test strategy

领域测试和应用测试的范围。

注：许多变体可能在领域工程中并未实现。为应对未实现变体对领域集成测试和系统测试的影响，应制定适当的测试策略。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3548]

3.3185

产品线配置项 product line-configuration item

构成产品线的成果性工件。

注：产品线配置项是对领域工程、应用工程及管理支持的配置项的统称。

[来源：ISO/IEC 26563:2022, 3.6]

3.3186

产品管理 product management

在产品开发周期中，对产品特性的定义、协调和控制。

示例：配置管理。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

3.3187

产品指标 product metric

用于度量软件开发过程中任何中间产品或最终产品特性的度量。

[来源：IEEE 1061-1998, 2.16]

3.3188

产品拥有者 product owner

负责产品功能、验收和使用的利益相关方。

注：产品负责人负责分享产品愿景、所需功能及其优先级以及验收标准。

[来源：ISO/IEC TR 24587:2021]

3.3189

产品质量 product quality

在规定条件下使用时，产品满足明确和隐含要求的能力。

注：该定义不同于 GB/T 19000-2016 中的质量定义，在本文件中软件质量是指满足明确和隐含的要求，而 GB/T 19000 中质量定义是指满足需求。

[来源：GB/T 25000.41-2018, 4.11, 有修改]

3.3190

产品要求 product requirement

为软件开发人员（生产者）描述（产品）的需求。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3191

产品风险 product risk

产品在功能、质量或结构某些特定方面存在缺陷的风险。

[来源：ISO/IEC/IEEE 29119-3:2013, 4.8]

3.3192

产品路线图 product roadmap

具有产品生存周期高级里程碑的时间表，特别是产品生产部署的时间表。

[来源：ISO/IEC 26556:2018, 3.2]

3.3193

产品安全标签 product safety label

产品上的一种标签，用于告知一个或多个潜在危险，并描述避免该危险所需的安全预防措施或行动。

[来源：IEC/IEEE 82079-1:2019 3.29]

3.3194

产品范围 product scope

构成产品、服务或成果的特征与功能。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3195

产品范围说明 product scope description

对产品范围的文档化叙述性描述。

[来源：ISO/IEC/IEEE 24765:2017, 3.3110-product scope description]

3.3196

产品范围界定 product scoping

产品线范围界定的子过程，用以确定产品路线图，包括：1) 目标市场；2) 产品线组织应开发、生产、营销和销售的产品类别；3) 为达到组织长短期业务目标产品应提供的共同和可变特性；4) 产品投放市场的时间表。

[来源：ISO/IEC 26550:2015 软件和系统工程 - 产品线工程和管理参考模型, 3.20]

3.3197

产品规格 product specification

规定系统或系统组件生产版本必须实现的设计的文档。

[来源: ISO/IEC/IEEE 24765:2017 系统和软件工程 – 词汇]

3.3198

产品标准 product standard

定义软件工程过程中（正式或非正式地）使用或生产的项目的完备性和可接受性构成要素的标准。

[来源: ISO/IEC/IEEE 24765:2017 系统和软件工程 – 词汇（有修改）]

3.3199

产品支持 product support

为在预期环境中安装并使软件投入运行，以及向用户分发改进后的功能，而提供信息、帮助和培训。

[来源: ISO/IEC/IEEE 24765:2017 系统和软件工程]

3.3200

生产 production

（软件或文档）已完成、发布并打包待分发的生存周期阶段。

[来源: ISO/IEC/IEEE 24765:2017 系统和软件工程]

3.3201

生产与部署 production and deployment; P&D

生产与部署。

[来源: ISO/IEC/IEEE 24748-7:2019, 3.2]

[来源: 同义词: P & D]

3.3202

产品资料库 production library

包含当前已批准投入使用的软件的软件库。

相对: 软件开发库、软件存储库、系统库。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3562]

3.3203

生产计划 production plan

描述如何应用领域资产来开发产品线中的成员产品的说明。

[来源: ISO/IEC 26551:2016 软件和系统工程 – 产品线需求工程的工具和方法, 3.17]

3.3204

生产率 production rate

衡量单位时间内完成工作量的指标，例如每周完成的用户故事数或功能数量。

参见: 燃尽速率 (burndown rate)、迭代速率 (velocity)。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — Fifth Edition]

3.3205

外廓 profile

剖面

指一组适当的标准流程及其成果、活动与任务的组合，用于实现特定功能。

注: [来源: ISO/IEC TR 10000-1]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3568]

3.3206

配置文件组 profile group; PG

由过程构成（如活动、任务）、能力等级或两者兼具而相互关联的一组概要文件。

[来源: ISO/IEC 29110-1-2:2024, 3.71]

3. 3207

程序 program

编程（动）

写计算机程序。

参见：计算机程序

[来源：GB/T 11457-2006, 2.1225]

3. 3208

程序架构 program architecture

计算机程序部件之间的结构和关系。程序体系结构也可以包括它和程序运行环境之间的程序界面。

[来源：GB/T 11457-2006, 2.1226]

3. 3209

程序块 program block

在面向问题的语言中，计算机程序的子部分，它用于把语句分组、划分例行程序、规定存储分配、确定标号的可使用性、或者为其他一些目的而将计算机程序的分段。

[来源：GB/T 11457-2006, 2.1227]

3. 3210

程序结构 program construct

一个或多个过程部分以及一个可为隐式的控制部分的集合。

注：每个过程部分由一个或多个待执行的操作组成，或可为空。

[来源：ISO/IEC/IEEE

[来源：24765i:2020; ISO/IEC/IEEE 24765 FDIS, 3.3571]]

3. 3211

程序计数器 program counter

指示下一条将要执行的计算机指令所在位置的寄存器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.2328]

3. 3212

程序设计语言 program design language; PDL; PDL

一种具有特殊结构和（有时还包括）验证协议的规范语言，用于开发、分析和记录程序设计。

参见：硬件设计语言、伪代码。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3572]

3. 3213

计划评审技术 Program Evaluation and Review Technique; PERT

在单个活动估算存在不确定性时，应用乐观、悲观和最可能估算的加权平均值进行估算的技术。

[来源：ISO/IEC/IEEE 16326:2019, 3]

3. 3214

计划评估审查技术 Program Evaluation Review Technique; PERT

一种估算技术，当单项活动估算存在不确定性时，该方法对乐观、悲观和最可能的估算值应用加权平均。

[来源：ISO/IEC/IEEE 24765h:2019; ISO/IEC/IEEE 24765 FDIS, 3.3573]

3. 3215

程序扩展 program extension

对现存软件进行增强以扩大程序能力的范围。

[来源: GB/T 11457-2006, 2.1232]

3.3216

程序指令 program instruction

源程序中的计算机指令。

注: 程序指令与通过汇编、编译或其他解释过程产生的计算机指令是不同的。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3574]

3.3217

程序插装 program instrumentation

插入到计算机程序中的探头如指令或断言, 以利于执行监控、正确性证明、资源监控或其他活动。

[来源: GB/T 11457-2006, 2.1235]

3.3218

程序库 program library

为辅助软件开发、使用或维护而设计的软件及相关文档的有控制收集。

参见: 存储库。

示例: 类型包括主库、生产库、软件开发库、系统库。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4348]

3.3219

程序列表 program listing

打印输出或以其他人类可读的方式展示构成计算机程序的源代码, 有时也包括目标语句。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3576]

3.3220

程序维护手册 program maintenance manual

提供维护程序所需信息的文档。

[来源: ISO/IEC 2382:2015 信息技术 词汇]

3.3221

项目群管理 program management

将知识、技能和原则应用于项目群, 以实现项目群目标, 并获得通过单独管理项目群各组成部分无法获得的效益和控制。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3578-program management]

3.3222

程序突变 program mutation

一种故意从原始版本进行修改的计算机程序, 用于评估测试用例检测修改的能力。

参见: 变异测试。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3579]

3.3223

程序网络图 program network chart

显示两个或多个计算机程序之间关系的图表。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3580]

3.3224

程序保护 program protection

为预防对计算机程序的任何非授权存取或修改而施行的内部或外部控制。

[GB/T 11457-2006, 2.1241]。

3.3225

程序规范 **program specification**

描述程序结构和功能的详细文档，以便于编程和维护。

[来源：ISO/IEC 2382:2015 信息技术 词汇]

3.3226

程序状态字 **program status word; PSW**

包含描述计算机程序当前状态信息的计算机术语。

注：这些信息可以包括错误指示符、下一指令的执行地址，或者当前启用的中断。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3582]

3.3227

程序综合 **program synthesis**

使用软件工具辅助将程序规范转换为实现该规范的程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3583]

3.3228

可编程断点 **programmable breakpoint**

一种断点，当其被触发时，会自动调用之前指定的调试流程。

参见：代码断点、数据断点、动态断点、尾声断点、序言断点、静态断点。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3585]

3.3229

可编程计数器阵列 **programmable counter array; PCA**

微控制器单元上的一组计数器模块可提高其计时能力。

示例：计数参数以编程方式配置。

[来源：ISO/IEC/IEEE 24765d:2015; ISO/IEC/IEEE 24765 FDIS, 3.3586]

3.3230

可编程逻辑器件 **programmable logic device**

允许用户编程（配置）实现所需逻辑功能的器件。

[来源：GB/T 37691-2019, 1.187]

3.3231

可编程逻辑器件软件 **programmable logic device software**

针对 FPGA、CPLD 等可编写程序传递控制器进行设计并生成的程序、文档和数据。

[来源：GB/T 37691-2019, 3.2]

3.3232

可编程脉冲发生器 **programmable pulse generator**

产生时钟信号的微控制器单元的一部分，可通过编程进行调整。

[来源：ISO/IEC/IEEE 24765d:2025; ISO/IEC/IEEE 24765 FDIS, 3.3587]

3.3233

可编程只读存储器 **programmable read-only memory; PROM**

只读存储器单元，不具有重写或擦除数据功能，用户只能编程一次。

参见：可编程只读存储器（programmable read only memory）。

[来源：ISO/IEC/IEEE 24765c:2014; ISO/IEC/IEEE 24765 FDIS, 3.3588]

3.3234

可编程重载定时器 **programmable reload timer; PRT**

可通过递减、状态控制及重新加载寄存器来重启功能的可重配置设备。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3589]

3.3235

可编程终端 programmable terminal

内置数据处理能力的用户终端。

3.3236 [[来源：ISO/IEC 2382:2015, X.X]

可编程终端智能终端 programmable terminal intelligent terminal

内置数据处理能力的可编程智能终端。

[来源：ISO/IEC 2382:2015, 信息技术 词汇]]

3.3237

编程参考点 programmatic reference point

可以建立编程接口以允许访问功能的参考点。

[来源：ISO/IEC 10746-2:2009, 15.3.1]

3.3238

程序员手册 programmer manual

为特定计算机系统开展软件开发或软件修改工作，提供所需必要信息的文件。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3592, 有修改]

3.3239

编程 programming

程序的设计、编写、修改和测试。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.3240

编程语言 programming language

用于表达计算机程序的语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3594]]

3.3241

程序设计支持环境 programming support environment

通过单一命令语言访问的软件工具集成集合，为软件生存周期提供编程支持功能。

参见：脚手架（scaffolding）。

注：有时称为集成编程支持环境，该环境通常包括用于规范、设计、编辑、编译、加载、测试、配置管理和项目管理的工具。

[来源：ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE 24765 FDIS, 3.3595]

3.3242

编程系统 programming system

一组编程语言和在给定计算机系统中使用这些语言所必需的支持软件（编辑器、编译器、链接器等）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3596]]

3.3243

程序敏感故障 program-sensitive fault

当执行某些特定的程序步骤序列时导致失败的故障。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3584]]

3.3244

渐进明细 progressive elaboration

随着更多信息和更准确的估算可用，逐步增加项目管理计划细节的迭代过程。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3245

禁令 prohibition

禁止某种特定行为的规定。

[来源：ISO/IEC 10746-2:2009 11.2.6]

3.3246

项目 project

按照确定的开始与结束准则，依据给定的资源和需求创建产品或服务开展的工作。

注：项目可看做是包含协作和受控活动的独特过程，可由本标准中定义的项目过程和技术过程的活动组成。

[来源：ISO/IEC/IEEE 15288:2023 3.33; GB/T 22032-2021]

3.3247

项目平衡 project balance

以等时间间隔的现金额序列，表示的某替代方案至今的累积价值。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3248

项目简报 project brief

项目目标、可交付成果及流程的高层级概述。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3249

项目日历 project calendar

标识可用于预定活动的工作日和班次的日历。

参见：资源日历（resource calendar）。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3250

项目沟通管理 project communications management

项目沟通管理是项目管理的一个核心知识领域，其目的是通过执行一套系统的过程，确保项目信息在其整个生存周期内（从规划到最终归档）能够被及时、适当地生成、分发、存储、检索、控制和处置，以满足干系人的信息需求并支持有效的项目决策。其核心过程包括规划沟通管理、管理沟通过程和控制沟通。

[来源：ISO/IEC/IEEE 24765:2017, 3.3156]

3.3251

项目控制 project control

项目控制是通过一系列活动，将项目在进度、方向、质量和资源利用等方面的实际执行情况与预先制定的项目计划进行持续比较、监控和评估的管理活动。其核心目的是识别偏差，并通过采取纠正或预防措施，确保项目目标的实现。它是项目管理循环（计划-执行-监控）中的关键环节。

[来源：ISO/IEC 2382:2015 (project control); 同时与《INCOSE 手册》中的“Project Assessment and Control Process”（项目评估与控制过程）定义及目的完全一致。]

3.3252

项目成本管理 project cost management

项目成本管理是项目管理的一个核心知识领域，旨在通过系统性的过程，对项目成本的规划、估算、预算、融资、资金筹措、管理和控制进行综合处理，最终目标是确保项目在不超出批准预算的前提下成

功完成。其涉及的关键过程包括计划成本管理 (plan cost management, 6793)、估算成本 (estimate costs, 1882)、制定预算 (determine budget, 1459) 和控制成本 (control costs, 1008)。

[来源: ISO/IEC/IEEE 24765:2017, 3.3158-project cost management]

3.3253

项目文件 project file

归集项目相关资料的核心存储载体。

注: 内容通常包含备忘录、方案计划、技术报告及其他相关资料。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3605]

3.3254

项目簿 project file project notebook

中心储藏处的与项目相关的资料。内容典型地包括: 备忘录、计划、技术报告和相关项。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3605]]

3.3255

项目功能点数 project function point count

项目功能点数 是使用功能点作为单位, 对开发项目或增强项目的功能规模进行量化的结果。它代表了项目完成后为最终用户增加、修改或删除的全部功能总量。此测量结果主要用于项目估算, 例如预测实现新软件或修改现有软件所需的工作量 (人月)。

[来源: ISO/IEC/IEEE 24765:2017, 3.3160]

3.3256

项目功能规模 project functional size

用功能点表示的开发项目或者增强项目的规模, 包括要添加、更改和删除的全部功能。

注: 项目功能点计数使我们能够确定实现新软件, 或者修改既有软件的功能所需要的投入。在后面一种情况下, 一个项目功能点计数涉及到对功能的添加、修改和删除。

[来源: GB/T 42588-2023, X.X]

3.3257

项目资金需求 project funding requirements

项目资金需求是项目管理中一个财务管理度量, 指根据成本基准 (cost baseline) 计算的、为满足项目周期性 (如每月、每季度) 或整体需求而必须预备的预测性现金支付需求。它不仅包括计划内的预计支出, 还包括预计负债, 是制定项目预算 (project budget, 3910) 和执行资金限额核对 (funding limit reconciliation, 2204) 以确保持续运营的关键输入和依据。

[来源: ISO/IEC/IEEE 24765:2017, 3.3161]

3.3258

项目治理 project governance

项目治理是通过建立一套框架 (framework)、功能 (function) 和过程 (process), 由项目发起人 (sponsor) 和项目管理团队 (project management team) 具体实施, 旨在确保项目目标始终与所在更大组织 (项目集或组织) 的战略目标保持高度一致, 并有效管理关键项目决策的有效性和问责性 (accountability)。它虽然受更大组织治理环境的影响, 并需要与之契合, 但在管理范围和决策层面聚焦于项目本身, 因此被视为 组织治理 (organizational governance) 在项目层面的延伸和具体应用。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3606]

3.3259

计划启动 project initiation

项目启动 (Project Initiation) 是项目管理的一个初始阶段或活动, 指通过启动一个正式的论

证和规划过程（如制定项目章程），以最终获得组织对新项目的正式授权或对现有项目新阶段的批准。这一过程是整个项目生存周期（project life cycle）的起点，属于项目管理五大过程组中的启动过程组（initiating process group, 6379）。

[来源：ISO/IEC/IEEE 24765:2017, 3.3164]

3.3260

项目整合管理 project integration management

包括识别、定义、组合、统一和协调项目管理过程组内各过程和项目管理活动所需的过程和活动。

[来源：ISO/IEC/IEEE 24765:2017, 3.3165]

3.3261

项目负责人 project lead

通过协调项目的工作来帮助项目团队实现项目目标的人。

参见：项目经理

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3262

项目生存周期 project life cycle

项目从启动到收尾所经历的一系列阶段集合。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3263

项目管理 project management (PM)

将知识、技能、工具和技术应用于项目活动，以满足项目要求。

[来源：ISO/IEC/IEEE 24765, 3.3609]

3.3264

项目管理知识领域 project management knowledge area

由知识要求定义的项目管理特定领域，并依据其组成部分的过程、实践、输入、输出、工具和技术进行描述。

[来源：ISO/IEC/IEEE 24765:2017, 3.3170]

3.3265

项目管理办公室 project management office; PMO

一种管理结构，用于标准化与项目相关的治理流程，并促进资源、方法论、工具和技术共享。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3266

项目管理计划 project management plan; PMP

描述项目将如何被执行、监控和控制的文件。

[来源：ISO/IEC/IEEE 24765, 3.3612]

3.3267

项目管理过程组 project management process group

项目管理过程组是项目管理知识体系中的一个核心框架，指对项目管理过程的逻辑性分类与集合。这些过程组并非项目阶段，而是代表了项目管理活动的类型（横向切面），它在项目生存周期的任何阶段都可能被依赖和重复应用。通常包括启动、规划、执行、监控与控制、收尾五个基本过程组。这一概念提供了组织和管理项目活动的标准方法，并与项目管理知识领域（project management knowledge area）形成矩阵，共同构成了项目管理的完整理论与实践体系。[同义词或关联术语：process group（进程组）]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3613]

3.3268

项目管理软件 project management software

专门设计用于帮助项目管理团队规划、监控和控制项目的应用程序,包括成本估算、进度安排、协作和风险分析。

[来源: ISO/IEC/IEEE 24765:2017; ISO/IEC/IEEE 24765 FDIS, 3.3614]

3.3269

项目管理人员 project management staff

项目管理人员 特指 项目团队 (project team, 6952) 中一个特定的子集,他们是执行项目具体管理活动 (如制定计划、进度管理、沟通管理、风险管控等) 的专业成员。在某些小型项目中,项目管理人员可能覆盖了几乎全部的团队成员。他们通常隶属于 项目管理团队 (project management team, 6931) 并服务于项目经理 (project manager, PM), 是项目成功的关键执行力量。

[来源: ISO/IEC/IEEE 24765:2017, 3.3175]

3.3270

项目管理系统 project management system

项目管理系统 是指导和管理项目执行的综合性框架,它整合并规范了组织内或项目层面所采用的各种管理要素,包括管理过程、工具和技术、方法、资源以及程序。这个系统作为指导项目从启动到收尾全过程的操作基础,旨在系统地提升项目成功率与一致性,同时也为项目管理信息系统 (PMIS, 3928)、工作授权系统 (WAS, 5921)、变更控制系统 (change control system) 等具体子系统以及 项目治理 (6919) 提供了基本的环境与结构支持。

[来源: ISO/IEC/IEEE 24765:2017, 3.3176]

3.3271

项目管理团队 project management team

项目管理团队是项目团队 (6952) 的核心子集,特指直接负责执行项目核心管理职能 (如规划、组织、指导、控制) 的核心成员。这些成员直接向项目经理 (PM) 负责。核心成员通常包括项目经理 (PM)、项目管理人员 (6929), 并可能包括在管理职责的项目发起人。在小型项目中,由于角色复合度高,项目管理团队与项目团队的范围可能高度重叠。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3615]

3.3272

项目经理 project manager; PM; PJM

由执行组织指派、领导团队负责实现项目目标的人员。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3616]

3.3273

工程业绩 project performance

项目绩效是衡量项目执行情况的指标,它反映项目在多大程度上符合了其在预算、进度和技术 (范围与质量) 要求方面的规划目标。它是一个综合性的概念,通常是通过收集和分析工作绩效数据,利用诸如挣值管理 (EVM) 等技术,与绩效测量基线 (PMB) 进行比较而得出。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3617]

3.3274

项目阶段 project phase

项目阶段是项目生存周期中为完成一个或多个特定的主要可交付成果而进行的一系列逻辑相关联的项目活动集合。阶段通常以决策门 (如阶段评审) 作为结束,以决定是否进入下一阶段。它不同于项

目管理过程组。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3618]

3.3275

项目计划 project plan

项目计划是一个规定性文档，用于全面描述和规定项目执行所采用的技术与管理方法。作为项目管理的基准文档，是项目管理计划（project management plan, 6927）和软件开发计划（software development plan, 1490）等特定计划的广义父集或统称。一个典型的项目计划应包含要做的工作、所需的资源、需用的方法、遵循的程序、应满足的日程以及项目组织形式等核心内容。在更广泛的系统工程意义上，它是执行和协调项目规划过程（project planning, 6935）的主要输出成果。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3619]

3.3276

项目规划 project planning

项目规划是一个执行过程，旨在通过一系列活动（如范围定义、工作分解、进度制定、资源估算和风险管理规划等）来制定和协调一套有效且可行的计划（包括项目管理计划等），以指导项目的执行、监控和收尾。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3620]

3.3277

项目组合 project portfolio

项目组合是一个组织的战略资产集合，指为实现组织的战略目标而被定义、授权和管理的一组项目。它体现了组织在特定领域或时间段内的战略投资与资源配置决策。项目组合本身是**组合管理过程（Portfolio Management Process）的核心管理对象，并作为项目规划过程（Project Planning Process）**的关键输入，用于定义新项目的起始目标和范围。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3621]

3.3278

项目采购管理 project procurement management

项目采购管理是项目管理的一个知识领域，包含从采购策略规划到合同执行控制所需的全部过程，以确保从项目团队外部采购或获取产品、服务或成果来满足项目需求。其核心过程及产物包括：制定采购管理计划（procurement management plan）、编制采购工作说明书（procurement statement of work）、发布采购文件（procurement documents，如RFP/RFQ）、以及进行采购绩效审查等。通过有效管理合同与供应商关系，保障项目价值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3622]

3.3279

项目质量管理 project quality management

项目质量管理是项目管理中的一个知识领域或活动集合，它涵盖了制定质量方针、目标与职责所必需的所有过程和活动，以确保项目及其产出物能满足既定需求。它包括质量管理过程，如计划质量管理（Plan Quality Management）、执行质量保证（Perform Quality Assurance）和执行质量控制（Perform Quality Control）。其核心是通过质量规划工具（如QFD），制定质量管理计划（Quality Management Plan）并建立质量管理体系，对项目绩效进行持续的项目评估与控制。

[来源：ISO/IEC/IEEE 24765:2017, 3.3186]

3.3280

项目资源约束 project resource constraint

项目资源约束是对项目执行过程中资源（如特定技能的人力、设备、资金等）的使用或分配施加的

一种限制或约束，例如在特定时间段内可用资源的类型、数量或成本上限。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3281

项目资源管理 project resource management

项目资源管理是为确保成功完成项目而对所需资源进行识别、获取、分配和管理的一系列项目管理过程，它关注对项目活动和工作分解结构进行资源平衡和优化，并应对项目资源约束。

3.3282

项目风险 project risk

项目风险是与项目管理相关的风险，即可能对一个或多个项目目标（如范围、进度、成本和质量）产生正面或负面影响的不确定事件或条件，常见的例子包括人员短缺、时间期限紧迫和需求变更。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3624]

3.3283

项目风险管理 project risk management

项目风险管理是项目管理中的一个领域，它指在项目中为了处理和管理风险（包括威胁和机遇）所进行的活动集合，包含计划风险（如制定风险管理计划）、识别风险、分析风险、计划风险应对措施、实施这些应对措施，以及监控和控制风险等一系列在项目全生存周期中展开的持续过程。

[来源：ISO/IEC/IEEE 24765:2017, 3.3189]

3.3284

项目风险概况 project risk profile

项目风险概况是项目当前和历史风险相关信息的集合，是所有项目个体风险状况的概要或汇总。它包括相关的风险管理背景信息、按时间顺序排列的风险记录及其优先顺序、应对状态、应急计划、风险行动请求等。项目风险概况是所有个体风险状况的集合，这些个体风险状况又记录了其当前和历史状态。

[来源：ISO/IEC/IEEE 24765:2017, 3.3190]

3.3285

项目进度 project schedule

项目进度是进度模型的产物，它是一份列有项目里程碑、活动及可交付成果及其计划开始与完成日期的关联清单，它提供了项目各项任务的时间安排、资源分配及相互关系的信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3625]

3.3286

项目进度管理 project schedule management

项目进度管理是项目管理的一个知识领域，包含为确保项目按时完成所需的过程，例如计划进度管理、定义活动、排列活动顺序、估算活动持续时间、制定进度计划以及控制进度。

[来源：IEEE_CS_SE_Vocabulary-2021]

3.3287

项目进度网络图 project schedule network diagram

项目进度活动之间逻辑关系的图形显示。

[来源：项目管理知识体系指南(PMBOK® 指南)——第七版]

3.3288

项目进度网络图 project schedule network diagram

活动网络图 activity network diagram

展示项目进度活动之间逻辑关系的图形化呈现形式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3626]

3.3289

项目范围 project scope

项目范围是成功的项目所要求完成的全部工作，定义为为交付具有特定特性和功能的产品、服务或结果所必须执行的工作。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3627]

3.3290

项目范围管理 project scope management

项目范围管理是项目管理的一个知识领域，旨在确保项目包含并仅包含成功完成项目所必需的全部工作。它包括为确保项目范围得到有效定义、规划、监控与控制而进行的一系列过程，例如计划范围管理、定义范围、创建工作分解结构、验证范围以及控制范围。

[来源：ISO/IEC/IEEE 24765:2017, 3.3194]

3.3291

项目范围说明书 project scope statement

项目范围说明书是用于正式定义和描述项目范围、主要可交付成果、项目假设以及项目约束的文件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3628]

3.3292

项目规范 project specification

项目规范是描述项目目标、要求、范围及其与其他项目关系的规格说明文件。

[来源：ISO/IEC 2382:2015, 20.07.07]

3.3293

项目干系人管理 project stakeholder management

项目相关方管理，识别受项目影响的所有人员或组织所需的过程，分析相关方对项目的期望和影响，并制定适当的管理策略，以便有效地让相关方参与项目决策和执行。

[来源：ISO/IEC/IEEE 24765:2017, 3.3197]

3.3294

项目状态报告 project status report

为相关方准备的报告，旨在传达整个项目活动的状态、结果和成果。包括项目既定目标的达成情况、项目工作健康度和成熟度的信息、项目裁减和执行的状态，以及项目人员可用性和工作效率的状态。

[来源：INCOSE 系统工程手册, 4th ed., 附录 E]

3.3295

项目剪裁策略 project tailoring strategy

为指定项目整合和裁剪组织标准生存周期过程集所需的方法、工作计划、资源和特定注意事项。

[来源：INCOSE 系统工程手册, 4th ed., 附录 E]

3.3296

项目团队 project team

项目团队是指支持项目经理执行所有项目工作以实现项目目标的所有成员的总称，通常包含项目经理（PM）、项目发起人以及直接或间接对项目经理报告、并担负项目工作职责的人员。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3630-project team]

3.3297

项目团队目录 project team directory

项目团队目录是用于记录项目团队 成员及其角色和联系信息的正式文档，是项目沟通管理的重要工具，通常包含在项目沟通管理计划 中，作为团队结构、职责和联系方式的中央参考。 [来源：

ISO/IEC/IEEE 24765:2017, 3.3199-project team directory]。

3.3298

项目时间管理 project time management

项目时间管理是为确保项目按时完成而对所需进行的活动及过程进行规划、安排和控制的综合性管理领域。它包括定义活动、排列活动顺序、估算活动资源、估算活动持续时间、制定进度计划以及控制进度等一系列过程。

[来源: ISO/IEC/IEEE 24765:2017, 3.3200-project time management]

3.3299

项目愿景陈述 project vision statement

对项目的简洁、高层级描述,用以阐述项目目的并激励团队为之贡献力量。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3300

项目工作绩效领域 project work performance domain

与建立项目流程、管理实物资源及培养学习环境相关的活动和职能领域。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3301

基于项目的组织 project-based organizations

项目式组织。定义为: 涉及为执行项目而创建临时系统的一系列组织形式。项目型组织将其大部分活动作为项目进行, 并且/或者优先采用项目方式而非职能方式。

[来源: ISO/IEC/IEEE 24765:2017, 3.3201]

3.3302

项目化组织 projectized organization

项目化组织是一种组织结构, 其中项目经理在其项目职责范围内被赋予完全的、非共享的职权, 能够独立地分配优先级、调配资源并指导项目指派人员的工作, 是最具“项目导向”特征的典型组织形式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3633]

3.3303

项目定义的软件过程 project's defined software process (PDSP)

项目所采用的软件过程的操作性定义。项目定义的软件过程是一个妥善定义的软件过程, 它用软件标准、规程、工具和方法来描述, 并且通过剪裁组织的标准软件过程以适合项目的具体特征。

参见: 组织的标准软件过程

3.3304

序断点 prolog breakpoint

前同步码断点 preamble breakpoint

直至进入程序或例程启动的断点。同义词: 2.1159 条 前同步码断点 (preamble breakpoint)。

参见: 代码断点、数据断点、动态断点、可编程断点、静态断点

3.3305

序言断点 prologue breakpoint

进入程序或例程时触发的断点。

参见: 收尾断点、代码断点、数据断点、动态断点、可编程断点、静态断点。

[来源: ISO/IEC/IEEE FDIS 24765 3.3634]

3.3306

提示 prompt

由计算机系统显示的，请求从系统的用户输入的符号或消息。
程序发送的用于请求或引导用户响应的视觉或听觉消息。

3.3307

概念验证 proof of concept

对一个想法或技术的初步实现，以展示其可行性。

[来源：INCONE 系统工程手册, 第五版.]

3.3308

正确性证明 proof of correctness

用于从数学上证明计算机程序满足其指定要求的形式化技术。

参见：断言、正式规格说明、归纳断言方法、部分正确性和完全正确性

3.3309

量化属性 property to quantify

与质量测度元素相关，并能由某一测量方法予以量化的目标实体的性质。

注1：软件制品是目标实体的一种示例。

注2：子属性与属性相关。

[来源：GB/T 25000.21-2019, 4.11]

3.3310

利害关系属性 property-of-interest

其损失被视为负面影响的对象。

注：引入“利害关系属性”的概念是为了描述后果的负面影响。在安全领域，人的生命和健康是利害关系属性的实例。在安全背景下，资产也是利害关系属性的实例。

[来源：ISO/IEC 15026-1:2019, 3.4.12]

3.3311

投标书 proposal

供方为提供系统或服务而出具的要约文件，通常包含收益、成本、风险、机遇及其他影响决策的相关要素。

注：包含商业论证文件。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3642]

3.3312

提案评估技术 proposal evaluation techniques

提案评估技术是采购管理中用于审查供应商提交的提案的一套方法和过程，其目的是为合同授予决策提供支持。这些技术通常涉及对提案的完整性、技术方案、成本、商业条款、供应商能力及风险评估等进行系统性的比较和评价。

[来源：ISO/IEC/IEEE 24765:2017, 3.3212-proposal evaluation techniques]

3.3313

改进建议 proposed change

异常报告，从想法被记录到被指定变更机构处理期间所需或建议的改进。

注：处理方式可以是拒绝、推迟以进一步分析，或接受。一旦接受，改进建议即成为被批准的修改。改进建议与被批准的修改之间可能存在一对一、一对多或多对多的关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3643]

3.3314

命题/主张 proposition

涉及一个或多个实体的可观测事实或事态，可对其是否适用于这些实体作出肯定或否定的判定。

[来源：ISO/IEC 10746-2:2009, 6.2]

3.3315

保护 `protected`

仅对该类或该类的接收实例可见（仅在该类或其子类的方法内可用）。

参见：私有、公有、隐藏。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3645]

3.3316

保护异常 `protection exception`

当程序企图写至存储器中受保护的区域时引起的异常。

参见：寻址异常、数据异常、操作异常、溢出异常和下溢异常

3.3317

协议 `protocol`

支配系统内进程、设备和其他部件交互的一组约定。

3.3318

协议对象 `protocol object`

通道中的工程对象，该对象与同一通道中的其他协议对象进行通信，以实现基本工程对象之间的交互（这些对象可能位于不同的集群、不同的封装或不同的节点中）。

[来源：ISO/IEC 10746-3:2009, 8.1.12]

3.3319

原型 `prototype`

原型指系统或产品在开发过程中所创建的初步模型或早期可工作的实例，用于模拟、试验、评估最终产品的部分或全部特性。其主要目的在于：1) 作为后续开发或最终产品的参考模型；2) 作为一种需求获取与验证方法，通过提供可视、可交互的早期版本，从用户或相关方获取早期反馈，以改进人机界面、进行可行性研究或澄清需求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3650]

3.3320

原型法 `prototyping`

一种软硬件开发技术，通过开发部分或全部软硬件的初步版本，以获取用户反馈、确定可行性，或研究时间安排或其他问题，从而支持开发过程。

参见：快速原型法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3651]

3.3321

提供仪器活动 `provide instruments activity`

提供仪器活动。

[来源：ISO/IEC 29155-2:2013, 4]

3.3322

建议 `provision`

3.3323

伪指令 `pseudo instruction; pseudo-operation`

编译指示 `pragma`

伪操作 `pseudo-op`

为汇编程序或编译程序提供信息或方向而不转换为目标语言指令的源语言指令。同义词：2.1157 条 编译指示 (pragma)、2.1282 条 伪操作 (pseudo-op) 和 2.1281 条 伪操作 (pseudo operation)。

示例：规定所希望的源代码列表格式的指令。

[来源：GB/T 11457-2006, 2.1157]

3.3324

伪码 pseudocode; pseudo code

用于表达计算机程序设计的编程语言结构和自然语言的组合。例如：。

IF the data arrives faster than expected (若 数据到达比预料的快,)。

THEN reject every third input. (则在每当两个输入后 拒绝第三个输入。)

ELSE process all data received. (否则 处理接收到的所有数据。)

ENDIF。

3.3325

伪预言 pseudo-oracle

用于生成结果的测试项的独立派生变体，与基于相同测试输入的原始测试项的结果进行比较。

示例：一个已存在的系统、一个由独立团队开发的系统，或一个使用不同编程语言实现的系统。

[来源：ISO/IEC TR 29119-11:2020, 3.1.59]

3.3326

伪状态 pseudo-state

在状态机中，一种虽有状态形式却无状态行为的顶点。伪状态包括初始顶点和历史顶点。

3.3327

伪静态随机存取存储器 pseudostatic random access memory; PSRAM

带有自动刷新电路的动态随机存取存储单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3661]

3.3328

心理测验学 psychometrics

开发有效和可靠的心理学测量相关理论和技术的研究领域。

[来源：ISO/IEC 25022:2016, 4.14]

3.3329

公开的 public

为多个例程或模块所知。

注：对任何请求者可见（无限制地可供所有人使用）。

参见：隐藏、私有、受保护。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3666]

3.3330

公有链 public blockchain

各个节点可以自由加入或退出网络，并参与链上数据读写的区块链系统。

[来源：GM/T 0111:2021, 3.7]

3.3331

公有云 public cloud

云部署模式，其中云服务可能对任何云服务客户开放，而资源则由云服务提供商控制。

[来源：ISO/IEC 22123-1:2023, 3.2.5]

3.3332

已公布模型 published model

一种已予“冻结”，且对存储库实例化与支持地其他模型定义成为可用的模型。“已冻结”模型的模型元素不能变更。

3.3333

发布流水线 publishing pipeline

一系列已定义且相互关联的处理步骤，用于将内容从源格式转换为最终可交付格式。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.25]

3.3334

脉冲宽度调制 pulse width modulation; PWM

具有不同周期和占空比的电信号的控制。

[来源：ISO/IEC/IEEE 24765d:2015]

3.3335

计数目的 purpose of the count

执行功能点计数的原因。

[来源：ISO/IEC 20926:2009, 3.45]

3.3336

下推式存储器 pushdown storage

按后进先出（LIFO）的方法处理数据的存储器。在这种方法中下一个要取的项是仍在存储器中的最后存入的那个项。

[来源：GB/T 11457-2006, 2.1286]

3.3337

质量保证报告 quality assurance report

为相关方准备的报告，用于传达质量保证活动的状态、结果和成果。包括产品生存周期中与标称条件偏离的信息，以及当质量保证目标和目的未达成时应采取的行动。

[来源：INCONE 系统工程手册, 4th ed., 附录 E]

3.3338

资格认证 qualification

证明实体是否能够满足规定要求的过程。

[来源：ISO/IEC/IEEE 12207:2017, 3.1.39]

3.3339

鉴定（系统） qualification (system)

证明实体是否有能力满足特定要求的过程。

注：该过程包括测试和分析硬件和软件配置项，以证明设计能够经受住预期累积的验收测试环境，以及预期的搬运、储存和操作环境，并具有指定的鉴定裕度。

[来源：ISO/IEC/IEEE 12207:2026, 3.1.51];

3.3340

资格认证机构 qualification body

颁发资格证书的实体。

[来源：ISO/IEC/IEEE 24765 系统与软件工程——词汇表, 3.3687]

3.3341

鉴定限制 qualification limit

证明设计在预期环境中具有足够的裕度。该过程包括测试和分析硬件和软件配置项，以证明设计能够经受住预期累积的验收测试环境，以及预期的搬运、储存和操作环境，并具有指定的鉴定裕度。

[来源：INCONE 系统工程手册, 4th ed., APPENDIX C]

3.3342

鉴定裕度 qualification margin

为应对认证不确定性而预留的安全余量。

3.3343

鉴定需求 qualification requirement

一组必须满足的标准或条件，用于判定一个软件产品是否符合其规格说明，并已准备好在其目标运行环境中投入使用，或与其所归属的更大系统完成集成。

[来源：ISO/IEC 12207:2008, 4.32]

3.3344

资格认证方案 qualification scheme

满足后即可颁发资质证书的一系列要求。

[来源：ISO/IEC 24773-1:2019, 3.15]

3.3345

合格性测试 qualification testing

对硬件元素、软件元素或系统进行的测试，以评估是否符合规定要求。

参见：验收测试、开发测试、运行测试。

[来源：IEEE 1012-2024, 3.1]

3.3346

认证（人员） qualification/qualified (person)

特定的专业认可、头衔或标志，可能表明在特定领域的熟练程度、技能或知识，但仅一次性完成。

[来源：ISO/IEC 24773-1:2019, 3.14]

在适用的情况下，被证实的教育、培训和工作经历。

[来源：ISO/IEC 24773-1:2019, 3.13]

3.3347

资格人员 qualified personnel

具备代表组织履行职责能力的人员，包括官员、员工和承包商。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3348

限定符 qualifier

一种关联属性或属性元组，其值将跨某一关联与一个对象有关的对象集加以划分。

3.3349

定性风险分析 qualitative risk analysis

通过评估并结合风险发生的可能性和影响程度，对风险进行优先级排序，以便后续进一步分析或采取应对措施。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3690]

3.3350

质量 quality

产品、服务、系统、组件或过程满足顾客或用户需求、期望或要求的程度。

[来源：ISO/IEC TR 7052:2023, 3.35]

3.3351

质量分析 quality analysis

通过对多个质量特性的评定结果进行分析，以确定目标实体质量的客观评分或可接受性。

[来源：ISO/IEC 25040:2024, 3.6]

3.3352

质量保证 quality assurance; QA

质量管理的一部分，致力于提供质量要求会得到满足的置信度。

参见：质量管理、质量控制。

注：质量保证既有内部目的，也有外部目的：在组织内部，质量保证为管理层提供信心；在合同情境中，质量保证为客户或其他方提供信心。一些质量控制和质量保证措施是相互关联的。除非质量要求能充分反映用户的需求，否则质量保证不一定能提供足够的信心。

[来源：ISO/IEC/IEEE 15288:2023, 3.34]

3.3353

质量保证评估报告 quality assurance evaluation report

为相关方准备的报告，旨在传达项目质量保证活动是否有效的证据。包括对所有项目相关流程的评估，以及任何建议的改进措施或必要的纠正措施。为改进组织的生存周期模型实施提供建设性意见。

[来源：INCONE 系统工程手册, 4th ed., 附录 E]

3.3354

质量保证计划 quality assurance plan; QAP

为项目量身定制的项目质量保证活动集，旨在监控开发和系统工程（SE）过程。描述了质量保证组织以及适用的审计、评估和监控活动。包括一系列政策和程序，包括适用于组织内部和单个项目内部质量保证实践的具体方法和技巧，还包括可衡量的流程和系统的质量目标，以及与组织内部质量管理（QM）所分配的责任和权限的联系。该计划还参考了由质量保证组织监控或审计的其他组织或职能部门所执行的活动。

[来源：INCONE 系统工程手册, 4th ed., 附录 E]

3.3355

质量保证记录 quality assurance record

与质量保证相关的数据、信息或知识的永久、可读形式。

[来源：INCONE 系统工程手册, 4th ed., 附录 E]

3.3356

质量属性 quality attribute

软件的特性，或适用于质量因素的通用术语。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3694]

3.3357

质量特性 quality characteristic

与要求有关的，产品、过程或系统的固有属性。

注：严格意义上的质量特性一般包括与健康、安全性、信息安全性、可靠性、可信性、可用性、支持性相关的内容。

[来源：GB/T 19000-2008, 3.5.2, 有修改]

3.3358

(软件产品或系统的) 质量特性 quality characteristic (of software product or system)

支撑软件产品或系统质量的质量属性的类别。

[来源：ISO/IEC 25000:2014, 4.34, 有修改]

3.3359

质量核对单 quality checklist

用来核实所要求的一系列步骤是否已得到执行的结构化工具。

[来源：项目管理知识体系指南（PMBOK® 指南）——第六版]

3.3360

质量控制 quality control; QC

〈软件与系统工程〉用于评价已开发或已生产产品质量的一组活动。

〈信息技术服务〉监视服务绩效或产品质量，记录结果，并提出必要的改进建议。

参见：质量控制、质量保证。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3696]

3.3361

质量控制措施 quality control measurements

对质量控制活动的结果的书面记录。

[来源：ISO/IEC/IEEE 24765:2017, 3.3266]

3.3362

质量评价 quality evaluation

系统检查一个实体能够满足指定要求的程度。

[来源：ISO/IEC 25040:2024, 3.7]

3.3363

品质因数 quality factor

质量因数

有助于软件质量的面向管理的属性。

参见：质量属性（quality attribute）。

注：在质量属性的层次中，高级属性可能称为质量因数，低级属性，称为质量属性。

[来源：IEEE 1061-1998 (R2004), 2.18]

3.3364

质量因子样本 quality factor sample

从度量数据库中提取的一组质量因子数值，专门用于度量验证。

未找到属于来源。

3.3365

质量因子值 quality factor value

将软件质量因子（比如可靠性、性能、易用性、可维护性等）量化后得到的具体数值。

3.3366

质量功能部署 quality function deployment; QFD

一种将用户质量需求，逐步变成可落地、能实现的具体技术方法。

3.3367

使用质量（测度） quality in use (measure)

系统或产品在指定使用情境下使用时，满足或超出相关方需求以达成指定有益目标或结果的程度。

[来源：ISO/IEC 25019:2023]

3.3368

使用质量测度 quality in use measure

在指定使用周境中，系统和软件产品由指定用户使用时，达到其规定目标并能满足用户要求的程度

的测度。所规定目标包括有效性、效率、抗风险、周境覆盖和满意度等。

[来源：GB/T 25000.1-2021，3.32]

3.3369

质量管理 quality management; QM

在质量方面指挥和控制组织的协调的活动。[GB/T 19000-2008，3.2.8]。

[来源：GB/T 22032-2021（ISO/IEC/IEEE 15288:2015，IDT）]

3.3370

质量管理和控制工具 quality management and control tools

质量规划工具的一个类别，用来分析已识别活动的相互关联和顺序。

[来源：ISO/IEC/IEEE 24765:2017，3.3275]

3.3371

质量管理纠正措施 quality management corrective action

未达成质量目标时采取的措施。这些措施源于项目相关和过程相关的评审和审计。

[来源：INCONE 系统工程手册，4th ed.，附录 E]

3.3372

质量管理评估报告 quality management evaluation report

为相关方准备的报告，用于传达组织质量管理活动是否有效的证据。包括对所有组织相关流程的评估，以及任何建议的改进措施或必要的纠正措施。为改进组织生存周期模型的实施提供建设性意见。

[来源：INCONE 系统工程手册，4th ed.，附录 E]

3.3373

质量管理指南 quality management guideline

组织内部、单个项目内部以及作为系统生存周期过程执行一部分的质量实践指南。

[来源：INCONE 系统工程手册，4th ed.，附录 E]

3.3374

质量管理计划 quality management plan

项目或项目集管理计划的组件，描述如何实施适用的政策、程序和指南以实现质量目标。

[来源：ISO/IEC/IEEE FDIS 24765，3.3701]

3.3375

质量管理记录 quality management record

与质量管理相关的数据、信息或知识的永久、可读形式。

[来源：INCONE 系统工程手册，4th ed.，附录 E]

3.3376

质量管理报告 quality management report

为相关方准备的报告，用于传达质量管理活动的状态、结果和成果。包括任何客户满意度调查的结果以及任何需要解决的问题。

[来源：INCONE 系统工程手册，4th ed.，附录 E]

3.3377

质量测度 quality measure; QM

至少两个质量测度元素值的测量函数的导出测度。

[来源：GB/T 25000.21-2019，4.13]

3.3378

质量测度元素 quality measure element; QME

依据质量的某一性质及对其量化的测量方法所定义的测度。测量方法可以包括数学函数变换等。

注：通过计算一个软件质量测度，而后导出实体的软件质量特性或子特性。

[来源：GB/T 25000.1-2021]

3.3379

外部性质的质量测度 quality measure on external property

在特定条件下使用时，系统或软件产品使其行为能满足系统（包括软件）的明确和隐含需要程度的测度。

示例：在测试期间发现的失效数是一种与存在于计算机系统故障数有关的软件质量外部测度。这两种测度不一定是相同的，因为测试不能发现所有的故障；并且在不同的情况下，某一故障会导致明显不同的失效。

注：在测试和运行期间，通过执行系统或软件产品来测量、验证和/或确认行为的属性。

[来源：GB/T 25000.20-2021, 3.2]

3.3380

内部性质的质量测度 quality measure on internal property

在指定条件下使用时，软件产品的静态属性满足明确和隐含需要程度的测度。

示例：在走查中发现的复杂度和故障的数量、严重程度以及失效频率，是从软件产品自身取得的软件内部质量测度。

注1：静态属性包括那些与软件架构、结构及其组件、数据结构及其格式、屏幕上图形显示的结构和外观以及用户或服务接收者的菜单有关的属性。

注2：静态属性可通过评审、审查、模拟和/或自动化工具来验证。

注3：内部性质的质量测度通常与可在需求中规定或从需求中派生的静态性质和属性的质量需求相关。

[来源：GB/T 25000.20-2021, 3.3]

3.3381

质量计量 quality metric

〈质量管理〉对一个实体具备某一给定质量属性程度的定量测量。

〈软件与系统工程〉以软件数据为输入、输出为单个数值的函数，该数值可表示软件具备某一给定质量属性的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3706]

3.3382

质量模型 quality model

为规定质量需求和评价质量提供一种框架而定义的特性及其间关系集合。

[来源：GB/T 25000.1-2021, 3.35]

3.3383

服务质量 quality of service

对一个或多个对象的整体行为所提出的一组质量要求。

[来源：ISO/IEC 10746-2:2009]

3.3384

质量政策 quality policy

组织实施质量管理体系时应遵循的一套基本原则。[来源：项目管理知识体系指南（PMBOK®指南）第七版]。

3.3385

质量特性 quality property

与质量测量元素相关、且可通过测量方法进行量化的目标实体性质。

[来源：ISO/IEC 25002:2024, 3.22]

3.3386

质量评定模块 quality rating module

针对特定类别目标实体开展质量测量与质量评级所用的一套质量度量指标、运行环境及实施方法。

[来源: ISO/IEC 25040:2024, 3.8]

3.3387

质量记录报告 quality record report

通过动态测试执行和代码分析产生,用于记录测试结果和其他输出的报告。

注:包含测试结果、静态代码分析报告、测试事件报告和测量等内容。

[来源: GB/T 41905-2022, 3.1]

3.3388

质量报告 quality report

包含质量管理问题、纠正措施建议以及质量控制活动结果总结的项目文件,也可包含对过程、项目和产品改进的建议。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3389

质量需求 quality requirement

针对某一 ICT 产品、数据或服务的质量属性,为满足使用目的要求随之产生的需求。

[来源: GB/T 25000.30-2021, 3.15, 有修改]

3.3390

质量子特性 quality subcharacteristic

表示质量特性独特方面的单个或多个质量属性的集合。

注:质量子特性有助于将质量特性细分为各个具体方面,便于将其映射到质量属性。

[来源: ISO/IEC 25002:2024 3.24]

3.3391

质量子因素 quality subfactor

将质量因素或质量子因素分解为其技术组件。

[来源: IEEE 1061-1998 (R2009) IEEE, 2.22]

3.3392

定量控制 quantitative control

适于分析软件过程、标识软件过程性能变化的特殊原因和使软件过程性能处于妥善定义的范围内的任何一种定量技术或基于统计的技术。

[来源: GB/T 11457-2006, 2.1302]

3.3393

定量风险分析 quantitative risk analysis

对已识别风险对项目总体目标影响的数值化分析。

[来源: ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表, 3.3717]

3.3394

量子计算 quantum computing

可在量子计算机上进行的计算。

[来源: ISO/IEC 4879:2024, 3.4.11]

3.3395

查询语言 query language

用于访问存储在数据库中的信息的语言。与 X. XXXX 条 编程语言(programming language)和 X. XXXX 条 规格说明语言(specification language)相对。

[来源: GB/T 11457-2006, 2.1303]

3.3396

队列 queue

一种把项添加到它的最后位置而从它的第一个位置检索的列表。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3719]

3.3397

静寂 quiescing

由拒绝工作的新请求使设备或系统暂停的过程。

[来源: GB/T 11457-2006, 2.1305]

3.3398

随机失效 random failure

故障的发生是不可预测的,除非在概率或统计意义上比照。

参见: X. XXX 条 间歇故障(intermittent fault)和 X. XXXX 条 瞬时错误(transient error)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3727]

3.3399

随机测试 random testing

基于规格说明的测试设计技术,通过生成测试用例,随机选取测试项输入执行测试。

[来源: ISO/IEC/IEEE 29119-4:2021, 3.41]

3.3400

随机存取存储器 random-access memory; RAM

一种易失性半导体存储器件,无论数据物理位置如何,都能以大致相同的时间写入或访问数据。

注:常用于计算机的高速缓存或主存,也集成在微控制器(MCU)芯片中。与按顺序存储和访问数据的光盘(CD)形成对比。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3729]

3.3401

快速原型法 rapid prototyping

一种原型开发方式,强调在开发过程的早期阶段快速构建原型,以便尽早获得用户反馈并开展分析,从而支持后续的开发过程。参见:瀑布模型、以数据结构为中心的设计、增量式开发、输入-处理-输出、模块化分解。[来源: ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表, 3.3730]。

3.3402

单调速率算法 rate-monotonic algorithm

一种实时调度算法,该算法为周期较短的任务分配更高的优先级。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.3403

评定 rating

把与系统和软件产品质量特性相关的测量值映射到相应评定级别的活动。

[来源: GB/T 25000.1-2021, 3.36]

3.3404

评级区间 rating interval

从被测系统(SUT)达到稳定运行状态开始,到测量结果满足所需统计显著性为止的时间区间。

[来源: ISO/IEC 14756:1999 信息技术——基于计算机的软件系统性能的测量与评定, 4.13]

3.3405

评定级别 rating level

在有序标尺上对测量标度进行分级的刻度点。

注1: 评定等级能使软件产品按照明确或隐含的要求进行分类(评定)。

注2: 合适的评定等级与质量的不同视角有关, 如“用户”、“管理者”或“开发者”的视角。

[来源: GB/T 25000.1-2021]

3.3406

比例尺度 ratio scale

一种尺度, 其中测量值之间的等距对应于属性的等量, 其中零值对应于无该属性。

参见: 区间尺度、名义尺度、序数尺度。

注: 例如, 以 LOC(行数)为单位的软件组件大小就是一种比例尺度, 因为零值对应于无代码行, 且每增加一个单位代表相等数量的代码。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.3407

统一软件开发过程 rational unified process; RUP

一种软件工程过程, 它为开发组织内的任务分配和职责划分提供了一种规范化的方法。其目标是在可预测的进度和预算内, 确保生产出满足最终用户需求的高质量软件。统一软件开发过程是一个以用例为驱动、以架构为中心、迭代且增量式的软件开发过程框架。

[来源: The Rational Unified Process: An Introduction (3rd Edition), 2003]

3.3408

可达性图 reachability graph

一种由节点和有向边构成的有向图, 其中节点对应可达标识, 有向边对应变迁的发生。

[来源: ISO/IEC 15909-1:2019, 3.30]

3.3409

可达集合 reachability set

网络可达标记集, 包括初始标记。

[来源: ISO/IEC 15909-1:2004, 2.1.14.5]

3.3410

可达标识 reachable marking

从初始标识经由变迁发生所能到达的网标识。

[来源: ISO/IEC 15909-1:2019, 3.31]

3.3411

重新激活 reactivation

集群停用之后对其进行实施克隆操作。

[来源: ISO/IEC 10746-3:2009, 8.1.26]

3.3412

响应式方法 reactive approach

根据规定的需求或客户要求开发产品线或产品变体的方法。[来源: ISO/IEC 26553:2018 3.16]。

3.3413

反应性 reactivity

IT 服务对用户请求的及时响应程度。

[来源: ISO/IEC TS 25011:2017 , 3.2.6.3]

3.3414

读弧 read arc

一种特殊类型的弧,用于测试相关位置是否存在某些令牌,无需消耗令牌。

[来源: ISO/IEC 15909-3:2023.6]

3.3415

读 read

读取类型 read type

将数据组从持久存储器中移动到需要这些数据的功能过程内部的一种数据移动类型。

注:读一般包括实现读操作所需的某些关联数据操纵活动。

[来源: GB/T 42452-2023]

3.3416

可读性 readability

系统源代码易于阅读和理解的程度,尤其是在详细的语句级别上。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3754]

3.3417

图注 reader note

读者对图所做并放在图页上的注解。

注:图注不是图本身的一部分,而是用于在模型开发期间对图进行沟通。

[来源: ISO/IEC/IEEE 24765m:2024, 2.1.97]

3.3418

读引用 reading reference

来自另一软件或系统(其包含在BFC中检索到的数据)的数据存储实体或记录,或者接口记录。

注:对于所有可用的BFC类型,其读引用的数量大于或等于0。

[来源: GB/T 42448-2023 FiSMA1.1 方法]

3.3419

只读 read-only

不会引起状态变化的属性。

参见: read only。

注:即它不会引起任何状态变化。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.155]

3.3420

只读存储器 read-only memory; ROM

非易失性半导体存储设备,数据一旦写入就不能从中删除。

[来源: ISO/IEC/IEEE FDIS 24765 3.3751]

3.3421

准备好的 ready

在开始某项指定活动或任务之前,待办事项或其他工作成果所应满足的必备质量属性说明。

参见: done。

[来源: ISO/IEC 33202:2024, 3.14]

3.3422

实地址 real address

虚拟存储系统的主存储部分中存储位置的地址与 X. XXXX 条 虚拟地址 (virtual address) 相对。

[来源: GB/T 11457-2006, 2. 1311]

3. 3423

实存储器 real storage

虚拟存储系统的主存储部分与 X. XXXX 条 虚拟存储器 (virtual storage) 相对。

[来源: GB/T 11457-2006, 2. 1312]

3. 3424

实型 real type

一种数据类型, 其成员可以采用实数作为值, 并且可以通过实数算术运算进行运算, 例如, 加、减、乘、除和开方进行运算。

参见: 字符型、枚举型、整型、逻辑型

[来源: GB/T 11457-2006, 2. 1314]

3. 3425

实现 realization

通过指定算法和表示属性表示接口职责。

注: 实现说明责任“如何”得到满足; 它是职责方法的声明。实现包括任何必要的表示属性以及算法 (如果有的话)。

实现可以涉及表示属性或算法, 或两者兼而有之。

[来源: ISO/IEC 26557:2016, 3. 10]

3. 3426

实时的 real-time

指一种系统或操作模式, 在该模式中, 计算是在外部进程实际发生的同时进行的, 以便计算结果可用于及时控制、监视或响应外部进程。

[来源: ISO/IEC/IEEE 24765:2017, X. X]

3. 3427

实时时钟 real-time clock

以人类时间单位追踪当前时间的集成电路。

参见: real time clock。

[来源: ISO/IEC/IEEE 24765d:2015]。

3. 3428

实时操作系统 (RTOS) real-time operating system; RTOS

用于在接收到交易请求时立即处理的操作系统。

[来源: ISO/IEC/IEEE FDIS 24765, X. X]

3. 3429

实时操作系统 (RTOS) real-time operating system (RTOS); RTOS

一种操作系统, 旨在收到交易请求后立即处理。

[来源: ISO/IEC/IEEE 24765c:2014]

3. 3430

实时调度理论 real-time scheduling theory

针对硬实时截止期并发任务的优先级调度理论。

注: 它解决了如何确定一组任务 (其各自的 CPU 利用率已知) 是否会满足其期限。

[来源: ISO/IEC/IEEE 24765:2017, X. X]

3. 3431

真实世界的对象 **real-world object**

以三维形式存在的实体，通过关联暗示与软件功能相似的属性或行为。

示例：打印机、文件柜、文件夹和一张纸。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3766]

3. 3432

合理可预见的误用 **reasonably foreseeable misuse**

以非供应商预期的方式使用产品或系统，但这种使用方式可能源于易于预测的人类行为。

注：可预见的人类行为涵盖各类用户的操作行为。

[来源：IEC/IEEE 82079-1:2019 3. 30]

3. 3433

推理技术 **reasoning technique**

使用演绎和归纳等逻辑技术，从可用信息中得出结论的一种人工智能形式。

[来源：ISO/IEC TR 29119-11:2020, 3. 1. 60]

3. 3434

召回率 **recall**

用于评估分类器的性能指标，衡量实际正例中被正确预测的比例。

[来源：ISO/IEC TR 29119-11:2020, 软件和系统工程——软件测试——第 11 部分：测试指南基于 AI 的系统, 3. 1. 61] 同义词：灵敏度

3. 3435

容器 **receptacle**

在计算组件中扮演客户端角色的操作接口。

[来源：ISO/IEC 10746-3:2009, 7. 1. 16]

3. 3436

推荐性条款 **recommendation**

提供建议或指导的条款。

[来源：ISO/IEC 14143-2:2011, 3. 9]

3. 3437

记录元素类型 (RET) **record element type; RET**

数据功能中用户可识别的数据元素类型子集。

[来源：GB/T 42449-2023, 1. 754]

3. 3438

记录元素类型 (RET) **record element type (RET); RET**

记录元素类型 (RET)

数据功能中用户可识别的数据元素类型子组。

[来源：ISO/IEC 20926:2009 , 3. 46]

3. 3439

记录类型 **record type**

逻辑文件中的实体类型。

[来源：GB/T 42588-2023 (ISO/IEC 24570:2018) 系统与软件工程 功能规模测量 NESMA 方法]

3. 3440

记录管理系统 **records management system**

识别、存储、检索和保留数据的过程、相关控制功能和工具。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3775]

3.3441

可恢复性 recoverability

产品在发生中断或故障时，恢复直接受影响的数据并重建系统预期状态的能力。

[来源：ISO/IEC 25010:2023, 3.5.4]

3.3442

递归 recursion

〈软件工程〉软件模块调用它自己的过程。

〈过程管理〉以它自己的方式定义或生成一进程或数据结构的过程。

参见：同时递归

[来源：GB/T 11457:2006, 2.1320]

3.3443

递归的 recursive

软件模块调用它自己的过程。

参见：同时递归

[来源：ISO/IEC/IEEE FDIS 24765, 3.3779]

3.3444

递归例程 recursive routine

一种例程。它可以作为自己的子例程来使用，它直接调用它自己或被它所调用的另一子例程所调用。使用递归例程通常要求在某处（例如下推表中）保持其尚未完成的状态的记录。

[来源：GB/T 11457-2006, 2.1322]

3.3445

冗余 redundancy

在容错系统中，系统中存在辅助的部件执行像其他元素同样的功能以防止失败或从失败恢复。参见：2.34 条 活动冗余（active redundancy）、2.493 条 多样性（diversity）、2.707 条 同构冗余（homogeneous redundancy）和 2.1565 条 备用冗余（standby redundancy）。

[来源：GB/T 11457-2006, 2.1323]

3.3446

再工程 reengineering

对软件进行审查和修改以重构其新形态，包括后续对新形态的实施。

[来源：ISO/IEC TR 19759:2016, 5.4.2]

3.3447

重入的 reentrant

可重入的 reenterable

属于一个软件模块，可以作为一个过程的一部分输入，同时也可以作为另一个过程的一部分执行，并且仍能达到预期的结果。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3782]

3.3448

重入点 reentry point

软件模块中在调用另一个模块后重新进入该模块的位置。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3783]]

3.3449

重构 refactor

为改进质量属性、便于未来扩展或调整、或遵循某种架构风格，在不改变其行为的前提下对软件代码进行重组。

参见：分解。

[来源：项目管理知识体系指南（PMBOK(R) Guide）-第五版]

3.3450

基准 reference

对模型元素的一种指称。

在类目之中，便于对其他类目导航的一种命名位槽。

参见：指针

[来源：GB/T 11457-2006, 2.1327]

3.3451

参考架构 reference architecture

体现领域架构和应用架构高层级架构概念的核心架构。

[来源：ISO/IEC 26552:2019, 3.9]

3.3452

参考知识体系 reference body of knowledge

用于对与某一认证体系相关的特定知识体系进行比对的知识体系。

[来源：ISO/IEC 24773-1:2019, 3.12]

3.3453

参考框架 reference framework

用于理解某些环境下实体之间重要关系以及用于制定支持该环境的连贯一致的标准或规范的结构。

注：参考框架提供了一个通用背板，以实现一致性、协作、共享和复用。

[来源：GB/T 45615-2025, 1.955]

3.3454

参考FSM方法 reference FSM method

参考有限状态机方法

用于在验证功能规模测量结果时进行比较的功能规模测量方法。

[来源：ISO/IEC TR 14143-4:2002, 3.3]

3.3455

参考信息 reference information

旨在为已熟悉产品功能的用户快速提供特定细节的信息。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.43]

3.3456

参考模式 reference mode

旨在为通常熟悉软件功能的软件用户提供快速访问特定信息的使用模式。

[来源：ISO/IEC/IEEE 26514:2022, 4.42]

3.3457

参考结点 reference node

Petri 网的节点代表另一个节点，可能在网图的另一页上定义。

[来源：ISO/IEC 15909-2:2011, 4.1.14]

3. 3458

参考位置 reference place

代表一个地方并指代另一个参考地方或一个地方的参考节点。

[来源: ISO/IEC 15909-2:2011, 4.1.15]

3. 3459

参照点 reference point

架构中定义的交互点, 用于选择作为符合该架构的规范中的一致性点。

[来源: ISO/IEC 10746-2:2009, 10.6]

3. 3460

参考转换点 reference transition

表示一个转换并引用另一个转换的参考节点。

[来源: ISO/IEC 15909-2:2011, 4.1.16]

3. 3461

参考用户需求集合 reference user requirement collection

为满足特定评估目的而选定的参考用户需求子集。

[来源: ISO/IEC TR 14143-4:2002, 3.5]

3. 3462

参考用户需求 reference user requirements; RUR

符合要求的标准用户需求集。

[来源: ISO/IEC TR 14143-4:2002, 3.4]

3. 3463

参照完整性 referential integrity

确保关系的所有指定条件都成立。

示例: 如果一个类被声明为需要至少一个相关状态类的实例, 那么允许一个没有这种关系的实例将是无效的。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3.3796]

3. 3464

精化 refinement

细分

将一个规范转化为更详细规范的过程。

[来源: ISO/IEC 10746-2:2009, 9.6]

3. 3465

反映型构念 reflective construct

在构念与其度量指标之间的关系中, 被视为度量依据来源的那个构念。

[来源: ISO/IEC 33003:2015, 3.13]

3. 3466

自反的 reflexive

在关系中, 指同一数据对象同时扮演两个(二元关系)或多个(n元关系)角色时所处的状态。

参见: 递归。

[来源: ISO/IEC 15476-4:2005, 6.5]

3. 3467

自反祖先(一个类) reflexive ancestor (of a class)

(一个类的)自反祖先

类本身及其任一泛型祖先。

参见：泛型祖先。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.158]

3.3468

刷新 refresh

通过重写数据以防止其从内存中消失，从而将数据保持在易失性存储器中的方法。

注：对于具有单稳态电路架构的内存，需要此操作。

[来源：ISO/IEC/IEEE 24765c:2014]

3.3469

寄存器 register

登记册

处理单元中的小型存储单元。

注：寄存器可设置于CPU、微控制器、数字信号处理器或微处理器中。

[来源：ISO/IEC/IEEE 24765c:2014]

3.3470

寄存器组 register bank

在微处理器中集成的一组寄存器。

[来源：ISO/IEC/IEEE 24765e:2015)]

3.3471

注册标识符 registration identifier; regid

实体的唯一标识符。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3806]

3.3472

注册表 registry

用于保存工作产品及其关联信息项的官方列表或记录的书籍或系统。

注：存储库和库中的条目可以记录在注册表中，以便更好地管理和治理这些条目。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3807]

3.3473

回归 regression

针对给定输入产生数值或连续输出值的一种机器学习函数。

[来源：ISO/IEC TR 29119-11:2020, 3.1.62]

3.3474

回归分析 regression analysis

一种分析技术，通过检查一系列输入变量与其相应输出结果之间的关系，来建立数学或统计关系模型。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.3475

回归测试 regression test

为检测因修改而引入的缺陷所进行的重新测试。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3810]

3.3476

正则表达式 regular expression; Regex

用于匹配搜索结果的字符串模式。

示例: `^admin*` - 找到所有以 ‘admin’ 开头且后续包含任意字符序列的匹配项、`\d{5}$` - 找到所有以数字 5 结尾的匹配项、`^[0-9() -]+$` - 找到包含 10 位电话号码的匹配项。

注: 模式可指定匹配项以特定字符序列开头或结尾, 或允许使用通配符匹配序列中的任意字符。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.26]

3.3477

正则表达式 (Regex) **regular expression (Regex)**

正则表达式

一种允许使用模式来匹配搜索结果的字符串。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.26]。

示例: `^admin*`匹配所有以 ‘admin’ 开头、后续包含任意字符序列的内容, `\d{5}$`匹配所有以 5 位数字结尾的内容, `^[0-9() -]+$`匹配仅包含数字、括号、连字符的内容 (可用于匹配 10 位电话号码格式)。

注: 模式可以规定匹配结果以特定的字符序列开头或结尾, 或允许使用通配符来匹配序列中的任意字符。

3.3478

监管背景 **regulatory context**

系统可以运行的司法管辖区内的法律、法规或任何其他要求。

[来源: IEEE 7002:2022, 3.1]

3.3479

强化学习 **reinforcement learning**

使用试错和奖励的过程来构建机器学习模型, 以实现特定目标的任务。

注: 强化学习任务可包括以类似于监督学习的方式训练机器学习模型, 并利用在人工智能系统运行阶段收集的未标记输入进行训练。每当模型做出预测时, 系统会计算一次奖励, 并通过进一步的试错来优化该奖励。在强化学习中, 目标或成功的定义可由系统设计者设定。在强化学习中, 奖励可以是一个计算得出的数值, 代表人工智能系统在特定试错中距离实现该目标的接近程度。

[来源: ISO/IEC TR 29119-11:2020, 3.1.64]

3.3480

关系 **relation**

相同关系类型的一组关系。

[来源: ISO/IEC 14769:2001, 3.2.3]

两个或多个实体域之间的关联。

[来源: ISO/IEC 10746-2:2009, 8.13]

3.3481

关系数据库管理系统 **relational database management system; RDBMS**

关系数据库管理系统。

注: 为了使用关系型数据库管理系统 (RDBMS), 必须描述一个关系数据模型, 该模型用特定特征 (如表、唯一键) 来组织数据。

[来源: ISO/IEC 25024:2015, 4.34]]

3.3482

联系, 关系 **relationship**

一种模型元素之间的语义连接。

示例: “关联 ‘’ 泛化 “。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3819, 有修改]

3.3483

关系实例 relationship instance

相关类的特定实例之间的关联。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.160]

3.3484

关系经理 relationship manager; RM

负责开发和管理与客户或供应商互动的角色。

[来源: ISO/IEC 29110-1-2:2024, 3.75]

3.3485

关系名称 relationship name

反映图中所示两个实体之间所表达关系含义的动词或动词短语。

注: 采用关键样式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.161]

3.3486

关系类型 relationship type

表达角色的数量和类型的关系类别。

[来源: ISO/IEC 14769:2001, 3.2.2]

3.3487

相对地址 relative address

通过加上偏移量进行调整以确定要访问的存储位置地址的地址。参见: 绝对地址、基地址、变址地址、自相对地址。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3824]]

3.3488

相对链频率 relative chain frequency

第 i 类模拟用户使用第 1 类链路类型的相对频率。[来源: ISO/IEC 14756:1999 信息技术——基于计算机的软件系统性能测量与评级, 4.14]。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3825]]

3.3489

相对估计 relative estimating

在综合考虑了工作量、复杂度和不确定性因素后, 通过与同类工作进行对比来生成估算值的方法。

[来源: PMBOK® 指南 7th Edition]

3.3490

相对装入程序 relative loader

将可重定位代码读入主内存并调整代码中的地址以反映代码已加载的存储位置的加载器。参见: 绝对加载器来源[来源: ISO/IEC/IEEE FDIS 24765, 3.3846]。

3.3491

相对时间 relative time

一种根据非绝对时间的参考活动递增的计时源。

参见: 绝对时间, 自然单位。

示例: 在多核虚拟化环境中执行某应用进程所消耗的 CPU 秒数, 或某一地区惯用的业务时段与工作日。

[来源: IEEE 982:2024, 3.1]

3. 3492

发布 release

一项配置管理行为，它说明某配置项的一个特定版本已准备好用于特定的目的（例如发布测试产品）。

[来源:GB/T 11457-2006, 2.1333]

3. 3493

发布待办事项清单 release backlog

计划在下一版本中实施的产品待办事项清单子集。

[来源: ISO/IEC 33202:2024, 3.24]

3. 3494

发布地图 release map

关于软件功能何时发布以及如何将功能分组到各个版本中的展示性预测。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3832]

3. 3495

发布包 release package

包含软件系统安装或更新所需配置项及其他技术文档的集合。

注：通常包含应用软件介质、技术文档、工具程序、配置文件、授权材料及安装工具。这些组件通常与目标系统的唯一配置标识符或版本号相关联，并可通过许可证密钥或其他用户/目标环境标识符进一步识别。

[来源: IEEE 982:2024, 3.1]

3. 3496

发布计划 release plan

描述系统功能的哪些部分将在哪些版本中实现，以及每次发布背后依据的计划。

[来源: IEEE 828-2012 系统与软件工程配置管理标准, 2.1]注 1: 该计划包含或引用了对发布内容、发布进度、发布影响及发布通知的说明。

3. 3497

发布规划 release planning

确定产品、交付成果或价值增量发布或过渡的高层次计划的过程。

[来源: PMBOK® 指南 7th Edition]

3. 3498

相关利益相关者 relevant stakeholder

确定参与特定活动并包含在计划中的利益相关者。

[来源: ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表, 3.3836]

3. 3499

可靠性 reliability

系统或组件在规定条件下、指定时间段内，执行其所需功能的能力。

[来源: GB/T 45615-2025, 3.1.27]

3. 3500

可靠性评估 reliability assessment

确定现有系统或系统部件可靠性所达到的水平的过程。

[来源: [GB/T 11457-2006, 2.1335]]

3. 3501

可靠性等级 reliability class

依据以下依据对系统需求与系统行为进行划分子类的类别：1) 安全性、保密性、功能、性能或利用率等典型质量特性；2) 通用可靠性等级（不含能力类型子集）；3) 一个或多个特定于系统的可靠性等级。

注：可靠性等级是确定、监控和评估该等级故障强度的基础依据。

[来源：IEEE 982:2024, 3.1]

3.3502

可靠性数据 reliability data

在软件生存周期中在选择的点上评价软件可靠性所需要的信息。例如可靠性模型中使用的错误数据和时间数据，程序属性（如复杂性），程序设计特性（如使用的开发技术及程序员的经验）。

[来源：[GB/T 11457-2006, 2.1336]]

3.3503

可靠性增长 reliability growth

通过纠正缺陷而实现的可靠性提升。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3839]

3.3504

可靠性模型 reliability model

预测、估计或评估可靠性所使用的模型。

参见：2.1335 条 可靠性评估 (reliability assessment)。

[来源：[GB/T 11457-2006, 2.1339]]

3.3505

可靠性值 reliability numeric

在规定时间间隔内和规定条件下，一配置项将实现所要求功能的概率。

[来源：[GB/T 11457-2006, 2.1340]]

3.3506

可靠性测试 reliability testing

为评价测试项执行其要求功能的能力的一种测试，包括评价在指定条件下运行一段时间中故障发生的频率。

[来源：GB/T 38634.1-2020, 3.33]

3.3507

可重定位的 relocatable

指可以加载到主内存任意位置的代码。

参见：重定位加载器。

注：起始地址由加载器确定，加载器随后调整代码中的地址以反映代码已加载的存储位置。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3841]]

3.3508

可重定位地址 relocatable address

当包含该地址的计算机程序被加载到内存时，由加载器调整的地址（ISO/IEC/IEEE 24765:2017 系统与软件工程——词汇）。

参见：绝对地址。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3842]]

3.3509

可重定位代码 relocatable code

包含在代码加载时由加载器调整以反映代码加载存储位置的地址的代码。

参见：绝对代码。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3843]]

3.3510

可重定位机器[代]码 **relocatable machine code**

要求在计算机执行之前把相对地址翻译为绝对地址的机器语言代码。与 2.14 条 绝对机器代码 (absolute machine code) 相对照。

[来源：[GB/T 11457-2006, 2.1344]]

3.3511

重定位 **relocate**

将机器码从主内存的一个区域移动到另一个区域，并调整地址以使代码可以在新位置执行。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3844]]

3.3512

重定位地址 **relocating address**

当计算机程序包含被装入至内存的地址时，由装入程序调整的地址。与 2.9 条 绝对地址 (absolute address) 相对。

[来源：[GB/T 11457-2006, 2.1342]]

3.3513

重定位汇编程序 **relocating assembler**

生成可重定位代码的汇编器。参见：绝对汇编器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3845]]

3.3514

重定位[代]码 **relocating code**

代码包含由装入程序调整以反映代码装入的存储单元的地址。与 2.11 条 绝对代码 (absolute code) 相对。

[来源：[GB/T 11457-2006, 2.1343]]

3.3515

重定位装入程序 **relocating loader**

相对装入程序 **relative loader**

将可重定位代码读入主内存并调整代码中的地址以反映代码已加载的存储位置的加载器。

参见：绝对加载器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3846]]

3.3516

重定位字典 **relocation dictionary**

目标模块或加载模块中标识重定位发生时被调整地址的部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3847]]

3.3517

重定位因素 **relocation factor**

计算机程序已加载起始地址与汇编起始地址之间的差值；

2. 加到相对地址上以确定要访问的存储位置地址的数值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3081]]

3. 3518

搬迁透明度 relocation transparency

对与其绑定的其他接口屏蔽某一接口重定位的分布透明性。[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：架构，4.4.1.5]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3848]]

3. 3519

重定位器 relocater

管理接口位置存储库（包括支持这些接口的集群管理功能的位置）的对象。[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：架构，14.3.1.1]。

3. 3520

远程作业入口 remote job entry; RJE

远程批量输入。

通过数据链路连接计算机的远程输入设备提交作业。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3850, 有修改]

3. 3521

远程作业入口 remote job entry; RJE

远程批处理入口 remote batch entry

通过数据链路连接到计算机的远程输入设备提交作业。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3850]

3. 3522

远程终端仿真器 remote terminal emulator; RTE

可仿真一组用户的数据处理系统。

[来源：ISO/IEC 14756:1999, 4.15]

3. 3523

修复 repair

缺陷去除 defect removal

为缓解系统故障或失效而采取的纠正性、紧急性或预防性支持措施。

示例：不导致应用程序失效的缺失功能（外部设计错误）或导致停止运行情况的错误（代码错误）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3852]

3. 3524

变基 reparent

将在某一支下开发的变更移入另一个分支。

注：这些变更不会提交到原始分支。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3853]

3. 3525

可重复性 repeatability

见：2.1717 条 测试可重复性 (test repeatability)。

总表有“test repeatability”建议合并。

3. 3526

(测量结果的) 重复性 repeatability (of results of measurement)

在相同测量条件下，对同一待测物理量执行连续测量所产生的结果接近一致的程度。

[来源：GB/T 25000.21-2019, 4.15]

3. 3527

重复寻址 repetitive addressing

隐含寻址方式，其中计算机指令的操作字段被理解为寻址上一条已执行指令的操作数。

参见：前向寻址。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3855]]

3. 3528

可替换性 replaceability

一个产品在相同环境下为相同目的替代另一个指定软件产品的程度。

注 1：软件产品新版本的可替换性对用户升级时很重要。可替换性将降低锁定风险，以便可以使用其他软件产品代替当前软件产品。

同义词：adaptability, installability。

[来源：ISO/IEC 25010:2011, 4.2.8.3]

3. 3529

重演 replay

一种调试技术，记录程序执行历史，然后在用户控制下按正向或反向重放。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3951]]

3. 3530

复制 replication

将软件产品从一种介质复制到另一种介质。[来源：ISO/IEC/IEEE 90003:2018 软件工程——ISO 9001:2015 在计算机软件中的应用指南, 3.13]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3857]]

3. 3531

复制模式 replication schema

对对象复制约束的规范，包括对对象可用性的约束和对对象性能的约束。[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：架构, 16.7.1.1]。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3858]]

3. 3532

复制透明度 replication transparency

屏蔽使用一组相互行为兼容对象来支持某一接口的分布透明性。[来源：ISO/IEC 10746-3:2009 信息技术——开放分布式处理——参考模型：架构, 4.4.1.6]。

注：复制通常用于提高性能和可用性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3859]]

3. 3533

仓库膨胀 repo bloat

记录在仓库中但对项目历史毫无用处的变更。

示例：一个庞大的错误提交及其相关的回退操作。

注：仓库管理员可通过仓库清理来减少仓库膨胀。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3. 3534

仓库清理 repo surgery

绕过版本控制系统的常规命令，直接对版本库进行的修改操作。

示例：通过直接操作数据库命令或直接操纵系统文件来修改版本库。

[来源：ISO/IEC/IEEE 24765:2017, X. X]

[来源：条目注释 1：通过仓库清理，仓库管理员可以执行版本控制系统本身不直接支持的操作，例如永久清除历史记录中的敏感信息或压缩仓库膨胀]

3. 3535

报告 report

以用户指定的布局下的数据输出。

注：输出介质对于 FPA 来说是无关紧要的，记录可能同时涉及到外部输出和外部查询。

[来源：GB/T 42588-2023, 1. 846]

3. 3536

报告绩效 report performance

[过程]收集和分发绩效信息的过程，包括状态报告、进度测量和预测。

[来源：项目管理知识体系指南（PMBOK(R) 指南）——第七版）]

3. 3537

报告标准 report standard

描述工程和管理活动结果的特征的标准。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3. 3863]

3. 3538

储藏库 repository

用于支持数据检索、有组织且持久的数据存储。

注：工件例如软件工程环境。基准存储库是为基准测试目的、指定作为比较度量来源的仓库。在仓库中，工作产品和其他项被保存以供将来需要时检索；而在库中，工作数据仅临时存储并按需检索。

[来源：ISO/IEC/IEEE 26511:2018, 3. 1. 24]

3. 3539

表示 representation

对物理、操作或概念图像或情况的逻辑描述。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 1. 162]

3. 3540

表示属性 representation property

算法运行的属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 1. 163]

3. 3541

表示标准 representation standard

描述工程和管理产品外观特征的标准。

[来源：GB/T 11457-2006, 2. 1358]

3. 3542

再现性 reproducibility

在改变测量条件下，对同一被测量进行多次测量所得结果之间的一致程度。

注：重复性和再现性可借助于各结果的离差特性定量表达出来。

[来源：ISO/IEC 25021:2012, 4. 16 ;

[来源：ISO/IEC TR 14143-3:2003, 3. 9;]

[来源：GB/T 25000. 21-2019, 1. 209]]

3. 3543

(测量结果的) 再现性 reproducibility (of results of measurement)

在不同测量条件下, 对同一待测物理量执行测量所产生的结果接近一致的程度。

[来源: [GB/T 25000.21-2019 (ISO/IEC 25021:2012, IDT)]]

3. 3544

请求 request

从一个对象(发送者)发送到另一个对象(接收者)的消息, 指示接收者履行其职责之一。

参见: 消息。

注: 具体而言, 请求可以是获取属性值、参与者属性值、调用操作, 或验证约束条件是否成立。

[来源: ISO/IEC/IEEE 15289:2019, 5.22]

3. 3545

变更请求 request for change; RFC

对系统、服务、组件或服务管理系统进行更改的建议。

参见: 变更请求、修改请求。

注: 服务变更包括提供新服务或删除不再需要的服务。

[来源: ISO/IEC 16350-2015, 4.30]

3. 3546

招标 request for proposal; RFP

由采购方使用, 用于向潜在投标方宣告其采购特定系统、软件产品或软件服务意向的文档。

[来源: ISO/IEC/IEEE 24765 :2025 FDIS, 3.3873]

3. 3547

需求表 request form

可以多次评估或执行以引起请求发出的描述或模式。

[来源: ISO/IEC 19500-1:2012, 5.3.2]

3. 3548

请求的更改 requested change

正式记录的变更请求, 提交给整体变更控制过程审批的正式书面变更请求。

[来源: ISO/IEC/IEEE 24765:2017, 3.3428]

3. 3549

必需输入 required inputs

为执行任何生存周期活动中所规定的最低限度的验证与确认(V&V)任务所必需的一组项目。

[来源: IEEE 1012-2024, 3.1]

3. 3550

所需输出 required outputs

由于执行任何生存周期活动中规定的最低验证和验证任务而产生的一组项目。

[来源: IEEE 1012-2024 IEEE, 3.1]

3. 3551

需求 requirement

<系统、软件、企业>诠释或表达需要及其限制与条件的陈述。

[来源: ISO/IEC/IEEE 12207:2017, 4.1.31; ISO/IEC/IEEE 15288:2023, 3.36]

<用户>用户为解决某一问题或达到某个目标所需要的条件或能力。

[来源: GB/T 11457-2006, X.X]

〈系统〉系统或系统部件必须满足或处理的条件或能力以满足合同、标准、规格说明或其他正式的强制性文件的要求。

参见：设计需求、功能需求、实现需求、接口需求、性能需求、和物理需求、

[来源：GB/T 11457-2006, X. X]

3. 3552

需求评审 requirement review; requirements review

将系统、硬件项或软件项的要求提交给项目人员、经理、用户、客户或其他相关方征求意见或批准的过程或会议。

参见：代码评审、设计评审、正式资格评审、测试准备评审。

注：类型包括系统需求评审、软件需求评审。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3891]

3. 3553

需求标准 requirement standard

描述需求规格说明的特征的标准。

[来源：GB/T 11457-2006 , 2. 1362]

3. 3554

需求分配 requirements allocation

为完成较低级别的分区功能分配或编制顶层功能或非功能需求的预算。

注：以这种方式，执行全部或部分特定需求的系统元素被识别。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3878]

3. 3555

需求分析 requirements analysis

研究用户需求以得出系统、硬件或软件需求定义的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3879]

3. 3556

需求属性 requirements attributes

与需求相关的一组属性。

[来源：ISO/IEC TR 24766:2009, 3. 7]

3. 3557

需求推导 requirements derivation

通过分析将需求更改或转换为适合低级分析或设计的形式。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 3881]

3. 3558

需求文档 requirements document

包含即用型软件产品（RUSP）要满足的要求或法规的任意组合的文件。

参见：需求文档。

示例：技术或人体工程学标准、来自团体（例如市场部门、技术或用户协会）的需求列表（或模型需求规范）、法律或法令。

[来源：ISO/IEC 25051:2014, 4. 1. 13]

3. 3559

需求文件 requirements documentation

产品需求和其他产品信息的记录，以及用来记录并加以管理的任何信息。

[来源：项目管理知识体系指南（PMBOK® 指南） 第七版]

3. 3560

需求获取 requirements elicitation

使用系统技术，如原型设计和结构化调查，主动识别和记录客户（3. 1. 9）和最终用户（3. 1. 35）的需要。

[来源：GB/T 45802-2025（ISO/IEC/IEEE 29148:2018, IDT）]

3. 3561

需求工程 requirements engineering

在需方（3. 1. 1）和供方（3. 1. 31）之间进行跨学科领域的沟通协调，以明确和维护相关系统、软件或服务待满足的需求（3. 1. 19）。

注：需求工程涉及发现、获取、开发、分析、验证、确认、沟通、记录和管理需求。

[来源：GB/T 45802-2025（ISO/IEC/IEEE 29148:2018, IDT）]

3. 3562

需求逐级分解 requirements flow-down

将系统需求系统性地拆解为分配需求和衍生需求，并合理分配至低层功能组件的过程。

[来源：ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表，3. 3886]

3. 3563

需求管理 requirements management

在系统、产品或服务的整个生存周期中识别、记录、维护、沟通、追溯和跟踪需求（3. 1. 19）的活动。

[来源：GB/T 45802-2025（ISO/IEC/IEEE 29148:2018, IDT）]

3. 3564

需求管理计划 requirements management plan

项目或项目集管理计划的组件，描述将如何分析、记录和管理需求。

[来源：ISO/IEC/IEEE FDIS 24765，3. 3888]

3. 3565

需求划分 requirements partitioning

将顶层需求或设计进行拆分或分解，形成逐级细化的低层级具体需求或设计的过程。

[来源：ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表，3. 3889]

3. 3566

需求阶段 requirements phase

软件生存周期中的一个阶段，在此期间对软件产品的需求（如功能和性能方面的能力）进行定义并编制出相应的文档。

[来源：GB/T 11457-2006，2. 1365]

3. 3567

需求规格说明 requirements specification

指定系统或组件要求的文件。

参见：设计说明、功能规范、性能规范。

注：通常包括功能要求、性能要求、接口要求、设计要求和开发标准。

[来源：ISO/IEC/IEEE FDIS 24765，3. 3892]

3. 3568

需求规格说明语言 requirements specification language

具有特殊结构的规范语言，有时还有验证协议，用于开发、分析和记录硬件或软件需求。

参见：设计语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3893]

3.3569

需求可追溯性 requirements traceability

领域和应用需求各自可追溯性以及它们之间的可追溯性。

[来源：GB/T 45804-2025 (ISO/IEC 26551:2016, IDT)]

3.3570

需求跟踪矩阵 requirements traceability matrix; RTM

将需求与其更高层级的需求或需要，或更低层级的实现联系起来的结构化信息制品。

[来源：GB/T 45802-2025, 1.1034]

3.3571

需求跟踪工具 requirements traceability tool

在逐项软件需求规格说明、设计元素、代码元素和测试用例之间建立可追溯性的软件开发工具。

注：它还支持各种关联查询、分析和报告生成功能。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3896]

3.3572

需求确认 requirements validation

证实需求（单独或作为集合）正确定义了利益相关方预期的系统（3.1.28）。

[来源：GB/T 45802-2025 (ISO/IEC/IEEE 29148:2018, IDT)]

3.3573

需求验证 requirements verification

通过检查证实需求（3.1.19）（单独和作为集合）是构造良好的。

注：这意味着已对一个或一组需求进行了评审，以帮助确保实现良好需求的特征，并对需求集进行了良好的组织。

[来源：GB/T 45802-2025 (ISO/IEC/IEEE 29148:2018, IDT)]

3.3574

需求验证和可追溯性矩阵 Requirements Verification and Traceability Matrix; RVTM

由需求、其验证属性及其追溯关系构成的列表，并包括由于验证活动引起的对系统需求的任何拟议更改。

[来源：INCOSE《系统工程手册》第4版]

3.3575

基于需求的测试 requirements-based testing

一种基于对原子需求进行验证的、以规格说明为依据的测试用例设计技术。

[来源：ISO/IEC/IEEE 29119-4:2021 系统与软件工程——软件测试——第4部分：测试技术, 3.42]

3.3576

储备 reserve

项目管理计划中预先的准备，旨在减轻成本和/或进度风险；前面经常有一个修饰语（例如管理储备、应急储备），以便进一步说明要减轻哪些类型的风险。[来源：项目管理知识体系指南（PMBOK(R) 指南）第七版]。

同义词：应急储备，参见：缓冲。

3.3577

储备分析 reserve analysis

一种用于评估项目风险量以及进度和预算储备的方法，以确定这些储备是否足以应对剩余风险。
[来源：项目管理知识体系指南（PMBOK(R) 指南） 第七版]。

3. 3578

储备 reserve

应急储备 contingency allowance

项目管理计划中用于降低成本和/或进度风险的准备措施。通常与修饰词一起使用(例如:管理储备、应急储备),以进一步说明旨在缓解的风险类型。[项目管理知识体系指南(PMBOK® 指南)——第七版]。

参见:缓冲。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3901]

3. 3579

保留字 reserved word

编程语言中的词，其含义由该语言的规则确定，并且在某些或所有上下文中，程序员只能按其预定用途使用，不能用作其他用途。

例如: IF, THEN, WHILE。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3903]

3. 3580

复位 reset

把变量、寄存器或其他存储单元设置回预先描述的状态。

参见: 清除、初始化。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3. 3904]

3. 3581

复位弧 reset arc

清空输入库所的特殊弧类型。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3905]

3. 3582

常驻控制程序 resident control program

始终保留在主内存中的操作系统部分 2. 封装系统基本功能或功能集的软件模块。参见: 核心程序、监督程序。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 2503]]

3. 3583

残留控制 residual control

微程序技术，其中微指令中字段含义取决于辅助寄存器中的值。

参见: 比特转向，二级编码。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 3906]

3. 3584

残余风险 residual risk

风险处理后仍然存在的风险。

[来源: ISO/IEC 15026-1:2019, 3. 3. 7]

3. 3585

韧性 resilience

系统在面对逆境时提供所需能力的的能力。

[来源: GB/T 45615-2025, 1. 957; ISO/IEC/IEEE 24641:2023, 3. 1. 28]

3. 3586

承受力 resiliency

在工作负载运行期间，被测系统（SUT）在受到干扰的影响下无错误完成交易的能力。

[来源：GB/T 25000.45-2018，？]

3. 3587

抗攻击性 resistance

产品在遭受恶意攻击时维持运行的能力。

[来源：ISO/IEC 25010:2023，3.6.6]

3. 3588

资源 resource

过程执行期间使用或消耗的实体。

示例：包括不同的实体，例如资金、人力、设施、固定设备、工具，和公共设施，例如电力、水力、燃料和通信设施。

[来源：GB/T 45615-2025（ISO/IEC/IEEE 24641:2023，IDT）]

3. 3589

资源分配 resource allocation

将责任分配（划分）到不同的组织单位。

[来源：ISO/IEC/IEEE FDIS 24765，3.3911]

3. 3590

资源分解结构 resource breakdown structure; RBS

按类别和类型对资源进行层级化分类的表示方法。

[来源：PMBOK® 指南 7th Edition]

3. 3591

资源日历 resource calendar

标识各特定资源可用的工作日与轮班时间的日历。

参见：项目日历。

[来源：ISO/IEC/IEEE 24765:2017，3.3464]

3. 3592

资源直方图 resource histogram

显示资源在一系列时间段内计划工作的时间量的条形图。出于比较的目的，资源可用性可以被描述为一条线。对比条可能会显示随着项目进展实际使用的资源量。

[来源：[项目管理知识体系指南（PMBOK® 指南）-第五版]]

3. 3593

资源平衡 resource leveling

一种基于资源约束调整开始和结束日期的技术，目的是平衡资源需求与可用供给。

参见：资源优化技术、资源平滑。

[来源：ISO/IEC/IEEE FDIS 24765，3.3913]

3. 3594

资源管理 resource management

用于开发产品或执行服务的工具的标识、估计、分配和监控。

[来源：GB/T 11457-2006，2.1376]

3. 3595

资源经理 resource manger

对一项或多项资源具有管理权限的个人。

[来源：[项目管理知识体系指南（PMBOK(R) 指南）-第六版]]

3. 3596

资源监控任务 resource monitor task

确保对资源进行串行访问的任务。

[来源：ISO/IEC/IEEE FDIS 24765 系统与软件工程——词汇表，3.3915]

3. 3597

资源池 resource pooling

(1) 可以聚合物理或虚拟资源以向一个或多个服务客户提供服务的特性。

[来源：ISO/IEC TS 25052-1:2022, 3.1.1.2]

3. 3598

资源平滑 resource smoothing

一种调整进度模型活动的技术，使项目对资源的需求不超过某些预定义的资源限制。

[来源：项目管理知识体系指南（PMBOK® 指南）-Fifth Edition]

3. 3599

资源利用率 resource utilization

产品在规定条件下以不超过规定数量的资源执行其功能的能力。[来源：ISO/IEC 25010:2023, 系统与软件工程——系统与软件质量要求和评价（SQuaRE）——产品质量模型，3.2.2]。

参见：效率。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3917]

3. 3600

资源利用率度量 resource utilization measurement; RUM

用于提供与 IT 资产相关资源信息、以便于对其进行管理的结构。

注：RUM 结构专门包含与 IT 资产相关的资源消耗信息。

[来源：ISO/IEC 19770-4:2017, 3.8]

3. 3601

资源利用率测量 resource utilization measurement (RUM); RUM

提供与 IT 资产相关的资源信息以促进其管理的结构。[来源：ISO/IEC 19770-4:2017 信息技术——IT 资产管理——第 4 部分：资源利用率测量，3.8]。

注：RUM 结构特别包含有关与 IT 资产相关的资源消耗的信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3918]

3. 3602

资源有限的时间表 resource-limited schedule

项目进度表，其进度活动、计划开始日期和计划完成日期反映预期的资源可用性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3919]

3. 3603

再专业化 respecialize

将一个实例从其当前子类的实例更改为其当前集群中其他子类之一的实例。

参见：专业化，非专业化。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.165;

[来源: ISO/IEC 10746-2:2009, 13.4.2]]

3.3604

响应对象 *responding object*

参与通信的对象，不是发起对象。

[来源: ISO/IEC 10746-2:2009, 13.4.2]

3.3605

响应时间 *response time; RT*

在至交互的计算机系统的调查或命令的结束与系统的响应的开始之间消逝的时间。

参见：端口到端口的时间、思考时间和往返时间

[来源: ISO/IEC/IEEE FDIS 24765, 3.3922]

3.3606

职责 *responsibility*

就自身行为向某人或某组织作出说明的能力。

注：实例拥有知识，展示行为，遵守规则。这些统称为实例的责任。一个类抽象了它的实例的共同职责。责任可以应用于类的每个实例（实例级）或整个类（类级）。

[来源: IEEE 7005 2021, 3.1]

3.3607

职责分配矩阵 *responsibility assignment matrix (RAM); RAM*

显示分配给每个工作包的项目资源的网格。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3608

责任分配矩阵 *responsibility assignment matrix (RAM).; RAM*

一种展示项目资源在各个工作包中的任务分配的表格。

[来源: 项目管理知识体系指南 (PMBOK® 指南) 第七版]

3.3609

响应式网页设计 *responsive web design; RWD*

一种网页构建方法，用于检测用户的屏幕尺寸和方向，并动态调整布局。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.20]

3.3610

响应性 *responsiveness*

IT 服务及时响应并提供结果的程度。

[来源: ISO/IEC TS 25011:2017, 3.2.6]

3.3611

重启动 *restart*

在发生故障后，利用在检查点记录的状态与结果，使计算机程序恢复执行。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.3927]

3.3612

重启动点 *restart point*

计算机程序中发生故障后可以重新开始执行的点。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.3928]

3.3613

重启点 *restart point*

恢复点 rescue point

计算机程序中在发生故障后可以重新开始执行的点。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3928]

3.3614

结果 result

返回给客户端的信息。

参见:可交付成果、成果、产品。

[来源:项目管理知识体系指南(PMBOK® 指南) — 第七版]

3.3615

保留款 retainage

在合同完成之前暂扣的部分合同款项,用于确保合同条款得到全面履行。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3931]

3.3616

复测 retesting

重新执行测试结果为“不通过”的测试用例,以评价纠正措施的有效性。

[来源:GB/T 38634.1-2020, 1.309]

3.3617

复测 retesting

确认测试 confirmation testing

验证为纠正故障所做的修改是否已成功消除故障而进行的测试。

注:执行复测时,通常辅以回归测试,以帮助确保未修改的其他部分未因修改而受到意外不利影响。同义词:confirmation testing。

[来源:ISO/IEC/IEEE 29119-2:2021 软件与系统工程—软件测试—第2部分:测试过程, 3.15]”

3.3618

退役 retirement

由运维组织撤销主动支持、部分或全部替换为新系统、安装升级系统,或最终退役与处置。

[来源:ISO/IEC/IEEE 15288:2023, 3.38]

3.3619

退役阶段 retirement phase

软件生存周期中终止对软件产品的支持的时间段。

参见:软件生存周期,系统生存周期。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3934]

3.3620

回顾性 retrospective

定期举办的研讨会,参与者通过探讨工作成果来优化流程与产品。

[来源:PMBOK® 指南 7th Edition]

3.3621

回溯日志 retrospective trace

计算机程序执行期间记录的历史数据所生成的跟踪信息。

参见:执行日志、子程序日志、符号日志、变量日志。

注:追溯日志与普通日志不同,普通日志是在程序执行过程中累积生成的。

[来源:ISO/IEC/IEEE FDIS 24765, 3.3937]

3.3622

返回 **return**

〈软件〉将控制权从软件模块转移至调用它的模块。

〈数据〉为参数赋值，使调用模块能够访问该参数。

〈计算机〉执行返回操作的计算机指令或进程。

参见：返回码。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3938]

3.3623

返回码 **return code**

被调用模块返回后用于影响调用模块执行的代码。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3939]

3.3624

投资回报率 **Return on investment; ROI**

产出(产品或服务)收入与开发和生产成本的比率，它决定了组织是否从执行某项生产活动中获益。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3940]

3.3625

投资回报率 (ROI) **return on investment (ROI); ROI**

产出(产品或服务)收入与研发和生产成本的比率，它决定了一个组织是否从某项生产活动中获益。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3941]

3.3626

返回值 **return value**

由被调用模块赋值给参数由调用模块访问。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3941]

3.3627

可重用性，可复用性 **reusability**

产品能够作为资产在多个系统中使用，或用于构建其他资产的能力。

[来源：ISO/IEC 25010:2023, 3.7.2]

3.3628

可重用的，可复用的 **reusable**

指可在多个计算机程序或软件系统中使用的软件模块或其他工作产品。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3943]

3.3629

可重复使用的产品 **reusable product**

为一种用途而开发但具有其他用途的系统、软件或硬件产品，或者专门开发用于多个项目或在一个项目中扮演多个角色的系统、软件或硬件产品。

示例：商用现成 (COTS) 产品、收购方提供的产品、重用库中的产品和预先存在的开发人员产品。

注：每次使用都可以包括产品的全部或部分，并且可以涉及其修改。该术语可应用于任何软件或系统产品（例如，需求或体系结构），而不仅仅是软件或系统本身。

[来源：IEEE 1012-2024 IEEE, 3.1]

3.3630

可复用软件产品 **reusable software product**

为某一用途开发但可用于其他用途的软件产品，或专门开发以在多个项目或同一项目的多个角色

中可用的软件产品。(IEEE 1012-2024 系统、软件和硬件验证与确认 IEEE 标准, 3.1.25)。

注：每种用途可能包含软件产品的全部或部分，并可能涉及对其进行修改。本术语可适用于任何软件产品（例如：需求、架构），而不仅限于软件本身。

[来源：[IEEE_CS_SE_Vocabulary-2025]]

3.3631

重用 reuse

复用

至少部分地从现有部分构建软件系统以执行新的应用程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3945]

3.3632

复用来源声明 reused source statement

从外部来源获得的产品的未修改来源声明。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3946]

3.3633

显现需求 revealed requirement

系统产生无任何显式需求对其予以禁止或允许的异常或行为时，由此推导得出的需求。

注：异常或行为可被判定为可接受或有害，从而产生允许或禁止该行为的揭示性需求。揭示性需求被视为与明确需求具有同等法律效力。

[来源：IEEE 982:2024, 3.1]

3.3634

收入备选方案 revenue alternative

以其完整的现金流量流来描述的备选方案，包括费用和收入。

参见：替代服务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3948]

3.3635

逆向工程 reverse engineering

从代码中推导系统设计或需求的软件工程方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3949]

3.3636

逆向执行 reverse execution

一种调试技术，记录程序执行历史，然后在用户控制下按正向或反向重放。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3951]

3.3637

可撤销性 reversibility

云服务为客户提供取回其数据与应用制品的流程，同时为服务商提供在约定期限届满后删除全部客户数据，以及按合同约定删除由云服务衍生数据的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.6.2]

3.3638

可逆执行 reversible execution

一种调试技术：记录程序执行历史，然后在用户控制下正向或反向重放。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.3951]

[来源：同义词：反向执行，回放、回放、逆向执行。]

3.3639

回放调试 reversible execution; playback; replay; reverse execution

逆向调试 backward execution

记录程序执行历史然后在用户控制下向前或向后重放的调试技术。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3951]

3.3640

评审 review

流程或会议,在此期间,工作产品或一组工作产品,提交给项目人员、经理、用户、客户或其他相关方以征求意见或批准。

[来源: ISO/IEC 29110-1-2:2024, 3.79]

3.3641

评审集 review folder

用于绑定一个或多个相关评审的实体,包括评审列表以及各评审共用的信息。

注:评审共用信息可包括可参与或组织评审的人员信息,以及对评审过程中识别出的问题所赋予的分类信息。

[来源: ISO/IEC 23396:2020, 3.2]

3.3642

评审/审核输出 review/audit output

通过开展技术评审或审核预期得到的评审或审核产物,且可作为推出准则组成要素的成果。

[来源: ISO/IEC/IEEE 24748-8:2019, 3.1]

3.3643

审查/审核输出 review/audit output

审核输出 audit output

审查输出 review output

审查或审计通过技术审查或审计预期的工件,并且可以被视为退出标准的要素。

[来源: ISO/IEC/IEEE 24748-7:2019, 3.1; ISO/IEC/IEEE 24748-8:2019, 3.1]

3.3644

评审输出/审计输出 review/audit output; audit output; review output

通过开展技术评审或审计所预期形成的评审或审计工作产物,且可作为退出准则的组成要素。

[来源: ISO/IEC/IEEE 24748-8:2019, 3.1]

3.3645

撤销 revocation

撤销授权或授权模式的过程。

[来源: ISO/IEC 19770-3:2016, 3.1.21]

3.3646

奖励黑客行为 reward hacking

(1) 代理人为最大化其奖励函数而损害原始目标达成的行为。

[来源: ISO/IEC TR 29119-11:2020, 6.1.65]

3.3647

返工 rework

为使有缺陷或不合格的组件符合要求或规范而采取的行动。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3648

变更请求 RFC; request for change; RFC

对现有应用进行功能性或非功能性变更的提案。

参见：变更请求、修改请求。

注：对服务的变更包括提供新服务或移除不再需要的服务。

[来源：ISO/IEC 16350-2015, 4.30]

3.3649

权利 right

由软件授权所授予的特权或利益。

[来源：ISO/IEC 19770-3:2016, 3.1.22]

3.3650

行波时钟 ripple clock

当前一级时序逻辑的数据输出被用作下一级时序逻辑的时钟输入时的时钟信号。

[来源：GB/T 37691-2019, X.X]

3.3651

风险接受 risk acceptance

一种风险应对策略：项目团队决定接受风险，除非风险出现，否则不采取任何行动。

[来源：项目管理知识体系指南（PMBOK®指南）第七版]

3.3652

风险行动请求 risk action request

针对确定为高于风险阈值的一种或多种风险推荐的处理方案和支持信息。

[来源：ISO/IEC 16085:2006, 3.7]

3.3653

风险分析 risk analysis

检查已识别风险因素的发生概率、潜在损失和潜在的风险应对策略的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3967]

3.3654

风险偏好 risk appetite

一个实体在预期获得回报时愿意接受的不确定性程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3968]

3.3655

风险审核 risk audit

对已识别风险及其根本原因的风险应对措施有效性进行审查与记录，同时评估风险管理流程的有效性。

[来源：ISO/IEC/IEEE 24765h:2019]

3.3656

风险规避 risk avoidance

排除风险因素的行动。

[来源：ISO/IEC/IEEE 24765:2017]。

3.3657

风险分解结构 risk breakdown structure; RBS

潜在风险来源的层级化呈现。

[来源：PMBOK® 指南 7th Edition]

3.3658

风险类别 risk category

风险的类别或类型。

注：风险类别是对风险源的描述。

[来源：ISO/IEC/IEEE 16085:2021，3.8]

3.3659

风险标准 risk criteria

评估风险重要性的职权范围。

[来源：ISO/IEC/IEEE 24765:2017，3.3521]

3.3660

风险增强 risk enhancement

一种风险应对策略，项目团队采取行动以增加机会发生的可能性或影响。

[来源：ISO/IEC/IEEE FDIS 24765，3.3973]

3.3661

风险升级 risk escalation

一种风险应对策略，团队承认某一风险超出了其影响范围，并将该风险的责任转移到组织的更高层，从而更有效地进行管理。

[来源：ISO/IEC/IEEE FDIS 24765，3.3974]

3.3662

风险利用 risk exploiting

项目团队采取行动以帮助确保某一机会发生的风险应对策略。

[来源：ISO/IEC/IEEE FDIS 24765，3.3975]

3.3663

风险暴露 risk exposure

风险对个人、项目或组织造成的潜在损失。 [来源：ISO/IEC/IEEE 16085:2021，3.10]

注：风险暴露通常定义为概率与后果严重程度的乘积，即期望值或预期暴露。

3.3664

风险因素 risk factor

指可能对计划中的活动、项目或计划造成不利影响的潜在问题，其特征为问题发生的概率（ $0 \leq p \leq 1$ ）以及问题发生时可能造成的损失（如生命、金钱、财产、声誉等）。

注：概率与潜在损失均可能随时间变化。

[来源：ISO/IEC/IEEE FDIS 24765，3.3977]

3.3665

风险应对措施 risk handling

针对风险因素采取的应对措施。

注：包括风险接受、风险规避、风险转移和风险缓解。

[来源：ISO/IEC/IEEE FDIS 24765，3.3978]

3.3666

风险识别 risk identification

采用系统化方法确定计划活动、项目或方案相关风险因素。。。

[来源：ISO/IEC 25010:2023,, 3.9.2]

3.3667

风险杠杆因子 risk leverage factor; rlf

$RLF = (reb\_rea)/rmc$ ，其中 reb 表示风险缓解前的暴露水平，rea 表示风险缓解后的暴露水平，rmc 为风险缓解活动成本。

注：RLF 值越大，表明风险缓解策略越有效。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3980]

3.3668

风险管理 risk management; RM

用于识别和处理风险因素的系统化流程。

注：风险管理的核心目标是通过提前识别潜在问题并及时应对，从而避免危机发生。该流程涵盖风险因素的初始识别与处理，以及持续的风险管理。

[来源：ISO /IEC/IEEE FDIS 24765, X.X]

3.3669

风险管理计划 risk management plan

项目、计划或组合管理计划中描述风险管理活动结构与实施方式的组成部分。

注：另见风险登记册。

[来源：GB/T 11457-2006, 2.1395]

3.3670

风险管理流程 risk management process

在产品或服务的整个生存周期中系统地识别、分析、处理和监控风险的连续过程。

[来源：ISO/IEC 16085:2006, 3.12]

3.3671

风险缓解措施 risk metric

与要减轻的风险因素相关的客观措施。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3983]

3.3672

风险缓解 risk mitigation

项目团队采取行动以降低风险发生概率或影响的风险应对策略。

注：包括在风险指标越过预定阈值时（当风险变为问题或导致故障时）执行应急计划。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3984]

3.3673

风险监控 risk monitoring and control

跟踪已识别风险、监控剩余风险、识别新风险、执行风险应对计划，并在项目全生存周期内评估其有效性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3985]

3.3674

风险责任人 risk owner

明确具有风险管理问责与权限的人员。

[来源：ISO 31000:2018 5.4.3]

3.3675

风险概况 risk profile

风险集合的描述。

[来源: ISO/IEC/IEEE 16085:2021 , 3.8]

3.3676

风险再评估 risk reassessment

识别新风险、重新评估现有风险以及关闭已过时的风险。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3987]

3.3677

降低风险 risk reduction

降低风险因素的可能性和/或潜在影响。

注: 风险降低可能涉及研究、原型制作和其他探索方法。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3988]

3.3678

风险降低措施 risk reduction measure

为降低或缓解风险所采取的措施。

[来源: ISO/IEC/IEEE 15026-1:2019 , 3.3.9]

3.3679

风险跟踪表 risk register

记录风险管理流程输出的存储库。

注: 另见风险管理计划另见风险管理计划。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3990]

3.3680

风险报告 risk report

汇总单个项目风险信息和项目整体风险等级的项目文档。

[来源: 《项目管理知识体系指南 (PMBOK® 指南)》第七版]

3.3681

风险审查 risk review

分析现有风险状态并识别新风险的过程。

[来源: 《项目管理知识体系指南 (PMBOK® 指南)》第七版]

3.3682

风险分担 risk sharing

风险应对策略, 项目团队将机会的所有权分配给最能从中获益的第三方。(项目管理知识体系指南 (PMBOK(R) 指南——第七版)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3993]

3.3683

风险源 risk source

单独或组合存在时具有内在风险产生潜力的要素。

注: ISO 指南 73:2009 中将危害定义为风险源实例。在可靠性背景下, 故障、错误或失效均可构成风险源。在安全背景下, 威胁、威胁源或不利行为均可构成风险源。

[来源: ISO/IEC/IEEE 15026-1:2019 , 3.3.10]

3.3684

风险阈值 risk threshold

用于消除风险或将风险降低至可接受水平的流程。

[来源: ISO/IEC/IEEE 15026-1:2019 , 3.3.11]

3.3685

风险承受能力 risk tolerance

组织或个人能够承受的风险程度、规模或数量。

[来源: ISO/IEC/IEEE 16085:2021, 3.10]

3.3686

风险转嫁 risk transfer

将风险因素的管理责任转移至另一组织或职能实体,以使其更具备缓解该风险因素的能力。[来源: ISO/IEC/IEEE FDIS 24765, 3.3997]。

3.3687

风险转移[策略] risk transference

项目团队将威胁造成的影响以及应对责任一并转移给第三方的风险应对策略。

[来源: ISO/IEC/IEEE FDIS 24765, 3.3998]

3.3688

风险处理 risk treatment

消除风险或将风险降低至可接受水平的过程。[来源: ISO/IEC/IEEE 15026-1:2019, 3.3.11]。

3.3689

风险处置措施 risk treatment measure

保护措施。

消除危害或降低风险的行动或手段。

[来源: ISO/IEC 25059:2023, 3.1.4]

3.3690

风险触发 risk trigger

风险指标的预设阈值, 当该指标超过阈值时将触发应急计划的启动。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4001]

3.3691

风险紧迫性评估 risk urgency assessment

对需优先于其他风险项采取措施的时机进行评审与判定。

[来源: ISO/IEC/IEEE 24765:2017, 3.3548]

3.3692

风险调整后的待办清单 risk-adjusted backlog

包含产品工作以及应对威胁与机遇相关行动的待办清单。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3693

基于风险的测试 risk-based testing

基于风险分析确定的风险类型和级别, 有意识地管理、选择、排序和利用测试活动及资源的测试。

[来源: GB/T 38634.1-2020, X.X]。

3.3694

无风险最低期望收益率 risk-free MARR

未针对备选方案风险进行调整的最低期望收益率 (MARR), 因为在决策分析中认为该风险可忽略不计。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4004]

3.3695

路线图 roadmap

指导朝着目标前进的详细计划。

[来源: ISO/IEC/IEEE 26511:2018, 3.1.25]

3.3696

机器人 robot

具有一定自主性、在其环境中移动以执行预期任务的编程驱动机制。

注: 机器人包括控制系统和控制系统的接口。机器人可根据其预期用途分为工业机器人或服务机器人。

[来源: ISO/IEC TR 29119-11:2020, 3.1.66]

3.3697

健壮性, 稳健性 robustness

在存在无效输入或急迫的环境条件下, 系统或部件其功能正确的程度。

[来源: ISO/IEC 25059:2023, 3.2.5]

3.3698

角色 role

实体参与关系。

[来源: ISO/IEC 15474-1:2002, 4.2]

由项目团队成员执行的已定义功能, 例如测试, 归档, 检查, 编码。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

对一个对象在关系中所扮演角色的表述。

[来源: ISO/IEC 15476-4:2005, 6.5]

复合对象规范中的正式占位符, 用于标识某个组件对象为成为组合对象一部分所需的行为特征, 并将这些特征作为约束关联到组合实例中的实际对象。

[来源: ISO/IEC 10746-2:2009, 9.17]

3.3699

角色名称 role name

更具体地命名相关值类或状态类的性质的名称。

注: 对于关系而言, 角色名称是赋予关系中某个类的名称, 以阐明该类在关系中的参与方式, 即暗示相关实例所扮演的角色。对于属性而言, 角色名称是用于在其所属类的上下文中, 明确值类含义的名称。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.167]

3.3700

基于角色的评审 role-based reviewing

评审人员从不同利益相关者角色视角对工作成果进行评审的技术。

注: 典型的利益相关者角色包括特定的用户类型, 例如工作产品维护者、测试者和开发者。

[来源: ISO/IEC 20246:2017, 3.16]

3.3701

角色扮演会议 role-playing session

参与式技术, 通过交互式仿真, 让利益相关方参与到系统设计中涉及的动态交互过程模拟。

[来源: IEEE 2675-2021, 3.1]

3.3702

转入 roll in

将数据或计算机程序段从辅助存储器传输到主存储器。参见: 转出, 交换。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4019,]

3.3703

转出 roll out

将数据或计算机程序段从主存储器转存到辅助存储器,以腾出主存储器用于其他用途。参见: 转入, 交换。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4020]

3.3704

滚动式规划 rolling wave planning

一种迭代式规划技术, 对近期需要完成的工作进行详细规划, 对远期工作仅做高阶规划。

[来源: 《项目管理知识体系指南 (PMBOK® 指南)》第七版]

3.3705

根源 root cause

缺陷的根源, 如果它被移除, 缺陷就会减少或被移除。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4024]

3.3706

根本原因分析 root cause analysis

确定风险因素或问题的根本原因。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4025]

3.3707

根编译程序 root compiler

编译器, 与机器无关的、程序的中间级表示形式。

注: 根编译器, 当与代码生成器结合时, 即可构成一个完整的编译器。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4026]

3.3708

例程 routine

由其他程序和子程序调用的子程序。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4028]

可为单一目的而调用的函数或过程。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4028]

具有通用性或被频繁使用的程序或程序部分。

参见: 协同程序、子程序。

注: 例程、子程序、子例程这些术语在不同编程语言中的定义和用法各不相同。

[来源: ISO/IEC 2382:2015, X.X]

3.3709

规则 rule

对系统规约的约束要求。

注: 当所有条件均满足该规则的条件项时, 该规则即为满足。

[来源: ISO/IEC 10746-2:2009, 11.2.7]

3.3710

基于规则的语言 rule-based language

一种非过程语言, 它允许用户说明一组规则并表达使用这些规则的疑问和问题。

参见: 声明语言 (declarative language)。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4039]

3.3711

运行 run

软件工程中, 指计算机程序单次、通常连续的执行过程。

参见: 运行时 (run time)。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4042]

3.3712

运行态 run mode

运行方式

当计算机正自动地执行着存储在存储单元中的指令, 从而被认为是正在履行其功能时的那种状态。

[来源: GB/T 11457-2006 2.1405]

3.3713

运行流 run stream

程序或作业设置序列, 它使计算机能逐个自动处理而不需操作员的介入。

[来源: ISO/IEC/IEEE FDIS 24765 3.2491]]

3.3714

运行时 run time

运行时间 running time

计算机程序开始执行的时间段。

参见: 执行时间。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4043]

3.3715

运行时环境 runtime environment

面向特定运行环境部署的软硬件配置的具体实例, 所关注的已部署系统在此实例上运行。

注: 运行时环境包括可执行应用软件、数据存储、实用程序、本地部署服务、操作系统、网络接口以及所关注系统 (SOI) 运行所需的物理计算机系统, 但不包含所关注系统本身。

[来源: IEEE 982:2024, 3.1]

3.3716

运行环境 runtime platform

实现应用软件所用服务的硬件和软件组件集合。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4045]

3.3717

参考用户需求 RUR; Reference User Requirements; RUR

符合相关要求的标准用户需求集。

[来源: ISO/IEC TR 14143-4:2002, 4]

3.3718

安全集成 safe integration

产品在与一个或多个组件集成过程中及集成后, 能够维持安全性的能力。

[来源: ISO/IEC 25010:2023, 3.9.5]

3.3719

安全 safety

避免因使用或误用系统而造成损伤或伤害的能力。

[来源：GB/T 45615-2025, X. X]

3.3720

安全标准 safety criteria

危害相关可接受风险的限值。

注：这些限值可定义为强制性安全目标，或通过分析或开发政策制定。

[来源：ISO/IEC TS 15504-10:2011，3.4]

3.3721

安全关键功能 safety critical function

针对特定的危险事件，为达到或保持控制设备的安全状态而实现的功能。

[来源：[GB/T 37691-2019 可编程逻辑器件软件安全性设计指南]]

3.3722

安全示范 safety demonstration

证明某项产品在允许风险范围内具有安全性的证据体系及理论依据。

[来源：ISO/IEC TS 15504-10:2011，3.3]

3.3723

安全注入信号 safety injection signal; SIS

安全注入信号。。

[来源：IEEE 1012-2024n, 3.2]

3.3724

安全完整性要求 safety integrity requirement

安全相关系统在规定条件下能够良好执行所需安全功能的可能性。

[来源：ISO/IEC TS 15504-10:2011，3.6]

3.3725

安全生存周期 safety life cycle

实施安全流程的项目或产品生存周期。

[来源：ISO/IEC TS 33064:2025]

3.3726

安全说明 safety note

在有意义的组织体系中的文档或文档章节中收集或归类得到的安全相关信息，用于解释安全措施、提高安全意识并为人员进行安全相关培训提供基础。

[来源：IEC/IEEE 82079-1:2019，3.33]

3.3727

安全要求 safety requirement

为确保产品安全所需的要求。

[来源：ISO/IEC TS 15504-10:2011，3.8]

3.3728

安全标志 safety sign

给出一般安全信息的标志，通过颜色和几何形状的组合获得，并通过添加图形符号给出特定的安全信息。

[来源：IEC/IEEE 82079-1:2019, 3.34]

3.3729

安全验证 safety validation

通过检验与测试，确认安全目标充分且已实现的保障性工作。

[来源：ISO/IEC 23643:2020《软件和系统工程-软件安全与保密验证工具的能力，3.32]

3.3730

安全关键软件 safety-critical software

若软件故障或功能异常可能导致人员伤亡、设备或财产损失/严重损坏，或对自然环境造成损害的软件。

[来源：ISO/IEC 29110-5-1-2:2025，3.27]

3.3731

安全关键系统 safety-critical system; SCS

若失效或故障会导致以下一种或多种后果的系统：人员死亡或严重伤害、设备或财产损失或严重损坏，或对环境造成危害。

注：针对安全关键系统的软件工程强调：。

[来源：ISO/IEC 23643:2020，软件和系统工程-软件安全和安全验证工具的能力，3.15]

- (1) 过程工程与管理；
- (2) 为系统选择合适的工具与环境；
- (3) 遵循需求。

同义词：生命关键系统。

3.3732

SAM 项目范围/软件资产管理项目范围 SAM program scope

明确列出软件资产管理项目所覆盖的组织所有部门、软件类型、资产及平台的说明清单。

[来源：ISO/IEC 19770-5:2015，3.32]

3.3733

软件资产管理工具 SAM tool

用于协助和自动化部分软件资产管理过程的软件。

[来源：ISO/IEC 19770-3:2016，3.1.25]

[来源：别名：软件资产管理工具]

3.3734

SAM 工具 SAM tool

软件资产管理工具 software asset management tool

用于协助和自动化部分软件资产管理流程的软件。

[来源：[ISO/IEC 19770-3:2016，信息技术—IT 资产管理—第 3 部分：权利模式，3.1.25]]

3.3735

SAM 管理工具/软件资产管理工具 SAM tool; software asset management tool

用于辅助并自动化软件资产管理流程部分环节的软件。

[来源：ISO/IEC 19770-3:2016，3.1.25]

[来源：同义词：软件资产管理工具]

3.3736

示例实例图 sample instance diagram

一种呈现示例实例的形式，其中将实例显示为独立的图形对象。

参见：示例实例表。

注：当仅展示少量实例时，采用图形化方式呈现实例会比较直观。

[来源：ISO/IEC/IEEE FDIS 24765，3.1.169]

3.3737

示例实例表 `sample instance table`

一种展示示例实例的形式，将实例以表格形式呈现。

参见：示例实例图。

注：当需要展示同一个类的多个实例时，采用表格形式呈现实例会比较实用。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.170]

3.3738

满意决策 `satisficing`

用户因使用系统、产品或服务而产生的生理、认知及情感反应在多大程度上满足其需求与期望。

[来源：ISO TR 25060:2023, 3.1.9]。

3.3739

脚手架 `scaffolding`

为支持软件开发与测试而构建的计算机程序和数据文件，不纳入最终产品。

[来源：ISO/IEC/IEEE 24748-6:2023, 3.1.7]

3.3740

可扩展性 `scalability`

产品处理增长或缩减的工作负载，或调整其容量以应对变化的能力。

[来源：ISO/IEC 25010:2023, 3.8.2]

3.3741

扩展性测试 `scalability testing`

性能效率测试的一种。用于评估系统与软件适应外部性能需求变化（如用户负载支持、事务数量、数据量等）的性能表现。

[来源：GB/T 39788-2021, 3.4]

3.3742

原子值 `scalar`

原子值。

参见：集合值。

注：即不可再分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.171]

3.3743

标量类 `scalar-valued class`

每个实例都是单个值的类。

参见：收藏价值类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.173]

3.3744

标量值属性 `scalar-valued property`

映射到标量值类的属性。

参见：集合值属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.174]

[来源：同义词：标量属性]

3.3745

标量值属性 `scalar-valued property`

标量属性 scalar property

一种映射到标量值类的属性 参见：集合值属性。

[来源：ISO/IEC/IEEE FDIS 24765 3.4086]]

3.3746

标度 scale

一组有序连续或离散值，或一组与属性映射的类目。

示例：标度的类型有：标称标度——对应于一组类目；顺序标度——对应于一组有序的标尺点；间隔标度——对应于有等距标尺点的有序标度；比率标度——不仅具有等距标尺点而且拥有绝对零点。使用标称标度或顺序标度的测量生成定性数据，而使用间隔标度和比率标度的测量生成定量数据。

注：改写 1993 年版计量学基本与通用术语国际词汇。

[来源：ISO/IEC 15939:2007；GB/T 25000.1-2021, 1.573]

3.3747

可扩缩性 scalability

是指软件系统可以在不同规模、不同档次的硬件平台上运行的能力。

[来源：[GB/T 11457-2006, 2.1411]]]

3.3748

散点图 scatter diagram

一种双轴相关图表，用于绘制自变量与因变量的多组观测值，并生成回归线以刻画变量间的关系。。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4088]

3.3749

场景 scenario

对并发或顺序发生的一系列事件进行分步描述。

参见：用例。

注 1：场景可以是用户故事、用例、操作概念或软件可能会遇到的事件序列。

同义词：脚本。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4089]

3.3750

场景脚本 scenario script

对同时或依次发生的一系列事件的逐步描述 cf. 用例 **注 1：**场景可以是用户故事、用例、操作概念或软件可能遇到的事件序列。

[来源：ISO/IEC/IEEE FDIS 24765 3.4089]

3.3751

场景测试 scenario testing

一种基于规格说明的测试用例设计技术，基于执行测试项与其他系统之间的交互序列来设计。

注：在此语境下，用户被视为其他系统。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.43]

3.3752

基于场景的评审 scenario-based reviewing

基于场景的评审法：一种通过判定工作产品能否满足特定场景需求来引导评审过程的技术。

[来源：ISO/IEC 20246:2017《软件和系统工程-工作产品评审》 3.17]

3.3753

自变量进度 schedule as independent variable; SAIV

一种确定日期的项目进度安排方法,该项目具有指定完成日期,超过此日期后项目价值将急剧下降,或会对未按期完工处以违约金。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4093]

3.3754

进度基线 schedule baseline

可采用正式变更控制流程修改的进度模型批准版本,作为与实际结果对比的基准。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4094]

3.3755

进度压缩 schedule compression

在不缩小项目范围的前提下,用于缩短进度工期的方法。

参见: 赶工、快速跟进。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3756

进度数据 schedule data

用于描述和控制进度的信息集合。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3757

进度制定 schedule development

通过分析活动顺序、活动工期、资源需求及进度约束,制定项目进度计划的过程。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4097]

3.3758

进度预测 schedule forecast

基于计算进度时可获取的信息和知识,对项目未来条件和事件进行的估计或预测。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4098]

3.3759

进度管理计划 schedule management plan

项目管理计划的组成部分,用于制定开发、监控和控制进度的标准及活动。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4099]

3.3760

进度模型 schedule model

项目活动执行计划的呈现,包括持续时间、依赖关系及其他规划信息,用于生成项目进度表及其他进度管理文件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4100]

3.3761

进度网络分析 schedule network analysis

一种为项目进度活动中未完成部分确定最早开始时间、最晚开始时间以及最早完成时间、最晚完成时间的技术。

[来源: 项目管理知识体系指南 (PMBOK®指南) 第六版]

3.3762

进度绩效指数 schedule performance index; SPI

进度效率度量,以挣值与计划价值之比表示。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.3763

进度绩效指标 `schedule performance index (SPI)`; SPI

进度效率的衡量指标，以挣得价值与计划价值的比率表示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4101]

3.3764

进度偏差 `schedule variance`; SV

进度绩效衡量指标，表示为实际进度值与计划进度值之间的差值。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4633]

3.3765

进度绩效度量 `schedule variance (SV)`

进度绩效度量，以挣得价值与计划价值的差额表示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4102]

3.3766

计划完成日期(SF) `scheduled finish date`; SF

某项进度活动计划完成的时间点。计划完成日期通常介于最早完成日期与最迟完成日期所限定的日期区间内。其可能反映稀缺资源的资源平衡。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3767

计划开始日期(SS) `scheduled start date`; SS

某项进度活动计划开始的时间点。计划开始日期通常介于最早开始日期与最迟开始日期所限定的日期区间内。其可能反映稀缺资源的资源平衡。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3768

调度程序 `scheduler`

一种计算机程序，它通常作为操作系统的一部分，用以调度、启动和终止作业。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4103]

3.3769

图式 `schema`

在 MOF 语境中，一种与作为模型元素的容器类同的包。

[来源：GB/T 11457-2006, 2.1414]

3.3770

模式规则语言 `Schematron`

用于断言 XML 文档中模式存在或不存在的语言。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.27]

3.3771

软件配置管理 `SCM`

软件配置管理。

[来源：ISO/IEC/IEEE 24765i:2020]

3.3772

软件配置管理计划信息 `software configuration management planned information`; SCMPI

记录软件配置管理计划的信息，包括配置项识别、配置控制流程、配置状态记录、配置审核与评审。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3773

软件配置管理计划 software configuration management plan; SCMP

项目管理计划的组成部分，用于定义如何开展配置管理工作。它明确哪些事项属于可配置项、哪些事项需要正式变更控制，以及管控这些事项变更的流程。

[来源：ISO/IEC/IEEE 24765d:2015, X.X]。

3.3774

规格说明变更通知 specification change notice; SCN

配置管理中用于提请、传递并记录规格变更事项的文档。

参见：配置控制、工程变更、修订通知。

[来源：ISO/IEC/IEEE 24765:2017, X.X]。

3.3775

范围基线 scope baseline

范围说明书、工作分解结构（WBS）及其配套的工作分解结构词典的核准版本，仅可通过正式的变更控制程序修改，作为项目范围对比的基准依据。

[来源：项目管理知识体系指南®（PMBOK® 指南）—第七版]

3.3776

范围蔓延 scope creep

没有在原始需求中被指定，但是在已定义的功能和范围中被衍生确定的附加功能。

[来源：GB/T 42566-2023, 1.786]

3.3777

范围管理计划 scope management plan

项目或项目集管理计划的组成部分，描述项目范围将如何被定义、制定、监控、控制和验证。

[来源：项目管理知识体系指南®（PMBOK® 指南）第七版]

3.3778

功能规模度量范围 scope of the FSM

特定功能规模测量实例中所包含的功能用户需求集合。

示例 1：若组织需要了解其软件组合规模，则功能规模测量范围包含当前使用的全部功能用户需求；

示例 2：若项目经理仅需确定某一软件版本规模，则范围仅包含该项目所影响的功能用户需求。

注：功能规模测量范围由软件测量目的决定。

[来源：ISO/IEC 19761:2011, 2.24]

3.3779

屏幕图像 screen capture

软件使用过程中用户界面的可视化呈现。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4116]

3.3780

屏幕截图 screen capture

屏幕转储 screen dump

表示用户在使用软件时将看到的内容。

[来源：ISO/IEC/IEEE FDIS 24765 3.4116]

3.3781

脚本测试 scripted testing

测试者的动作由测试用例中的书面指令来规定的动态测试。

注：该术语通常适用于人工执行的测试，而不是自动化脚本的执行。

[来源：GB/T 38634.1-2020, 1.312]

3.3782

供应链风险管理 **SCRM**

供应链风险管理。

[来源：ISO/IEC/IEEE 24748-8:2019]

3.3783

敏捷开发框架 **scrum**

敏捷开发中采用的迭代项目管理框架，团队通过需求待办列表达成开发项目共识，并在短短数周内完成交付。

[来源：ISO/IEC 29110-1-2:2024 3.81]

3.3784

Scrum 主管 **scrum lead**

负责在团队或项目中推进 Scrum 流程的人。

[来源：ISO/IEC 29110-1-2:2024, 3.82]

3.3785

scrum 报告 **scrum report**

记录 Scrum 团队日常活动的文档，需记录所有待处理的问题或事项。

[来源：ISO/IEC/IEEE 24765h:2019]

3.3786

scrum 团队 **scrum team**

采用 Scrum 流程协作的敏捷开发团队成员，通常由 Scrum 负责人和项目负责人共同领导。

[来源：ISO/IEC/IEEE 26515: 2011, 4.13].

3.3787

s 曲线 **S-curve**

显示特定时间段内累计成本的图表。

[来源：项目管理知识体系指南 (PMBOK(R) Guide) — 第七版]

3.3788

系统开发生存周期 **SDLC; systems development life cycle; SDLC**

与系统开发相关的活动，包括系统的需求、设计、验证、部署、维护和生存周期结束。

[来源：IEEE 7002:2022, 3.1]

3.3789

系统设计评审 **system design review; SDR**

一种评审活动，用于评估系统需求向各配置项分配的方式、形成该分配方案的系统工程过程、下一阶段工作的工程规划、制造相关考量以及生产工程规划。

参见：关键设计评审、初步设计评审。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4675]

3.3790

搜索算法 **search algorithm**

系统地访问所有可能状态（或结构）的子集直到达到目标状态（或结构）的算法。

[来源：ISO/IEC TR 29119-11:2020, 3.1.68]

3.3791

第二范式 second normal form

数据规范化过程的结果，通过数据组转换，使每个非键属性都依赖于数据组的键属性和键属性的所有部分。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4134]

3.3792

二级数据 secondary data

已经从各种来源收集的数据，例如政府人口普查或统计局、非营利组织、学术机构和咨询机构。

[来源：IEEE 7010-2020, 2.1]

3.3793

辅用户 secondary user

与产品交互以支持主用户的用户。

[来源：ISO/IEC 25030:2019, 3.17]

3.3794

辅窗口 secondary window

包含依赖于另一个窗口（主窗口）中信息的信息的窗口。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4137]

3.3795

安全哈希算法(SHA) secure hash algorithm; SHA

通过从数据输入（可以是任意长度的消息）创建消息摘要来验证数据完整性的算法。

注：包括 SHA-1（160 位摘要）和 SHA-2（224 至 512 位摘要）。

[来源：ISO/IEC 19770-5:2015, 3.33]。

3.3796

安全处理需求 secure processing requirement; SPR

安全处理需求。

[来源：IEEE 7005-2021, 3.2]

3.3797

信息安全性 security

保护计算机硬件或软件免受意外或恶意访问、使用、修改、破坏、或披露。

[来源：IEEE 1012-2024 IEEE, 3.1, 有修改]

3.3798

安全认证 security accreditation

管理层出具的正式声明，信息技术系统获准以特定安全模式，使用规定的安全防护措施，在可接受的风险水平下运行。

独立认证机构出具的证明，信息技术系统符合预规的安全标准。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4140]

3.3799

安全与知识产权方案 security and intellectual property scheme

在实体内建立和运行的管理系统，以确保其信息项的安全性和知识产权。

[来源：ISO/IEC TR 29110-5-1-3:2017, 3.5]

3.3800

安全权限 security authority

负责实施安全策略的管理员。

[来源：ISO/IEC 10746-3:2009, 15.1.2]

3.3801

安全分支 security branch

分支，创建于发布时，仅对安全相关提交进行修改。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3802

安全控制 security control

为信息系统或组织制定的防护措施与应对措施，旨在保护其信息的保密性、完整性和可用性，并满足一系列既定的安全要求。

[来源：IEEE 7002:2022, 3.1]

3.3803

安全域 security domain

成员必须遵守由安全机构制定并管理的安全策略的领域。

[来源：ISO/IEC 10746-3:2009, 15.1.3]

3.3804

安全失效 security failure

系统行为或其缺失，导致安全可靠类别中的一项或多项要求未被满足。

[来源：IEEE 982:2024, 3.1]

3.3805

安全交互策略 security interaction policy

为实现不同安全域间的交互所必需的安全策略方面。

[来源：ISO/IEC 10746-3:2009, 15.1.4]

3.3806

安全内核 security kernel

与安全性有关的关键性语句的一个小的、自含的集合，作为操作系统的特权部分、为了使用一程序或访问某数据，必须满足核心所规定的全部准则。

3.3807

安全等级 security level

由分层安全密级与安全类别组合而成，表征客体的敏感程度或个人的安全许可权限。

注：最低安全级别表示所需的最低保护。

[来源：ISO/IEC 23643:2020, 3.17]

3.3808

安全政策 security policy

每个项目组织层面的“需知”和“信息获取”规则。

[来源：ISO/IEC/IEEE FDIS 24765, 2025]

3.3809

安全责任 security responsibility

服务客户和服务提供者的安全责任明确界定，安全事件由任何一方检测和报告的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.5.5]

3.3810

信息安全性测试 security testing

为评价测试项及相关数据和信息受到保护程度的一种测试，以确保未经授权的人员或系统不能使用、读取或修改它们，且不拒绝授权人员或系统的访问。

[来源：GB/T 38634.1-2021, 3.19]

3.3811

SEE 服务获取方 **SEE service acquirer**

获取 SEE 服务的参与者。

[来源：ISO/IEC 15940:2013, 2.12]

3.3812

SEE 服务供应商 **SEE service supplier**

提供 SEE 服务的参与者。

[来源：ISO/IEC 15940:2013, 2.13]

3.3813

软件工程环境服务 **SEE service; Software Engineering Environment Service; SEE service**

为支撑软件工程环境（SEE）的生存周期活动而提供的一项或多项服务操作。

[来源：ISO/IEC 15940:2013 系统和软件工程 软件工程环境服务, 2.8]

3.3814

种子 **seed**

组合测试中由用户定义应被测试的取值组合。

[来源：GB/T 38639-2020, 2.7]

3.3815

软件工程环境 **Software Engineering Environment; SEE**

为软件系统及相关领域（如项目管理、过程管理）的工程活动提供自动化系统上下文服务和软件专用服务的环境。

[来源：ISO/IEC 15940:2013, 2.7]

3.3816

程序段 **segment**

存储段

分段

构成整个系统的子系统之一或子系统的组合。

在存储分配中，计算机程序的独立部分，无需在主存储中维护整个程序即可执行。

数据的收集作为一个单元存储或传输。

在路径分析中，两个连续分支点之间的一系列计算机程序语句。

示例：金融系统的应付账款部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4156, 有修改]

3.3817

分段执行器 **segmented executor**

一组物理上不同的工件，执行器的物理分区。

注：每个段封装独立的状态，能够独立激活。每个段至少提供一个方面。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4316]

3.3818

选择性转储 **selective dump**

只指定存储单元区域的转储。

参见：变更转储、动态转储、内存转储、事后转储、快照转储、静态转储

3. 3819

选择性踪迹 selective trace

一种变量，它只引用可选的变量。

参见：执行踪迹、回顾踪迹、子例程踪迹、符号踪迹

3. 3820

自组装产品 self-assembly product

在没有主管或培训师帮助的情况下，仅依靠分步说明即可自行组装或搭建的独立组件产品。

注：包括以成套部件形式提供的产品和包含需要折叠、锁定或张紧的结构元件的产品（例如儿童推车或露营设备）。

Syn：扁平封装产品，即装即用产品。

[来源：IEC/IEEE 82079-1:2019, 3.35]

3. 3821

自包含 self-contained

不需要先前或后续处理步骤来启动或完成功能用户需求的过程。

[来源：GB/T 42449-2023, 1.755]

3. 3822

自描述性 self-descriptiveness

系统或部件包含足够的信息以解释它的目标和特性的程度。

参见：可维护性、可测试性、可使用性

3. 3823

自含[带]文档的 self-documented

关于源代码，它包含解释它的目标、操作和其他有用信息以了解和维护此代码的注释。

3. 3824

自学习系统 self-learning system

基于从试错实践中学习而改变其行为的适应性系统。

[来源：ISO/IEC TR 29119-11:2020, 3.1.69]。

3. 3825

自助供应 self-provisioning

服务支持按需自助服务的程度。

[来源：ISO/IEC TS 25025:2021, 3.1.8.4]

3. 3826

自相对地址 self-relative address

一种地址，它必须加至指令的地址以得到要访问的存储单元的地址。

参见：基地址、索引地址、偏移量、相对地址

3. 3827

卖方 seller

与采购方签订协议以提供产品或服务的组织或个人。

[来源：ISO/IEC/IEEE 12207:2017, 3.1.60]

3. 3828

卖家建议 seller proposals

卖方对征求建议书或其他采购文件的正式回应，具体说明价格、商业销售条款以及卖方将为请求组织提供的技术规格或能力，如果被接受，将约束卖方执行由此产生的结果协议。

[来源: ISO/IEC/IEEE 24765:2017, 3.3677]

3.3829

语义一致性 semantic agreement

被动互连, 指两个事物通过参照共同的事物或现象, 就陈述(符号排列)达成共同解释。

参见: 语法一致。

[来源: IEEE 1175.2-2006 IEEE CASE, 3.12]

3.3830

语义错误 semantic error

由在给定的语言中, 符号或符号组与它们的含义的关系的不了解导致的错误。

与 2.1641 条 句法错误(syntactic error) 相对。

3.3831

语义标签 semantic label

描述内容而非格式的标签。

[来源: ISO/IEC/IEEE 24765:2024; ISO/IEC/IEEE FDIS 24765, 3.4170]

3.3832

语义传递语言 semantic transfer language; STL

语义传递语言。

[来源: IEEE 1175.3-2004, X.X]

3.3833

语义传输语言 Semantic Transfer Language (STL); STL

用于表示软件应用程序行为描述的语言。

注: 语义传输语言具有计算机可解析的形式语法, 同时保持用户易于阅读和书写。

[来源: IEEE 1175.3-2004]

3.3834

语义变异点 semantic variation point

在元模型中, 语义发生变化的点。此种点为解释元模型的语义有意提供一定程度的自由。

[来源: GB/T 11457-2006 2.1432]

3.3835

语义 semantics

在给定的语言中, 符号或符号组与它们的含义之间的关系。

参见: 句法

3.3836

信号量 semaphore

用于同步并发进程的一种共享变量, 它通过指示某动作是否已完成或某事件是否已发生来同步。

3.3837

软件开发方法元模型 SEMDM

软件开发方法元模型。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4172]

3.3838

半形式验证 semi-formal verification

基于半形式化符号描述的方法。

[来源: ISO/IEC 23643:2020, 3.18]

3. 3839

系统工程管理计划 SEMP; Systems Engineering Management Plan; SEMP

用于管理系统工程工作的顶层技术计划，该计划定义项目技术工作如何组织、构建与实施，以及如何对系统工程过程进行控制，以提供满足干系人需求的产品。

[来源：ISO/IEC/IEEE 24748-4:2016, 4.14]

3. 3840

敏感性分析 sensitivity analysis

一种分析方法，通过将项目成果的变动与定量风险分析模型中各项要素的变动建立关联，确定单个项目风险或其他不确定性来源对项目成果具备最大潜在影响的程度。

[来源：项目管理知识体系指南（PMBOK(R) Guide）— 第七版]

3. 3841

情感 sentiment

从系统数据源推断出的情感值（正或负），例如通过自然语言处理对文本进行分析得到的情感值。

[来源：IEEE 7014-2024, 3.1]

3. 3842

独立文档 separate documentation

可以独立于软件使用的文档。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4181]

3. 3843

系统工程计划 systems engineering plan; SEP

用于管理系统工程工作的顶层技术计划，该计划定义项目技术工作如何组织、构建与实施，以及如何对系统工程过程进行控制，以提供满足干系人需求的产品。

同义词：系统工程管理计划，SEMP。

[来源：ISO/IEC/IEEE 24748-4:2016, 4.14]

3. 3844

顺序图 sequence diagram

一种表示对象之间按时间顺序排列的交互的图。

注：此种图展现了参与交互的对象和所交换消息的顺序。与协作图不同的是，时序图虽包括时间顺序，但不包括对象关系。时序图能以一般的形式（描述所有可能的场景），或以实例形式（描述一个实际场景）存在。时序图与协作图所表达的信息类似，但展现的方式不同。

参见：协作图

3. 3845

顺序的 sequential

两个或多个动作或事件以以下方式发生：一个必须在下一个开始之前完成。

参见：接续的、串行的

3. 3846

顺序聚类 sequential clustering

任务结构准则，其中约束顺序执行的对象被映射到一个任务。

[来源：ISO/IEC/IEEE 24765:202, 3.4184]

3. 3847

顺序内聚度 sequential cohesion

一类内聚度，在其中，一由软件模块执行的任务的输出作为由此模块执行的另一任务的输入。

与 2.229 条 一致内聚度(coincidental cohesion)、2.661 条 功能内聚度(functional cohesion)、2.874 条 逻辑内聚度(logical cohesion)和 2.1179 条 过程内聚度(procedural cohesion)相对。

3.3848

顺序进程 sequential processes

以这样一种方式执行的进程：在下一个动作开始之前本动作必须结束。与 2.299 条 并发进程(concurrent process)相对照。

[来源：GB/T 11457-2006, 2.1441]

3.3849

串行的 serial

整体的分别的部分，例如字符的位用同一设施顺序地传送、发生或处理。

与 2.1102 条 并行(parallel)(a)相对。

见：2.1438 条 顺序的(sequential)。

3.3850

串行总线接口 serial bus interface; SBI

用于双向串行数据通信的接口。

示例：RS232、RS485。

[来源：ISO/IEC/IEEE 24765e:2015]

3.3851

串行通信接口 serial communication interface; SCI

在微处理器与外设之间实现串行数据交换的部件。

[来源：ISO/IEC/IEEE 24765d:2015]

3.3852

串行构造 serial construct

由顺序步不包括决断或循环组成的程序结构。

同义词：顺序构造

3.3853

串行结构 serial construct

顺序构造 sequential construct

由一组不含判定分支或循环的步骤序列构成的程序结构。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4190]

3.3854

串行扩展接口 serial expansion interface; SEI

用于支持多条串行通道的连接接口。

[来源：ISO/IEC/IEEE 24765e:2015]

3.3855

串行外设接口(SPI) serial peripheral interface; SPI

嵌入式系统中用于两个设备之间的同步(全双工)串行通信接口。

[来源：ISO/IEC/IEEE 24765d:2015]

3.3856

服务型领导 servant leadership

通过专注于理解并满足团队成员的需求与发展，以实现团队最佳绩效的领导实践。

[来源：项目管理知识体系指南(PMBOK® 指南) — 第七版]

3. 3857

服务对象 server object

代表客户端对象执行某些服务的对象。

[来源: ISO/IEC 10746-2:2009, 13.4.6]

3. 3858

服务节点 server-side

节点、集群或胶囊, 其中: a) 包含或可能包含与计算服务器对象对应的基本工程对象, 以及支持涉及服务器对象的操作的通道中的存根、绑定器和协议对象; 或 b) 包含或可能包含一个协议对象, 该对象(可能通过与其他工程对象的交互)可以返回识别另一个服务器端的回复。

[来源: ISO/IEC 14752:2000, 3.3.10]

3. 3859

服务协议 service agreement

服务提供商和服务客户之间管理涵盖服务的书面协议。

注: 一份服务协议可以由一个或多个部分组成, 记录在一份或多份文件中。

[来源: ISO/IEC TS 25025:2021, 3.3.9]

3. 3860

服务替代方案 service alternative

被假定为在其生存周期内提供与另一个替代方案同等服务的替代方案; 为了简化比较, 忽略了所有收入现金流量, 仅显示了支出现金流量。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4199]

3. 3861

服务目录 service catalogue

有关组织向其客户提供的服务的文件化信息。

[来源: ISO/IEC TR 29110-1-2:2024, 3.86]

3. 3862

服务变更请求 service change request

提交配置项调整请求的正式程序。

参见: change request, request for change。

[来源: ISO/IEC 29110-1-2:2024, 3.87]

3. 3863

服务兼容性 service compatibility

服务可以与客户的系统或其他服务交换信息并执行其所需功能的程度。

[来源: ISO/IEC TS 25025:2021, 3.1.2]

3. 3864

服务组件 service component

服务的单个单元, 当与其他单元组合时将提供完整的服务。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4203]

3. 3865

服务交付策略 service delivery policy

包含组织的一般信念、道德规范、服务目标和服务宗旨的正式、简短和高级声明。

[来源: ISO/IEC TR 29110-1-2:2024, 3.88]

3. 3866

服务交付配置文件 service delivery profile

针对需要执行和管理服务交付流程的超小型企业（VSE）的配置文件，无论是针对他们开发的系统或软件产品还是由他人开发的。

[来源：ISO/IEC TR 29110-1:2016, 3.54]

3. 3867

服务设计 service design

服务解决方案的构建：通常包括实现预期功能的各类组件、支撑组件运行的技术架构、保障并管理该解决方案的相关流程、配套的考核指标（内部绩效指标或客户商定指标），以及供应链接口。

[来源：ISO/IEC 29110-1-2:2024，系统与软件工程-超小型实体生存周期规范 第 1-2 部分：术语，3.89]

3. 3868

服务台 service desk

面向客户的支持团队，负责集中处理与某项服务相关的事件、变更请求和投诉。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4207]

3. 3869

服务导出 service export

与发布服务报价的交易功能交互，通过将服务报价添加到一组已识别的服务报价中。

[来源：ISO/IEC 10746-3:2009, 14.5.1.2]

3. 3870

服务导入 service import

与搜索一组已识别服务的交易功能交互，以发现满足指定类型的服务可用的接口。

[来源：ISO/IEC 10746-3:2009, 14.5.1.3]

3. 3871

服务等级协议 service level agreement; SLA

服务提供商和客户之间确定服务和服务目标的书面协议。

[来源：ISO/IEC 29110-1-2:2024, 3.90]

3. 3872

服务等级协议信息完整性 service level agreement information completeness; SLA information completeness

服务等级协议包括建立服务质量水平所需的基本信息的程度。

参见：完整性。

注 1：基本信息包括覆盖服务、SLO（服务水平目标）和 SQO（服务质量目标）。

别名：SLA 信息完整性。

[来源：ISO/IEC TS 25052-1:2022, 3.1.8.5]

3. 3873

服务等级协议满意度 service level agreement satisfaction; SLA satisfaction

服务满足服务等级协议（SLA）规定的服务水平目标（SLO）和服务质量目标（SQO）的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.8.6]

[来源：别名：SLA 满意度]

3. 3874

服务等级目标（SLO） service level objective; SLO

服务提供商对服务的特定量化特性做出的承诺，其中值遵循区间尺度或比率尺度。

注：SLO 承诺可以表示为一个范围。

[来源：ISO/IEC TS 25052-1:2022, 3.3.8]

3.3875

服务可维护性 service maintainability

由指定维护人员对服务进行修改的有效性 with 效率程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.6]

3.3876

服务管理 service management

一组能力与流程，用于指导和控制服务提供商在服务的设计、转换、交付与改进方面的活动及资源，以满足服务需求。

[来源：ISO/IEC/IEEE FDIS 24765 系统与软件工程 术语, 3.4215]

3.3877

服务管理体系 service management system; SMS

用于指挥和控制一项或多项服务提供的管理体系。

注：服务管理体系包含服务设计、转换、交付、改进及满足要求所需的全部服务管理方针、目标、计划、过程、文档与资源。

[来源：ISO/IEC/IEEE 24765c:2014]

3.3878

服务管理系统 service management system; SMS

用于指导和控制一项或多项服务提供的管理体系。

注：服务管理体系（SMS）包含服务的设计、转换、交付和改进，以及满足相关要求所必需的全部服务管理方针、目标、计划、过程、文件和资源。

[来源：ISO/IEC/IEEE 24765:2014]

3.3879

服务经理 service manager; SM

直接监督服务交付、提供领导与方向指引，对所有活动拥有决策权，并且直接向组织最高层汇报者，或与组织最高层同级的角色。

注：服务经理在服务交付中可兼任多个角色（例如将控制经理与服务经理的职责赋予同一人员）。

[来源：ISO/IEC 29110-1-2:2024, 3.91]

3.3880

服务可测量性 service measurability

服务能够以计量方式提供，从而可对使用情况进行监测、管控、报告和计费的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.8.1]

3.3881

服务接口信息 service offer

描述接口的信息，如何绑定到它，以及可以使用它调用的服务。

[来源：ISO/IEC 10746-3:2009, 14.5.1.1]

3.3882

服务性能效率 service performance efficiency

服务在规定条件下满足性能要求的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.1]

3.3883

服务原语 service primitive

通道对象间交互的抽象定义，该定义导致通道内协议对象之间的协议交换。

[来源：ISO/IEC 14752:2000, 3.3.11]

3.3884

服务提供系统 service provision system

为用户提供 IT 服务的系统，包括人、过程、技术、设施和信息。

[来源：ISO/IEC TS 25011:2017, 3.3.6]

3.3885

服务可供性 service provisionability

服务按需、自动或与服务提供商进行最少交互即可为客户完成供给的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.8]

3.3886

服务质量定性目标 service qualitative objective; SQO

服务提供商针对服务的特定定性特性作出的承诺，其取值采用名义尺度或顺序尺度。

注：服务定性目标可采用枚举列表形式表述。定性特性通常需要人工解读。顺序尺度可用于表征特性的存在与否。

[来源：ISO/IEC TS 25052-1:2022, 3.3.7]

3.3887

服务质量特性 service quality characteristic

与服务质量相关的一类服务质量属性。

注：服务质量特性可细化为多个层级的子特性，最终细化为服务质量属性。

[来源：ISO/IEC TS 25052-1:2022, 3.2.3]

3.3888

服务可靠性 service reliability

服务在规定条件下、规定时间内完成规定功能的程度。

[来源：ISO/IEC TS 25052-1:2022, 3.1.4]

3.3889

服务请求 service request

对信息的请求，或对已完成风险评估的例行变更或程序的请求。

[来源：ISO/IEC/IEEE 15289:2019 系统与软件工程 生存周期信息项（文档），5.25]

3.3890

服务需求 service requirement

客户和服务使用者的需求，包括服务级别要求，以及服务提供者的需求。

[来源：ISO/IEC 20000-1:2011, 3.34]

3.3891

服务安全性 service security

服务保护信息和数据的程度，确保个人、其他产品或系统具有与其授权类型和级别相适应的数据访问权限。

[来源：ISO/IEC TS 25052-1:2022, 3.1.5]

3.3892

服务支持性 service supportiveness

根据服务级别协议中规定的支持范围，对客户活动提供支持服务的程度。

[来源: ISO/IEC TS 25052-1:2022, 3.1.9.1]

3.3893

服务可用性 service usability

在特定的使用周境下, 特定用户使用服务, 以有效性、效率和满意度实现特定目标的程度。

[来源: ISO/IEC TS 25052-1:2022, 3.1.3]

3.3894

服务 service

活动、工作或职责的履行。

注: 服务通常属于无形产品。服务可由其他服务构成。服务是自包含的、固有的、离散的, 可包含其他服务。

参见: 产品、结果、可交付成果

[来源: GB/T 22032-2021, 1.630]

3.3895

会话组件 session component

具有非持久性行为、瞬态和身份的 CORBA 组件。

[来源: ISO/IEC 19500-3:2012, 4.1]

3.3896

结算 settlement

开发方在项目验收后对项目的成本进行计算的过程。

[来源: GB/T 36964-2018]

3.3897

设置时间 set-up time

系统或部件正准备某种特定的操作的时间周期。

参见: 忙时间、停机时间、空闲时间、开机时间

3.3898

严重性 severity

故障的相对有害影响。

[来源: IEEE 982:2024, 3.1]

3.3899

软件工程 software engineering; SWE

系统运用科学技术知识、方法及实践经验, 开展软件的设计、实施、测试与文档编制工作。

[来源: ISO/IEC/IEEE 24748-5:2017, 3.16]

3.3900

开始到完成 start-to-finish; SF

“开始-结束”约束条件: 一项活动必须在另一项活动开始后才能完成。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4507]

3.3901

标准通用标记语言 standard generalized markup language; SGML

标准通用标记语言。

[来源: ISO/IEC/IEEE 23026:2023, 3.2]

3.3902

阴影类 shadow class

在某个其他视图中指定的视图中呈现的类。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4242]

3.3903

共享资产 shared asset

软件和系统工程生存周期中的数字制品构成了交付产品的一部分, 或者为工程过程提供支持, 以创建和维护产品成员。

示例: 机械、电气和软件的需求、设计规范或模型、源代码、构建文件或脚本、测试计划和测试用例、用户文档、维修手册和安装指南、项目预算、进度表和工作计划、产品校准和配置文件、机械物料清单、电气电路板和线束设计、工程管理计划、工程图纸、培训计划和培训材。

参见: 领域资产。

[来源: ISO/IEC 26580:2021, 3.17]

3.3904

共享资产超集 shared asset superset

共享资产的一种表示形式, 包含所有成员产品所需的全部内容。

[来源: ISO/IEC 26580:2021, 3.18]

3.3905

核心外层(外壳)程序 shell

在用户与计算机系统或程序之间提供接口的计算机程序或例程。

3.3906

合理成本估算 should-cost estimate

用于评估供应商建议价格合理性的产品或服务的成本估算。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4246]

3.3907

小外形薄型封装 shrink small outline package; SSOP

两侧具有鸥翼型引脚、较薄的矩形表面贴装集成电路器件。

[来源 1: ISO/IEC/IEEE 24765c:2014]。

3.3908

副作用 side effect

进行的处理或活动, 或得到的结果, 它们与程序、子程序、或操作的主要功能相比是处于第二位的。

[来源: GB/T 11457-2006, 2.1448]

3.3909

符号变更覆盖率 sign change coverage

在一组测试中, 具有正负激活值且同时被激活的神经元比例, 除以神经网络中神经元的总数(通常以百分比表示)。

[来源: ISO/IEC TR 29119-11:2020, 3.1.70]

3.3910

信号 signal

对在实例之间所递异步激励的规格说明。信号可有参数。

3.3911

信号接口 signal interface

所有交互都是信号的接口。

[来源: ISO/IEC 10746-3:200, 7.1.6]

3.3912

信号接口签名 signal interface signature

信号接口的接口签名。

注：信号接口签名包含一组有限的动作模板，一个用于接口中的每种信号类型。每个动作模板包括信号的名称、参数的数量、名称和类型，以及与实例化模板的对象相关的因果关系指示（启动或响应，但不是两者）。

[来源：ISO/IEC 10746-3:2009, 7.1.11]

3.3913

特征标记 signature

某一行为特征的名称和参数。特征标记可包括一个可选的返回参数。

3.3914

指引标识 signpost

帮助用户识别特定类型信息所在位置或当前显示中的信息适合整个文档的位置的文本、符号或小图形。

注：不同类型的信息可以用不同类型的符号或图形表示。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.45]。

3.3915

覆盖率等级 sign-sign coverage

如果通过改变每个神经元的符号可以显示单独导致下一层中的一个神经元改变符号而下一层中的所有其他神经元保持不变（即它们不改变符号），则达到覆盖水平。

[来源：ISO/IEC TR 29119-11:2020, 3.1.71]

3.3916

简单缓冲 simple buffering

一种缓冲技术，即在计算机程序整个执行期间为其分配一个缓冲区。

参见：动态缓冲、简单缓冲

[来源：ISO/IEC/IEEE FDIS 24765, 3.4258]

3.3917

简单知识组织系统 simple knowledge organization system; SKOS

简单知识组织系统。。

[来源：ISO/IEC/IEEE 26531:2023, 3.2]

3.3918

简单性 simplicity

系统或部件的设计和实现的简单和易于了解的程度。

参见：复杂性

3.3919

仿真 simulation

一个模型，当提供一组控制输入时，它的行为或操作像一给定的系统。

参见：模拟

3.3920

仿真组件 simulation component

用于虚拟样机系统仿真的组件。

注：它包含原子组件模型和复合组件模型。

[来源：GB/T 36457-2018, X.X]

3.3921

同时的 simultaneous

两个或多个事件在同一瞬间发生。

参见：并发

3.3922

同时递归 simultaneous recursion

两个软件模块相互调用的情况。

3.3923

单启动 single boot

只有一种模式启动计算机。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4265]

3.3924

单一继承 single inheritance

其中每一类型都只可有一个超类型的泛化的一种语义变化。

参见：多重继承

[来源：GB/T 11457-2006, 2.1458]

3.3925

单一实例方法 single instance approach

相同功能的每种交付方法只计算一次的情况。

参见：多实例方法。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.33]

3.3926

单地址指令 single-address instruction

仅包含一个地址字段的计算机指令。参见：多地址指令、二地址指令、三地址指令、四地址指令、零地址指令。

示例：将存储单元 A 中的内容装入累加器的指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3091]

3.3927

单规则决策表 single-hit decision table

任何一组条件将由一个且仅一个规则满足的决策表。

[来源：ISO 5806:1984, 3.2]

3.3928

单级编码 single-level encoding

一种微程序技术，其中，不同的微操作在微指令的相同字段编码为不同值。

与 2.1779 条 两级编码 (two-level encoding) 相对。

3.3929

单操作数指令 single-operand instruction

仅包含一个地址字段的计算机指令。参见：多地址指令、二地址指令、三地址指令、四地址指令、零地址指令。

示例：将存储单元 A 中的内容装入累加器的指令。

[来源：ISO/IEC/IEEE FDIS 24765, 3.3091]

3. 3930

单点估算 single-point estimating

一种估算方法，通过使用数据计算单一数值以反映最佳推测估计值。。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3. 3931

单步执行 single-step execution

依据外部信号，执行单条计算机指令或指令的一部分的一种调试技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4270]

3. 3932

单步操作 single-step operation; single-step execution; step-by-step operation

依据外部信号，执行单条计算机指令或指令的一部分的一种调试技术。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4270]

3. 3933

单值的 single-valued

当其多重性类型即上层属性设置为 1 时，修饰带有所定义多重性的一种模型元素。在任何时刻，术语单值都与属性、参数等所取的值的数目无关，原因是单值属性（例如，带有下界为零的多重性）可以没有值。与 2.999 条 多值的（multi-valued）相对。

[来源：GB/T 11457-2006, 2.1463]

3. 3934

单值属性 single-valued property

具有单值映射的属性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4271]

3. 3935

网站地图 site map

网站导航结构的文本或图形概览。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.21]

3. 3936

规模估计 sizing

对软件系统或部件所要求的源程序的行数或计算机存储的总量的估计过程。

[来源：GB/T 11457-2006, 2.1464]

3. 3937

skeleton

3. 3938

技能 skill

执行已掌握动作技能的可观察能力。

[来源：INCOSE SE Handbook, 第五版.]

3. 3939

切片 slice

停机时段的子集。。

[来源：IEEE 982:2024, 3.1]

3. 3940

冻结准备阶段 slush

功能或代码冻结前的准备阶段。

注：在此期间，开发人员可提交正在开发的代码，但不鼓励提交新的内容。若冻结期较长，可引入冻结缓冲期，通过允许提交部分额外内容来平稳度过冻结期。

[来源：ISO/IEC/IEEE FDIS 24765 系统与软件工程 术语，3.4289]

3.3941

中小企业 small and medium enterprise; SME

雇佣人数少于 250 人的企业。

[来源：ISO/IEC 29110-1-2:2024, 3.92]

3.3942

小型集成电路 small outline integrated circuit; SOIC

表面贴装集成电路（芯片）封装，带向外伸出的海鸥翼引脚。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4292]

3.3943

小外形封装 small outline package; SOP

两侧长边具有鸥翼引脚的矩形表面贴装集成电路器件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4293]

3.3944

快照转储 snapshot dump

一个或多个规定的存储区域的内容的动态转储。参见：2.196 条 变更转储（change dump）、2.519 条 动态转储（dynamic dump）、2.928 条 内存转储（memory dump）、2.1153 条 事后转储（postmortem dump）、2.1425 条 有选择性转储（selective dump）和 2.1582 条 静态转储（static dump）。

[来源：GB/T 11457-2006, 2.1466]

3.3945

监听 snoop

用于监控多层存储器缓存总线性能的技术。

示例：总线嗅探、缓存嗅探、总线监听。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4300]

3.3946

社会与伦理风险缓解 societal and ethical risk mitigation

人工智能系统减轻对社会潜在风险的程度。

注：社会与伦理风险缓解包括问责、公平性、透明性和可解释性、职业责任、促进人类价值、隐私、人类对技术的控制、社区参与和发展、尊重法治、尊重国际行为规范以及劳工实践。

[来源：ISO/IEC 25059:2023, 3.3.1]

3.3947

社会 society

泛指共同生活在社区或地缘政治界定区域内的人们。

注：可持续发展目标（SDGs）中使用的公民社会，以及地缘政治社会，即全社会。

[来源：ISO/IEC 25019:2023, 3.1.23]

3.3948

软拷贝图像 soft copy image

以音频或视觉格式非永久性输出的信息。

示例：视频显示器。

3.3949 [[来源: ISO/IEC 2382:2015, X.X]

软差错 soft error

发生一次或在不可预测的间隔发生的错误。

参见: 2.803 条 间歇故障 (intermittent fault) 和 2.1306 条 随机失效 (random failure)。

[来源: GB/T 11457-2006, 2.1467]

3.3950

软失效 soft failure

允许系统在部分运行能力下继续运行的故障。

[来源: GB/T 11457-2006, 2.1468, 有修改]

3.3951

软件获取 software acquisition

以获取软件产品或服务的决定为起点, 并以软件验收为结果的过程。

[来源: ISO/IEC/IEEE 41062:2024, 3.1.17]

3.3952

软件获取过程 software acquisition process

从决定购买软件产品开始, 到产品不再可用时结束的时间段。

[来源: ISO/IEC/IEEE 24765:2017 系统与软件工程 —— 术语]

3.3953

软件代理 software agent

感知环境并采取行动以最大限度地提高成功实现目标机会的数字实体。

[来源: ISO/IEC TR 29119-11:2020, 3.1.73]

3.3954

软件与系统产品线 software and systems product line; SSPL

软件与系统产品线。

[来源: ISO/IEC 26559:2017, 4]

3.3955

软件体系结构 software architecture

它是对系统整体结构设计的刻画, 包括全局组织与控制结构, 构件间通信、同步和数据访问的协议, 设计元素间的功能分配、物理分布、设计元素集成、伸缩性和性能、设计选择等。

[来源: GB/T 11457-2006, 2.1470]

3.3956

软件工件 software artifact

软件开发过程中创建的可由机器读取的有形单元或元素。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4312]

3.3957

软件即服务 software as a service; SaaS

通过瘦客户机接口或程序接口从客户端设备访问资源。

参见 IaaS、PaaS。

注: 提供给消费者的资源控制级别可能因服务提供者而异。在 DevOps 中, SaaS 可以作为 DevOps 管道的一部分进行自动化。

[来源: IEEE 2675-2021, 3.1]

3. 3958

软件资产 software asset

对组织具有潜在或实际价值的软件。

[来源：ISO/IEC 19770-1:2017 信息技术 ——IT 资产管理 第 1 部分：IT 资产管理体系 ——要求，3.50]

3. 3959

软件资产管理 software asset management; SAM

组织为从软件资产中实现价值而开展的协同活动。

注：在组织内部，贯穿软件资产生存周期全过程，对软件资产实施有效管理、管控与保护所必需的基础设施与流程。

[来源：ISO/IEC 19770-1:2017，3.51]

3. 3960

软件基线 software baseline

在软件生存周期的特定时间点，正式指定并确定的一组（一个或多个）软件配置项。

[来源：IEEE 828-2012 《IEEE 系统和软件工程配置管理标准》，2.1]

3. 3961

软件基线审核 software baseline audit

为验证基线是否与描述基线的文档相符，对软件基线库的结构、内容和设施进行的一种检查。

[来源：GB/T 11457-2006，2.1471]

3. 3962

软件基线库 software baseline library

存储配置项及其有关记录的仓库的内容。

[来源：GB/T 11457-2006，2.1472]

3. 3963

software behavior concepts

3. 3964

软件构造版 software build

软件系统或成分的一个可运行的版本，它包含最终的软件系统或成分将要提供的能力的一个规定子集。

[来源：GB/T 11457-2006，2.1473]

3. 3965

软件能力评估 software capability evaluation

为识别承制方是否有资格从事某项软件工作或监控现有软件工作所采用的软件过程的状态，由经过培训的专业人员组成的团队对软件能力所作的鉴定。

[来源：GB/T 11457-2006，2.1474]

3. 3966

软件特性 software characteristic

软件固有的可能是偶然的品质、质量或特性（例如，功能、性能、属性、设计限制、状态数、行数或分支数）。

[来源：GB/T 11457-2006，2.1475]

3. 3967

软件部件 software component

在特定分析层级下，系统内具有离散结构的实体，例如组件装配体或软件模块。

示例：数据，文档，模块，单元。

注：软部件指整体的组成部分，如软件产品的组成部分或软件标识标签的组成部分。

[来源：ISO/IEC 19770-5:2015, 3.36]

3.3968

软件配置 software configuration

软件产品在不同时期的组合。该组合随着开发工作的进展而不断变化。

[来源：GB/T 11457-2006]

3.3969

软件配置控制板 software configuration control board

负责评价和批准（或不批准）对配置项提出的更改，并负责保证那些已批准的更改能得到实施的一种组。

[来源：GB/T 11457-2006]

3.3970

软件配置项 software configuration item; SCI

已确立为配置项的软件实体。

（参见：计算机软件组件、计算机软件配置项、硬件配置项、软件项）。

注：当功能分配已明确区分设备功能与软件功能，且软件已被确立为可配置项时，计算机软件配置项即存在。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4319]

3.3971

软件配置管理 software configuration management; SCM

在整个软件生存周期中应用配置管理的过程，以帮助确保软件配置项（SCI）的完整性和正确性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4320]

3.3972

软件配置管理项 software configuration management item

置于软件配置管理之下的软件配置项的各种有关项目。包括各类管理文档、评审记录和文档、软件文档、源码及其可执行码、运行所需的系统软件和支持软件、以及各种有关数据等。

[来源：GB/T 11457-2006, 2.1480]

3.3973

软件使用者 software consumer

使用软件包授权权限的实体。

[来源：ISO/IEC 19770-5:2015 信息技术——IT 资产管理——第 5 部分：概述和术语, 3.37]

3.3974

软件更正 software correction

为解决问题、落实解决方案以弥补不足，并使软件具备满足规定运行要求所实施的变更。

注：“更正（correction）”这一术语主要用作一种维护类型，并用于对修改请求（MR）进行分类。参见：增强。

[来源：ISO/IEC/IEEE 14764:2021 软件工程 —— 软件生存周期过程 —— 维护，第 3.1.3 条]

3.3975

软件创建者 software creator

创建软件产品或软件包的个人或组织。

注：该实体可能并不拥有销售或分发软件的权利。

[来源：ISO/IEC 19770-5:2015 信息技术——IT 资产管理——第 5 部分：概述和术语, 3.38]

3.3976

软件数据库 software database

存放运行软件系统内部公共数据的数据定义及其当前值的文件。

[来源：GB/T 11457-2006]

3.3977

软件设计 software design

运用科学原理、技术信息和想象力来定义软件系统，以最高的经济性和效率实现预定功能。

[来源：ISO/IEC/IEEE 24765:2017 系统与软件工程 —— 术语]

3.3978

软件设计审计 software design audit

对软件产品进行评审，以确定其是否符合需求、标准及合同文件要求。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3979

软件设计理念 software design concept

可应用于系统设计的基本理念（如信息隐藏）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4326]

3.3980

软件设计描述 software design description; SDD

为便于分析、规划、实施及决策而创建的软件表现形式。

注：软件设计描述作为传递软件设计信息的媒介，可视为系统的蓝图或模型。

[来源：IEEE 1012-2024, 3.1.28]

3.3981

软件设计符号 software design notation

软件设计表示。描述软件设计的手段。

示例：结构图和伪代码。

注：它可以是图示的、符号的或文本的。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4328]

3.3982

软件设计符号 software design notation

软件设计表示法 software design representation

描述软件设计的预留手段。

示例：结构图和伪代码。

注：它可以是图解的、符号的或文本的。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.3983

软件设计验证 software design verification

评估设计以确定是否符合规定要求、设计标准、系统效率和其他标准。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.3984

软件开发人员 software developer

创建软件的人。

注：软件开发人员通常与软件制造商的其他开发人员合作创建商业应用程序。软件开发人员通常还可以作为软件的

内部开发人员工作，供软件开发人员自己的组织使用。

[来源：ISO/IEC/IEEE 24765e:2015]

3.3985

软件开发成本 software development cost

为达成软件项目目标开发方所需要支付的各种资源代价总和。

注：资源包包括人、财、物和信息等。

[来源：GB/T 36964-2018]

3.3986

软件开发周期 software development cycle

从决定开发软件产品开始，到软件完成开发结束的已交付的时间段。

参见软件生存周期。

注：该周期通常包括需求阶段、设计阶段、实现阶段、测试阶段阶段，有时还包括安装和检查阶段。上述阶段可能会重叠或迭代执行，具体取决于所采用的软件开发方法。该术语有时也用来表示一个更长的时间段，要么是当开发者不再对软件进行增强时结束的时段，要么是整个软件的生存周期。

[来源：ISO/IEC/IEEE 24765 2025, 3.4331]

3.3987

软件开发环境 software development environment

执行软件开发所需的设施、硬件、软件、程序和文档。

参见：赋能系统。

示例：可包括计算机辅助软件工程（CASE）工具、编译器、汇编器、链接器、加载器、操作系统、调试器、模拟器、仿真器、文档工具和数据库管理系统。

注 1：软件开发环境的计划可包括指定环境的构建地点，以及在哪些地点提供不同的环境或设施。例如，可要求在采购方和供应商的地点分别构建不同的测试环境。

[来源：ISO/IEC/IEEE 24748-5:2017, 3.14]。

3.3988

软件开发文件 software development file; SDF

与给定软件单元或一组相关单元的开发有关资料集合。。

注：内容通常包括这些单元的需求、设计、技术报告、代码清单、测试计划、测试结果、问题报告、进度表和备注。。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4333]

3.3989

软件开发文件 software development file; SDF

软件开发文件夹 software development folder

软件开发笔记本 software development notebook

单元开发文件夹 unit development folder

与给定软件单元或一组相关单元开发相关的材料集合。

注：内容通常包括需求、设计、技术报告、代码列表、测试计划和测试各单元的结果、问题报告、时间表和备注。

[来源：ISO/IEC/IEEE 24765 2025, 3.4333]

3.3990

软件开发收入 software development income

开发方向委托方交付软件开发工作成果所获得的收入。

[来源：GB/T 36964-2018]

3.3991

软件开发库 software development library

存放与软件开发工作有关的计算机可读信息和人们可读信息的软件库。

[来源: GB/T 11457-2006, 2.1487]

3.3992

软件开发库 software development library

项目库 project library

程序支持库 program support library

包含与软件开发工作相关的计算机可读和人类可读信息的软件库。

参见: 生产库(production library)、软件仓库(software repository)、系统库(system library)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4334]

3.3993

软件开发方法 software development method

软件开发过程所遵循的方法和步骤。它是规则、方法和工具的集成, 既支持开发, 也支持以后的演化过程。

[来源: GB/T 11457-2006]

3.3994

软件开发计划 software development plan; SDP

用于描述一项软件开发工作所应遵循的技术路线的信息项。

参见: 项目管理计划(project management plan)。

注: 软件开发计划说明组织或项目计划如何开展开发活动。其中对技术路线与管理路线进行了区分。

[来源: ISO/IEC/IEEE 24748-5:2017, 3.15]

3.3995

软件开发过程 software development process

将用户需求转化为软件产品的过程。

注: 该过程包括将用户需求转化为软件需求, 将软件需求转化为设计, 用代码实现设计, 测试代码, 有时还包括安装和检查软件以供操作使用。这些活动可以重叠进行, 也可以迭代执行。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4336]

3.3996

软件多样性 software diversity

一种软件开发技术, 其中, 两个或多个功能相同的部分由不同的程序员或程序员组用同一个规格说明进行开发, 目的是提供出错检测, 增加可靠性、附加的文档或减少影响结果的编程或编译错误的概率。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4337]

3.3997

软件元素 software element

软件的系统要素。参见: 系统要素, 软件/系统要素。

[[来源: ISO/IEC/IEEE 90003:2018, 3.8]

3.3998

软件工程经济学 software engineering economics

从经济学的观点来研究、分析如何有效的开发、发布软件产品和支持用户使用这一产品的学科。

[来源: GB/T 11457:2006 2.1495]

3.3999

软件工程组 software engineering group; SEG

负责项目的软件开发和维护活动（即需求分析、设计、编码和测试）的人员（包括经理和技术人员）的集合。软件工程组不包括执行软件有关工作的组，例如软件质量保证组，软件配置管理组和软件工程过程组。

[来源：GB/T 11457:2006 2.1497]

3.4000

软件工程过程组 software engineering process group; SEPG

负责组织所用软件过程的定义、维护和改进工作的专家组。在关键实践中，这个组一般称作“负责组织的软件过程活动的组”。

[来源：GB/T 11457:2006 2.1498]

3.4001

软件授权 software entitlement

软件许可使用权利，如软件许可方与软件消费者之间通过协议所定义。

注：有效使用权利需考虑所有合同和所有适用的许可，包括完整许可、升级许可和维护协议。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4341]

3.4002

软件授权核对 software entitlement reconciliation

将组织所拥有的软件授权与所需授权（已授予及已部署）进行比对的过程，通常用于确定是否符合软件许可协议。

[来源：ISO/IEC 19770-3:2016 信息技术——IT 资产管理——第3部分：授权模式，3.1.30]。

3.4003

软件失效 software failure

软件系统丧失完成规定功能能力的事件。

[来源：GB/T 37691-2019]

3.4004

软件特征 software feature

区分软件项的特征（例如，性能、可移植性或功能）。

由需求文档规定或隐含的软件特征（例如，功能、性能、属性或设计约束）。

[来源：GB/T 11457-2006, 2.1501]

3.4005

软件功能 software function

在软件中实现算法，最终用户或软件可以执行部分或全部工作任务。

注：一个功能不需要最终用户可调用（例如，自动备份或数据保存）。

[来源：ISO/IEC 25051:2014, 4.1.14]

3.4006

软件风险 software hazard

事故发生的前提条件软件状态。

[来源：IEEE 1228-1994, 3.1.5]

3.4007

软件识别标签 software identification tag; SWID

包含关于软件配置项识别信息的信息结构。

[来源：ISO/IEC 19770-6:2024, 3.1.6]

3.4008

软件在环仿真 software in the loop simulation; SILS

软件在环仿真。

[来源：IEEE 1012-2024, 3.2]

3.4009

软件集成 software integration

将选定的软件成分集成在一起，旨在提供最终软件系统具备的一组能力或者规定的一组子能力的过程。

[来源：GB/T 11457:2006 2.1502]

3.4010

软件项 software item

源代码、目标代码、作业控制代码、控制数据或这些项的集合。

[来源：GB/T 11457-2006, 2.1503]

3.4011

软件库 software library

软件和有关的文档说明的一个受控制的集合。目的是有助于软件开发、使用或维护。

示例：类型包括主库、产品库、软件开发库、程序库、软件储藏仓和系统库。参见：2.1236 条 程序库（program library）。

[来源：GB/T 11457-2006, 2.1505]

3.4012

软件库 software library

程序库 program library

为辅助软件开发、使用或维护而设计的软件及相关文档的有控制收集。

参见：存储库（repository）。

同义词：程序库（program library）。

示例：类型包括主库（Types include master library）、生产库（production library）、软件开发库（software development library）、系统库（system library）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4348]

3.4013

软件许可证 software license

根据软件许可方规定的条款和条件使用软件的合法权利。

注：使用软件产品可包括访问、复制、分发、安装和执行软件产品，具体取决于许可条款和条件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4349]

3.4014

软件被许可人 software licensee

获得使用特定软件产品许可的个人或组织。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4350]

3.4015

软件许可方 software licensor

拥有或持有特定软件包软件许可证颁发权的个人或组织。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4351]

3.4016

软件生存周期 software life cycle; SLC

由用户需求或感知到的客户需求所启动，至产品停止使用或软件不再可用时终止的软件系统或软件产品周期。

[来源：ISO/IEC/IEEE 24765e:2015]

3.4017

软件维护 software maintenance

为软件系统提供支持所需的所有活动。

示例：为发布包添加、更改或删除源代码并重新生成可执行文件。交付前活动包括规划交付后操作、可支持性设计及后勤保障方案制定。交付后活动包括软件变更、人员培训以及运维服务台值守。

注 1：软件维护指仅依赖系统开发环境所具备的专业技能、设施及其他资源才能开展的活动，运行环境一般不具备、无需配备也不常规提供这类资源与能力。

维护工作通常与在线业务运行并行开展，此种情况下，维护所耗费的时长与人力不计入系统停机时间。

[来源：ISO/IEC/IEEE 14764:2021，软件工程 — 软件生存周期过程 — 维护，3.1.12]

3.4018

软件非功能性评估过程 (SNAP) 类别 software non-functional assessment process (SNAP) category

用于度量软件非功能规模的一组组件、过程或活动。

注：类别对方法进行分类，且具有足够的通用性以适配未来技术。类别可划分为子类别，SNAP 类别根据相似操作或相似规模度量活动对子类别进行分组。

[来源：ISO/IEC/IEEE 32430:2025，3.1.34]

3.4019

软件非功能评估过程 (SNAP) 点 Software Non-functional Assessment Process (SNAP) point
用于表示软件非功能规模的度量单位。

注：软件非功能性评估过程的非功能性规模 (SNAP) 计数单位 (SCU) 在子类别中标识。在识别出所有满足非功能需求 (NFR) 的 SCU 后，将所有 SCU 的 SNAP 点 (SP) 相加，得到计算出的 SP。同义：SNAP 评估点。

[来源：ISO/IEC/IEEE 32430:2025，3.1.36]

3.4020

软件非功能评估过程 (SNAP) 子类别 Software Non-functional Assessment Process (SNAP) sub-category

项目中为满足非功能需求而执行的组件、过程或活动。

注：一个非功能需求可使用多个子类别进行规模度量。

同义：SNAP 子类别。

[来源：ISO/IEC/IEEE 32430:2025，3.1.37]

3.4021

软件非功能评估过程类别 software non-functional assessment process category; SNAP category

用于测量软件非功能规模的一组组件、过程或活动。

注：类别对方法进行分类，并具有足够的通用性以适应未来技术。类别划分为子类，从而 SNAP 类别基于相似的操作或相似类型的规模测量活动对子类进行分组。

[来源：ISO/IEC/IEEE 32430:2025，3.1.34]

3.4022

软件非功能评估过程计数单元 software non-functional assessment process counting unit; SCU

测量非功能规模的组件、过程或活动。

注：软件非功能评估过程计数单元根据每个子类的性质进行识别。它可包含功能和功能性特性；因此，软件非功能评估过程计数单元的规模测量包括使用功能点分析（FPA）进行的功能规模测量，以及使用 SNAP 进行的非功能规模测量。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.35]

3.4023

软件非功能评估过程点 software non-functional assessment process point; SP
用于表示软件非功能规模的度量单位。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.36]

3.4024

软件非功能评估过程子类 Software Non-functional Assessment Process sub-category; SNAP sub-category

在项目内执行的组件、过程或活动，以满足非功能性需求。

注：非功能性需求可使用一个以上的子类进行规模测量。

[来源：ISO/IEC/IEEE 32430:2025, 3.1.37]

3.4025

软件包 software package

一种完备并配有文档，供给有类同应用或功能要求的若干用户的程序集。

参见：软件产品（software product）。

注：与特定业务功能集相关联的文件集，可安装在计算设备上，并具有特定的许可要求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4359]

3.4026

软件打包器 software packager

对他人创建的软件进行打包或捆绑的实体。

示例：将软件包与嵌入式系统结合使用的增值经销商，或获得许可将多个不同软件产品组合成单一捆绑包的软件经销商。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4360]

3.4027

软件盗版 software piracy

非法使用或复制软件产品。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4361]

3.4028

软件计划 software plans

描述软件开发和(或)软件维护活动将如何进行的各类正式计划或非正式计划。例如软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

[来源：GB/T 11457:2006 2.1511]

3.4029

软件过程 software process

由组织或项目使用的，用以计划、管理、执行、监控和改进其软件相关活动的过程或过程的集合。

[GB/T 11457-2006, 2.1512]。

3.4030

软件过程评估 software process assessment

为了确定组织当前软件过程的状态、组织面临的急需处理的软件过程的有关问题，并获得组织对软件过程改进的支持，由一个经过培训的软件专业人员组成的团队对软件过程所作的评价。

[GB/T 11457-2006, 2.1513]。

3.4031

软件过程描述 software process description

在项目定义的软件过程或组织的标准软件过程中被标识的主要软件过程成分的操作性定义。它通过文档以一种完备的、精确的、可验证的方式记载软件过程的需求、设计、行为或其他特征。

参见：XXX 条 过程描述 (process description)。

[GB/T 11457-2006, 2.1514]。

3.4032

软件过程元素 software process element

描述软件过程的构成元素。每个过程元素包括一组妥善定义的、界定的、密切相关的任务（例如软件估计元素、软件设计元素、编码元素和同行评审元素）。过程元素的描述可以是待填入的模板、待完成的片段、待精炼的抽象概念、或者待修改的或已使用而无需修改的完整描述。

[GB/T 11457-2006, 2.1515]。

3.4033

软件过程改进计划 software process improvement plan

从软件过程评估建议中导出的一种计划，它标识为改进软件过程而将采取的具体措施，并且概述实施这些措施的计划。

[GB/T 11457-2006, 2.1516]。

3.4034

软件过程改进建议 software process improvement proposal

一个文档化的、对过程或过程有关项所作的、将提高软件过程能力和性能的更改建议。

参见：XXX 条 措施建议 action proposal。

[GB/T 11457-2006, 2.1517]。

3.4035

软件过程成熟度 software process maturity

一个具体过程被明确定义、管理、测量、控制以及有效的程度。成熟度隐含了能力上的增长潜力，并揭示组织的软件过程的丰富性和在全组织的所有项目中采用这些过程的一致性。

[GB/T 11457-2006, 2.1518]。

3.4036

软件过程相关文档 software process-related documents

预期未来项目在剪裁组织的标准软件过程时可能使用的实例文档或文档片段。实例文档可能包括诸如项目定义的软件过程、标准、规程、软件开发计划、测量计划和过程培训材料等。

[GB/T 11457-2006, 2.1519]。

3.4037

软件产品 software product

一组计算机程序、规程以及可能的相关文档和数据。

注 1：产品包括中间产品和意图用于开发方和维护方等用户使用的产品。

注 2：在 SQuaRE 标准中，软件质量和软件产品质量具有同样的含义。

[来源：GB/T 25000.1-2021，2.40]

3.4038

软件产品评估 software product evaluation

按照规程对软件产品的一个或多个特性进行评估的技术操作。

[来源：ISO/IEC/IEEE FDIS 24765，3.4363]

3.4039

软件项目 software project

一项需要协同工作的任务，它关注系统中软件成分的分析、说明、设计、开发、测试和（或）维护等工作及其相关文档。一个软件项目可以是建造硬件/软件系统的项目的一部分。

[GB/T 11457-2006，2.1521]。

3.4040

软件项目生存周期 software project life cycle; SPLC

适用于特定项目的整个软件生存周期的一部分。

注：这是通过将活动映射到选定的软件项目生存周期模型（SPLCM）上而创建的一系列活动。

[来源：ISO/IEC/IEEE FDIS 24765，3.4365]

3.4041

软件项目生存周期（SPLC） software project life cycle (SPLC); SPLC

适用于特定项目的整个软件生存周期的一部分。

注：它是通过将活动映射到所选软件项目生存周期模型（SPLCM）而形成的活动序列。

[来源：ISO/IEC/IEEE 24765:2017 系统和软件工程 术语]

3.4042

软件项目生存周期模型 software project life cycle model; SPLCM

各使用组织选择的框架，用于将 IEEE Std 1074 的活动映射到该框架上，从而产生软件项目生存周期（SPLC）。。

[来源：ISO/IEC FDIS 24765, 3.4366]

3.4043

软件项目生存周期模型（SPLCM） software project life cycle model (SPLCM); SPLCM

各使用组织所选定的框架，用于对 IEEE Std 1074 中的活动进行映射，以形成软件项目生存周期（SPLC）。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.4044

软件项目生存周期过程 software project life cycle process; SPLCP

通过将组织过程资产（OPAs）添加到软件项目生存周期（SPLC）及 OPAs 中，形成的针对项目特定过程的描述。。

[来源：ISO/IEC FDIS 24765, 3.4367]

3.4045

软件项目生存周期过程（SPLCP） software project life cycle process (SPLCP); SPLCP

通过将组织过程资产（OPA）添加到软件项目生存周期（SPLC）及组织过程资产（OPA）中，所形成的、面向具体项目的过程描述。

[来源：ISO/IEC/IEEE 24765:2017 系统和软件工程 术语]

3.4046

软件项目管理计划信息 Software Project Management Planned Information; SPMPI

软件项目管理计划信息。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.4047

软件质量 software quality

在规定条件下使用时, 软件产品满足明确和隐含要求的能力。

注: 质量取决于既定要求在多大程度上准确反映了利益相关者的需求、期望和要求。在 SQuaRE 标准中, 软件质量与软件产品质量含义相同。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4368]

3.4048

软件质量保证 software quality assurance; SQA

一组用于定义和评估软件过程充分性的活动, 旨在提供证据, 以确立对软件过程的适用性及其所生成软件产品满足预定用途质量要求的信任程度。

注1: 软件质量保证(SQA)的关键特性是其相对于项目的功能客观性。

注2: 软件质量保证功能在组织上可独立于项目, 即不受项目的技术、管理和财务方面的压力影响。

[来源: IEEE 730-2014, 3.2]

3.4049

软件质量特性 software quality characteristic

支撑软件质量的软件质量属性的类别。

注: 软件质量特性可细化为多级子特性, 最终细化为软件质量属性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4370]

3.4050

软件质量控制 software quality control; SQC

贯穿项目生存周期, 测量、评价和报告软件项目工件质量的一组活动。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) —第五版]

3.4051

软件质量控制(SQC) software quality control (SQC); SQC

在整个项目生存周期中, 对软件项目成果的质量进行测量、评估和报告的一组活动。

[来源: 项目管理知识体系指南 (PMBOK(R) Guide) —第五版]

3.4052

软件质量目标 software quality goal

为软件工作产品确定的定量的质量目标。

[来源: GB/T 11457:2006, 2.1524]

3.4053

软件使用质量 software quality in use

在特定使用周境中, 软件产品使得特定用户在达到有效性、效率和满意度等方面的特定目标的能力。

注: 软件产品发布前, 对于期望的用户、目标和使用周境, 使用质量可在某测试环境中被规定和测试。一旦进入使用, 使用质量可以针对实际的用户、目标和使用周境进行测量。用户实际的要求可能与需求规格说明中的预先定义的不一致, 所以实际的使用质量可能不同于测试环境中早期所测的使用质量。

[来源: GB/T 25000.40-2018, 4.59]

3.4054

软件质量管理 software quality management

为指导和控制组织在软件质量方面的活动而进行的协调。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4373]

3.4055

软件质量测量 software quality measure

对软件质量的内部测度、外部测度和软件使用质量测度的测量。

[来源: GB/T 25000.40-2018, 4.60]

3.4056

软件质量计量 software quality metric

一种项具有给定的质量属性的程度的定量度量。

一种函数, 它的输入是软件数据, 输出是能解释该软件具有给定的质量属性的程度的单一个数值。

参见: 质量计量

[来源: GB/T 11457-2006, 2.1526]

3.4057

软件质量要求 software quality requirement

要求软件中须具备某种软件质量属性的需求。

[来源: ISO/IEC/IEEE 24765:2017, 3.3842]

3.4058

软件发布管理 software release management

对向一个或多个客户发布一个或多个软件版本所开展活动的管理, 这些活动包括产品组成部分的标识、打包和交付。

参见: 软件配置管理, 版本。

[来源: IEEE 828-2012, IEEE 系统与软件工程配置管理标准, 2.1]

3.4059

软件可靠性 software reliability

在规定条件下, 在规定的时间内软件不引起系统失效的概率。该概率是系统输入和系统使用的函数, 也是软件中存在的缺陷的函数。系统输入将确定是否会遇到已存在的缺陷 (如果有缺陷存在的话)。

在规定的周期内所述条件下程序执行所要求的功能的能力。

[来源: GB/T 11457-2006, 2.1528]

3.4060

软件可靠性工程 software reliability engineering; SRE

在系统开发和运行期间, 对收集的数据应用统计技术, 以规定、估算或评估基于软件的系统可靠性。

[来源: IEEE 982:2024, 3.1]

3.4061

软件可靠性管理 software reliability management

通过一套方案来优化软件可靠性的过程, 该方案着重于软件差错预防、故障检测与排除, 并通过运用度量手段, 在项目资源 (成本)、进度和性能等条件的约束下, 最大限度地提升软件可靠性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4377]

3.4062

软件储藏库 software repository

为软件和相关文档提供永久性存档存储的软件库。

参见：生产库、软件开发库、系统库。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4378]

3.4063

软件需求 software requirement

用户解决问题或实现目标所需的软件能力。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4379]

3.4064

软件需求评审 software requirement review; system requirements review; SRR

一种对一个或多个软件配置项规定的需求的评审，它评价对系统需求的响应和解释以确定它们能否形成进一步的配置项的初步设计的满意的基础。参见：XXX 条 系统需求评审（system requirement review）。

[来源：GB/T 11457:2006, 2.1531]

3.4065

软件需求分析 software requirements analysis

研究用户需求以得出系统、硬件或软件需求定义的过程。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4066

软件需求工程 software requirements engineering

软件需求的获取、分析、规范、验证和管理 参见：需求工程 条目。

注：它涉及使用定义、分析、权衡研究和原型制作的迭代过程，将系统需求转化为软件需求、性能参数和软件配置的描述。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4067

软件需求管理 software requirements management

对需求从系统层到模块或部件层的识别、分配与逐级传递进行策划和控制的过程，包括接口、验证、修改及状态监控。

参见：需求管理。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4068

软件需求阶段 software requirements phase

在软件开发生存周期阶段中，对软件产品的需求（如功能和性能能力）进行定义、形成文档并开展评审的阶段。

[来源：ISO/IEC/IEEE 24765-2025, 3.4383]

3.4069

软件需求评审（SRR） software requirements review (SRR); SRR

对一个或多个软件配置项的既定需求进行评审，评估其对系统需求的适配度与转解读是否合理，并判定这些需求能否作为软件配置项概要设计开展的合格依据（参照：系统需求评审）。

注：这种审查被美国国防部称为软件规范审查。

[来源：ISO/IEC/IEEE 24765-2025 系统和软件工程——词汇, 3.4383]

3.4070

软件需求规格说明 software requirements specification; SRS

软件及其外部接口的基本需求的结构化集合。

[来源: GB/T 45802-2025]

3.4071

软件需求验证 software requirements verification

为保障软件系统或服务持续正常运行,对软件系统开展支持、维护与运营的一系列活动。

[来源: ISO/IEC/IEEE 24765-2025, 3.4386]

3.4072

软件安全性 software safety

软件运行不引起系统事故的能力。

[来源: GB/T 37691-2019, 3.3]

3.4073

软件安全计划 software safety program

降低软件风险的系统化方法。

[来源: IEEE 1228-1994, 3.1.7]

3.4074

软件信息安全性 software security

软件保护其资产免受恶意攻击者侵害的能力。

注: 软件安全性适用于软件资产,并可分解为以下一组属性: 保密性、完整性、可用性、身份鉴别、授权和不可否认性。

[来源: ISO/IEC 23643:2020, 3.22]

3.4075

软件服务 software service

实施与软件产品有关的活动、工作或义务,比如软件开发、维护和运作。

[来源: GB/T 11457:2006 2.1533]

3.4076

软件规格说明评审 software specification review; SSR

软件规格说明评审。

[来源: ISO/IEC/IEEE 24748-8:2019, 3.2]

3.4077

软件支持 software support

在运行环境中,为确立、保持、修改或移除系统按预期使用所必需的条件与资源而开展的在用活动。

参见: 软件维护

注: 支持活动仅限于可利用运行环境中现有的技能、工具及其他资源完成任务。软件支持可由制造商、营销机构、供应商或其他机构提供。在合同约定的特殊情况下,可允许用户自行维护或增强软件。

[来源: IEEE 982:2024, 3.1]

3.4078

软件持续保障 software sustainment

为支持、维护和运行软件系统而开展的一系列活动,以确保该软件系统或服务保持可操作性。

注: 软件持续保障包括过程、规程、人员、物资和信息。

[来源: ISO/IEC/IEEE 14764:2021, 3.1.13]

3.4079

软件系统 software system

软件对利益相关者至关重要的系统。参见: 软件密集系统。

[来源: ISO/IEC/IEEE 12207:2017, 3.1.55]

3.4080

软件系统元素 software system element

构成软件系统的一组元素中的成员。

注: 软件系统元素可包含一个或多个软件单元、软件元素、硬件单元、硬件元素、服务以及其他系统元素和系统。

软件系统元素可视为一种系统元素。

[来源: ISO/IEC/IEEE 12207:2017, 3.1.56]

3.4081

软件测试环境 software test environment; STE

对软件进行合格性测试或其他测试时所需要的设施、硬件、软件、固件、规程和文档集等。

[来源: GB/T 25000.51-2016, 4.1.15]

3.4082

软件测试待查事件 software test incident

在软件测试执行期间发生的要求调查的任一事件。

[来源: GB/T 11457-2006, 2.1536]

3.4083

软件测试 software testing

对程序在有限测试用例集上提供预期行为的动态验证, 该用例集需从通常无限的执行域中合理选取。

[来源: ISO/IEC 25051:2014, 4.1.15]

3.4084

软件工具 software tool

一种计算机程序, 用来帮助开发、测试、分析或维护一计算机程序或它的文件。例如, 比较器、交叉引用生成器、反编译程序、驱动程序、编辑程序、流程图器、监控程序、测试用例生成器、定时分析器。

[来源: GB/T 11457-2006, 2.1537]

3.4085

软件单元 software unit

一段可分开编译的代码。

[来源: GB/T 11457-2006, 2.1538]

3.4086

软件使用 software usage

软件使用许可项下的消耗量, 其计量方式遵循该许可项的条款和条件。

注: 根据具体条款和条件, 使用行为可包括访问、复制、分发、安装和执行软件。

[来源: ISO/IEC 19770-5:2015, 3.47]

3.4087

软件文档 software user documentation

一套文档, 它为使用系统或部件以期获得所希望结果的最终用户提供系统指令方面的信息。例如, 用户手册。参考: 11457 2.1809 条 用户文档 (user documentation)。

[来源: ISO/IEC/IEEE 26513:2017, 3.35]

3.4088

软件版本标识 software version ID

插入到每个配置项（包括每一个单独发布的版本）中的明确且不可变的版本标识符（名称或编号），可用于在任何实例或存储库中识别配置项的确切版本。

[来源：IEEE 828-2012 系统与软件工程中的配置管理 IEEE 标准，2.1]

3.4089

软件版本标识 software version ID (software version identification)

插入到每个配置项中的明确且不可变的版本标识符（名称或编号），包括每个单独的版本，可用于识别任何实例或存储库中配置项的确切版本。

[来源：IEEE 828-2012，2.1]

3.4090

软件工作产品 software work product

作为定义、维护或实施软件过程的一部分而生成的任何制品，包括过程说明、计划、规程、计算机程序和相关的文档等，无论是否打算将它们交付给顾客或者最终用户。

参见：软件产品（software product）。

[来源：GB/T 11457:2006，2.1539]

3.4091

系统元素 software/system element

组成系统的一组元素中的一个成员。

示例：硬件、软件、数据、人、过程（例如提供给用户服务的过程）、规程（例如操作指南）、设备、原料、自然存在的实体或其任意组合。

注：系统元素是系统中离散的部分，通过实现它可以完成规定的需求。

[来源：GB/T 22032-2021，4.1.47]

3.4092

基于软件的系统 software-based system

由软件控制的计算机系统。

[来源：ISO/IEC/IEEE FDIS 24765 3.4403]

3.4093

软件相关组 software-related group

从事支持（但不直接负责）软件开发和（或）维护工作的软件工程人员（包括经理和技术人员）的集合。例如软件质量保证组和软件配置管理组。

[来源：GB/T 11457:2006，2.1527]

3.4094

软件发布管理计划信息 software release management planned information; SRMPI

软件发布管理计划信息。

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.4095

软件 software; SW

用于运行计算机的程序或程序集。

[来源：ISO/IEC/IEEE 26514:2022，4.46]

3.4096

解域 solution domain

解决方案或解决方案集所处的环境。

参见：问题域。

[来源: ISO/IEC/IEEE FDIS 24765 , 3.4409]

3.4097

分类 sort

表示集合名称的符号。

[来源: ISO/IEC 15909-1:2019 , 3.33]

3.4098

排序生成器 sort generator

由给定标识（作为参数传入）构建而成的、用于生成新类型与操作符的生成器。

[来源: ISO/IEC 15909-3:2021, 3.7]

3.4099

分类 sorting

事务功能中行或记录的排序活动。

[来源: GB/T 42449-2023, 3.48]

3.4100

源地址 source address

要传送的数据所在的设备或存储单元的地址。与 2.460 目标地址（destination address）相对。

[来源: GB/T 11457-2006, 2.1540]

3.4101

源[代]码 source code

以适合于作为汇编程序、编译程序或其他转换程序输入的形式表示的计算机指令和数据定义。

注: 源程序由源代码构成。与 2.1031 条 目标代码（object code）相对。

[来源: GB/T 11457-2006, 2.1541]

3.4102

源代码生成器 source code generator

一种软件工具,它接受计算机程序的需求或设计作为输入并产生实现需求或设计的源代码。同义词:

2.1542 条 源代码生成程序（source code generator）。

参见: XXX 条 代码生成器（code generator）。

[来源: GB/T 11457:2006, 2.1542]

3.4103

源代码组织 source code organization

源代码的组织方式,包括单个文件内代码的布局以及源代码的模块化、类化、物理文件化等包装方式。

[来源: ISO/IEC/IEEE FDIS 24765 3.4421]

3.4104

源文件 source document

定义现有行业实践的标准或文件。

[来源: ISO/IEC 19770-8:2020, 3.2]

3.4105

源文档 Source documents

与感兴趣的系统采购活动的特定阶段相关的外部文件。包括与组织战略和政策相关的源文件中体现的书面指令。

[来源: INCOSE 系统工程手册 第四版 附录 E]

3.4106

源语言 source language

输入至机器辅助转换过程的语言。例如，用于写计算机程序的语言。与 2.1676 条 目标语言(target language) 相对。

[来源：GB/T 11457-2006，2.1543]

3.4107

源代码行数 source lines of code; SLOC

源代码行数，指程序在编译前的编程语言代码行数。

[来源：(ISO/IEC 20968:2002，10)]

3.4108

源节点 source node

与弧起点关联的节点。

[来源：ISO/IEC 15909-2:2011，4.1.17]

3.4109

源程序 source program

必须被编译、汇编或其他转换的计算机程序以被计算机执行。与 2.1678 条 目标程序(target program) 相对。

[来源：GB/T 11457-2006，2.1544]

3.4110

源语句 source statements; SS

软件产品的编码逻辑。

注：源语句可按功能分为可执行语句、数据声明语句、编译指令或注释语句；也可划分为可交付项或不可交付的。

[来源：ISO/IEC/IEEE FDIS 24765，3.4426]

3.4111

空间极小实体 space very small entity; space VSE

从事空间领域开发或维护系统或软件、且人数不超过 25 人的企业、组织（如政府机构、非营利组织）、部门或项目。

[来源：ISO/IEC 29110-1-2:2024，3.97]

3.4112

跨度 span

起止时刻均可测量的在役区间。

[来源：IEEE 982:2024，3.1]

3.4113

分割操作 spawn action

分割操作，其中已启用的链条不会合并。

[来源：ISO/IEC 10746-2:2009，13.1.6]

3.4114

特殊原因 special cause

非系统固有、不可预测且间歇性出现的变异源。

参见：普通原因。

示例：该原因可归因于系统缺陷。在控制图上，其表现为超出控制限的点或控制限内的非随机模式。

[来源：ISO/IEC/IEEE FDIS 24765，3.4432]

3.4115

特殊原因/可归因原因 special cause /assignable cause

非系统固有的、不可预测的、间歇性的变化来源。参见：共同原因（common cause）。

示例：可归因于系统中的缺陷。在控制图中，表现为超出控制限的点，或控制限内的非随机分布模式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4432]

3.4116

缺陷的特殊原因 special cause of a defect

在某种瞬态情况下所特有的而不是过程所固有的缺陷原因。特殊原因造成过程性能随机变化。

参见：XXX 条 共同原因（common cause(of a defect)）。

[来源：GB/T 11457:2006, 2.1545]

3.4117

特殊功能寄存器 special function register; SFR

用于控制或监控微处理器中功能的寄存器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4433]

3.4118

专业化 specialization

目标文档类型定义的规范，共享为更通用的类型和域开发的通用输出转换和设计规则。

[来源：ISO/IEC/IEEE 26531:2023, 3.1.28]

3.4119

实例化 specialize

通过实例化，使当前类的一个实例同时成为其当前子类的一个（或多个）子类的实例。参见：重新特化；解除特化。

注：特化实例获得不同的（更低级）低级类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4435]

3.4120

专业工程 specialty engineering

分析需要特殊技能来识别需求并评估其对系统生存周期影响的系统的特定特征。

[来源：INCOSE 系统工程手册 第四版 Appendix C]

3.4121

特定地址 specific address

永久分配给设备或存储单元的地址，该地址可直接标识设备或存储单元，无需转换或计算。

参见：相对地址、可重定位地址、符号地址、绝对汇编程序、绝对代码、绝对指令

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4122

特定代码 specific code

所有地址均为绝对地址的代码。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4123

特定配置组 specific profile group

适用于开发关键系统或关键软件且具有非典型情境因素的极小实体（VSE）的配置文件组。

参见：通用配置文件组

[来源：ISO/IEC 29110-1-2:2024, 3.98]

3.4124

特定符号 specific symbol

当流程或数据介质的精确性质或形式已知，且需要描绘实际介质时使用的符号。

[来源：ISO 5807:1985，3.2]

3.4125

斯马特原则 specific, measurable, achievable, relevant, and timely; SMART

具体、可衡量、可实现、相关且及时。

[来源：ISO/IEC/IEEE FDIS 24765，3.4294]

3.4126

规格说明，规约 specification

以完整、精确和可验证的方式识别实体的需求、设计、行为或其他预期特性的信息部件。

[来源：GB/T 11457-2006，2.1548，有修改]

3.4127

规格变更通知 specification change notice; SCN

配置管理中用于对规格说明提出变更、传递变更并记录变更的文件。

参见：配置控制、工程变更、修订通知

[来源：ISO/IEC/IEEE 24765:2017，X.X]

3.4128

规格说明语言 specification language

一种语言，它通常是机器可处理的自然语言和形式语言的组合。用来表示系统或部件的需求、设计、行为或其他特征。例如，设计语言或需求规格说明语言。

参见：编程语言、查询语言

[来源：GB/T 11457-2006，2.1550]

3.4129

规格说明树 specification tree

描写给定的系统的所有规格说明并表示它们相互之间关系的图。

参见：文档树

[来源：GB/T 11457-2006，2.1551]

3.4130

规格说明验证 specification verification

评价系统或部件，以确定软件开发周期中的一个给定阶段的产品是否满足在阶段的开始确立的需求的过程。

参见：验证

[来源：GB/T 11457-2006，2.1552]

3.4131

基于规格说明的测试 specification-based testing

主要基于测试项外部输入和输出的一种测试，通常是依据规格说明而不是源码实现或可执行软件。

注：基于规格说明的测试的同义词为黑盒测试。

[来源：GB/T 38634.1-2020，3.39]

3.4132

基于规格说明的测试 specification-based testing

功能测试 functional testing

黑盒测试 black-box testing

闭盒测试 closed-box testing

主要基于测试项外部输入和输出的一种测试，通常是依据规格说明而不是源码实现或可执行软件。

注：基于规格说明的测试的同义词为黑盒测试。

[来源：GB/T 38634.1-2020，1.314/1.279]

3.4133

峰值测试 spike testing

用于评估系统与软件在短时间内负载大幅度超出通常负载时的性能表现。

注：性能效率测试的一种。

[来源：GB/T 39788-2021，3.3]

3.4134

螺旋模型 spiral model

一种软件开发过程模型，在此种模型中，构成的活动，典型的为需求分析、初步和详细设计、编码、集成和测试是重复地执行的，直至软件完成。

参见：瀑布模型、增量开发、快速原型

[来源：GB/T 11457-2006，2.1553]

3.4135

分离键 split key

包含两个或多个属性的外键，其中至少一个属性是实体主键的组成部分，且至少一个属性不是主键的组成部分。

注：键样式。

[来源：ISO/IEC/IEEE FDIS 24765，3.4448]

3.4136

委托方 sponsor

软件项目的出资方。

[来源：GB/T 36964-2018，3.17]

3.4137

软件产品质量度量参考模型 SPQM-RM

软件产品质量度量参考模型。

[来源：ISO/IEC 25020:2007, 5]

3.4138

电子表格程序 spreadsheet program

显示由行和列排列的单元格表格的程序，其中一个单元格内容的改变可基于单元格间用户定义的关系，导致一个或多个单元格重新计算。

[来源：ISO/IEC 2382:2015，X.X]

3.4139

冲刺 sprint

项目中的一个较短时间区间，在此期间创建一个可用且具备发布潜力的产品增量。

注：在部分组织中，冲刺也被称为迭代。

[来源：ISO/IEC/IEEE FDIS 24765，3.4455]

3.4140

冲刺待办列表 sprint backlog

一组待办事项列表项，例如用户故事，这些项被选定纳入给定的迭代中。

参见：产品待办事项列表。

[来源：ISO/IEC 29110-1-2:2024, 3.100]

3.4141

系统性能规范 SPS

系统性能规范。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.1]

3.4142

结构化查询语言 SQL

结构化查询语言。

[来源：ISO/IEC/IEEE 23026:2023, 3.2]

3.4143

波形线 squiggle

一端连接到箭头标签、另一端连接到箭头线段的短 S 形线条。

注：波形线用于将对象类型集（箭头标签）与对象集（箭头线段）绑定。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4464]

3.4144

系统需求文档 SRD

系统需求文档。

[来源：IEEE 1012-2024, 3.2]

3.4145

系统与软件工程 SSE

系统与软件工程。

[来源：ISO/IEC/IEEE 24641:2023, 3.2]

3.4146

系统/子系统规格说明 SSS

系统 / 子系统规格说明。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.1]

3.4147

稳定性模式 stability schema

对一个对象不会呈现的故障模式的规格说明。

[来源：ISO/IEC 10746-3:2009, 16.2.1.1]

3.4148

稳定阶段 stabilization phase

测量过程中的时间区间，即从实时执行环境（RTE）开始提交任务到被测系统（SUT）达到稳定运行状态的时段。

[来源：ISO/IEC 14756:1999, 4.16]

3.4149

稳定分支 stable branch

不建议进行破坏稳定性变更的分支。

注：该分支用于发布产品的稳定生产版本。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4482]

3.4150

稳定过程 **stable process**

已消除所有过程变异特殊原因且已防止其复发，仅留存过程变异普通原因的过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4483]

3.4151

栈指针 **stack pointer**

存储栈顶地址（最新程序请求的地址）的寄存器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4484]

3.4152

分级表示 **staged representation**

一种结构，其中达成一组过程域的目标即可确立一个成熟度等级，且每个等级都为后续等级奠定基础。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4487]

3.4153

阶段 **stages**

实体生存周期中与其描述或实现状态相关的一段时间。

注：阶段与系统在其生存周期中的主要进展和成就里程碑有关。阶段可以重叠。

[来源：ISO/IEC/IEEE 15288:2023, 3.43]]

3.4154

利益相关方 **stakeholder**

拥有对某一系统或其具备满足自身需求和期望特性的权利、份额、主张或利益的个人或组织。

[来源：ISO/IEC/IEEE 15288:2023, 3.44]。

3.4155

利益相关者参与评估矩阵 **stakeholder engagement assessment matrix**

比较当前和期望的利益相关者参与水平的矩阵。

[来源：项目管理知识体系指南（PMBOK(R) 指南）——第七版]

3.4156

利益相关者参与计划 **stakeholder engagement plan**

项目管理计划的组成部分，确定促进利益相关者有效参与项目或项目群决策和执行所需的战略和行动。

[来源：项目管理知识体系指南（PMBOK® 指南）——第七版]

3.4157

利益相关方权益 **stakeholder equity**

利益相关方在目标系统或目标系统某一部分中所拥有的份额或主张的程度。

[来源：ISO/IEC TR 24766:2009, 3.9]

3.4158

利益相关方需要 **stakeholder need**

在特定使用情境中隐含或明确指出的，对于某个或一组利益相关者实现预期结果所必需的先决条件。

[来源：ISO/IEC 25002:2024, 3.1.27]

3.4159

利益相关者的需求 **Stakeholder needs**

通过与内外部利益相关者沟通，在理解其期望、诉求、需求、价值取向、问题、关注点以及感知到的风险与机遇的基础上所确定的需求。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4160

利益相关者(方)需求和要求定义记录 Stakeholder needs and requirements definition record
与利益相关方需求和要求定义相关的数据、信息或知识的永久性可读形式。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4161

利益相关者(方)需求和要求定义策略 Stakeholder needs and requirements definition strategy

为反映各利益相关方群体之间的共识，以建立一套通用的可接受需求，所需要的方法、计划、资源及具体考量。包括获取利益相关方需求、将其转化为利益相关方需求，并在整个生存周期中对这些需求进行管理的方法。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4162

利益相关方绩效域 stakeholder performance domain

涉及与利益相关方有关的活动和功能的绩效领域。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4163

利益相关方视角 stakeholder perspective

对某个关注实体的思考方式，尤其是当它与关注点相关时。

[来源：GB/T 45630-2025, 3.18]

3.4164

利益相关者(方)要求 Stakeholder requirements

来自各利益相关方、用于管控项目的需求，包括所需的系统能力、功能和 / 或服务、质量标准、系统约束以及成本与进度约束。利益相关方需求可在《利益相关方需求规格说明》(StRS)中予以定义。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4165

利益相关方需求规格说明 stakeholder requirements specification

利益相关方和与外部环境关系的需求的结构化集合。

[来源：GB/T 45802-2025, X.X]

3.4166

利益相关方需求可追溯性 Stakeholder requirements traceability

利益相关者需求的双向可追溯性。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4167

利益相关方满意度 stakeholder satisfaction

产品或系统在指定的使用周境中使用时，利益相关方的要求被满足的程度。

注：产品或系统的用户是利益相关方的类型之一，因此，用户满意度是利益相关方满意度的类型之一。

[来源：GB/T 25000.22-2019, 4.20]

3.4168

独立的，单立的 stand-alone

用于说明无需连接至其他部件即可执行其功能的硬件或软件。

示例：独立式文档处理系统。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4499]

3.4169

标准操作程序 standard operating procedure; SOP

小外型尺寸封装 small outline package

标准操作程序/小外形尺寸封装。

[来源：ISO/IEC/IEEE FDIS 24765 3.4411]

3.4170

标准过程 standard process

用于指导组织内过程实施的一组过程定义。

注：这些过程定义涵盖纳入组织各项目已定义过程的基本过程元素（及其相互关系）。标准过程确保组织内活动的一致性，有助于实现长期稳定性和持续改进。组织的标准过程集描述了构成项目已定义过程的基本过程元素，同时也描述了这些过程元素之间的关系（如排序和接口）。

[来源：ISO/IEC 33001:2015, 3.1.8]

3.4171

标准技术英语 standard technical English; STE

受控语言，包括一套写作规则和一本用于编写技术文档的基础词典。

注：STE 规范由航空航天、安全与国防（ASD）简化技术英语维护小组（STEMG）负责维护。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.28]

3.4172

标准化配置文件 standardized profile

经国际协商一致的协调性文件，描述一个或多个配置文件。

[来源：ISO/IEC 29110-1-2:2024, 3.102]

3.4173

标准实施器 standards enforcer

一种软件工具，它确定指定的开发标准是否得到遵循。标准可以包括模块大小、模块结构、注释的约定、某些语句形式的使用以及文件编制约定。

[来源：GB/T 11457:2006, 2.1564]

3.4174

待机模式 standby mode

休眠模式，即微控制器未处于主动使用状态时节约电能的运行模式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4505]

3.4175

待机模式 standby mode

睡眠模式 sleep mode

微控制器未处于活跃使用状态时可节约功耗的工作模式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4505]

3.4176

备用冗余 standby redundancy

在容错技术中，冗余元素在主元素发生故障前保持未运行状态的使用方式。

参见：主动冗余。

[依据：ISO/IEC/IEEE FDIS 24765, 3.4506]。

3.4177

站立会议 **stand-up meeting**

敏捷开发方法中使用的简短项目每日状态或计划会议。

参见：XXX 条 每日站会（daily stand-up）。

[来源：ISO/IEC/IEEE 26515:2018, 3.14]

3.4178

起始地址 **starting address**

主存储器中计算机程序第一条指令的地址。

参见：起点、汇编起点、装入起点。

条目注释 1：该地址是否与程序的起点一致，取决于第一条指令之前是否存在数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4509]

3.4179

从头到尾 **start-to-finish; SF**

一个活动在另一个活动开始之前不能结束的约束。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4507]

3.4180

从头到头 **start-to-start; SS**

一个活动只有在另一个活动已经启动后才能启动的约束条件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4508]

3.4181

状态 **state**

描述功能、子功能或元素在某一时间点上行为的条件表达式（3.1.6）。

注：状态可用于定义可能随时间变化的功能集或系统特性。

[来源：GB/T 45802-2025 系统与软件工程 生存周期过程 需求工程]

3.4182

状态类 **state class**

表示一组具有共同知识或行为的实际或抽象对象的类。

示例 1：人、地点、事件、概念、物品、物品组合。

示例 2：条目注释 1：状态类表示具有可变状态的实例。状态类的组成实例可以产生和消失，且状态会随时间变化，即其属性值可以发生改变。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4511]

3.4183

状态数据 **state data**

定义测试单元内部状态，并用于建立该状态或与现有状态进行比较的数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4512]

3.4184

状态图 **state diagram**

描述系统或部件可呈现的状态，并展示导致状态间转换或由状态转换引发的事件或情形的图。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4513]

3.4185

状态不变条件 **state invariant condition**

关于约束或关系的陈述，可用于区分所有满足该条件的特定属性状态的子集。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4514]

3.4186

状态机 state machine

规定如下状态的序列的一种行为：某一对象或交互在其生存线内，为对各事件作出响应所经历的（状态）以及该对象的响应和动作。

[来源：GB/T 11457:2006, 2.1571]

3.4187

状态名称 state name

软件执行状态的唯一标识符。

[来源：ISO/IEC 11411:1995, 2.1]

3.4188

状态转移图 state transition diagram

描写系统或部件能假定的状态并表示从一种状态变更为另一个状态的事件或情况的图。

参见：XXX 条 状态图（state diagram）。

[来源：GB/T 11457:2006, 2.1572]

3.4189

状态转移测试 state transition testing

一种基于规格说明的测试用例设计技术，通过在状态模型中执行状态转移实现。

[来源：ISO/IEC/IEEE 29119-4:2021, 3.44]

3.4190

语句覆盖率 statement coverage

测试项中所有可执行语句被测试集覆盖的百分比。

[来源：GB/T 38634.1-2020]

3.4191

目标陈述 statement of objectives; S00

目标陈述。

[来源：ISO/IEC/IEEE 24748-7:2019, 3.2]

3.4192

工作说明书 statement of work; SOW

建议选择：预期成果的陈述及实现成果所需工作的概述。

[来源：ISO/IEC 33202:2024, 3.25]

3.4193

语句测试 statement testing

设计测试用例用于执行测试项中某条语句的测试设计技术。

[来源：GB/T 38634.1-2020]

3.4194

状态转换 StateTransition

3.4195

静态的 static

用于说明无需执行计算机程序即可发生的事件或过程。

参见：动态的。

示例：静态分析、静态绑定。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4520]

3.4196

静态分析 static analysis

基于系统或部件的格式、结构、内容或文档，对其进行评价的过程。

参见：动态分析、审查、走查。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4521]

3.4197

静态分析程序 static analyzer

一种软件工具，它有助于分析计算机程序而无需执行该程序，例如语法检验程序、编译程序、交叉引用表生成程序、标准实施器以及流程图。与动态分析程序（dynamic analyzer）相对。

[来源：GB/T 11457:2006, 2.1578]

3.4198

静态属性 static attribute

硬件标识（HWID）中在产品生存周期内不会改变或在创建时即已确定的元素。

[来源：ISO/IEC 19770-6:2024, 3.1.7]

3.4199

静态绑定 static binding

在计算机程序执行前完成，且执行期间不发生变更的绑定。

参见：动态绑定。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4523]

3.4200

静态断点 static breakpoint

可在编译时设置的断点，例如进入指定例程时的断点。

参见：动态断点、代码断点、数据断点、跛断点、可编程断点、序断点。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4524]

3.4201

静态分类 static classification

其中的对象不可变更类目的泛化的一种语义变化。与动态分类（dynamic classification）相对。

[来源：GB/T 11457:2006, 2.1581]

3.4202

静态转储 static dump

在计算机程序执行前或执行后生成的转储。

参见：动态转储、变更转储、内存转储、事后转储、选择性转储、快照转储。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4525]

3.4203

静态差错 static error

与输入的时变特性无关的错误。

参见：动态错误。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4526]

3.4204

静态模型 static model

描述一组相互关联的类（或主题域）及其关系和职责的模型。

参见：动态模型。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4527]

3.4205

静态产品 static product

用于评审的非执行状态的系统或软件产品。

[来源：GB/T 25000.45-2018]

3.4206

静态程序分析 static program analysis

在不以目标（二进制）格式执行代码的情况下分析软件代码属性的形式化方法的子领域。

[来源：ISO/IEC 23643:2020, 3.24]

3.4207

静态随机存取存储器 static random access memory; SRAM

无刷新过程的随机存取存储器，只要通电就能保持数据。

注：基于具有双稳定状态的电路架构。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4530]

3.4208

静态随机存取存储器 static random access memory (SRAM)

无需刷新过程的随机存取存储器，通电时可保持数据。

注：基于双稳态电路架构。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4530]

3.4209

静态模式 static schema

在某一时间点，受任意不变模式约束的一个或多个信息对象状态的规格说明。

注：因此，静态模式是某一时间点一个或多个信息对象类型的规格说明，这些类型是不变模式中所规定类型的子类型。

[来源：ISO/IEC 10746-3:2009, 6.1.2]

3.4210

静态测试 static testing

在不运行代码的情况下，通过一组质量准则或其他准则对测试项进行检查的测试。

示例：评审、静态分析。

[来源：GB/T 38634.1-2020]

3.4211

统计过程控制 statistical process control

基于统计的方法对过程及其绩效指标进行分析，以识别过程绩效中变异的普遍原因和特殊原因，并确保过程绩效保持在限定范围内。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4212

统计抽样 statistical sampling

抽取目标总体中的部分个体用于检查。

[来源：ISO/IEC/IEEE 24765:2017, 3.3981]

3.4213

统计测试模型 statistical test model

一种模型，它把程序故障与输入数据集（或多个数据集）联系起来。模型也给出了这些故障将引起程序失效的概率。

[来源：GB/T 11457:2006, 2.1584]

3.4214

统计管理过程 statistically managed process

通过基于统计的技术进行管理的过程，其中对过程进行分析，识别过程变异的特异性原因，并将性能控制在明确定义的限值内。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4215

状态码 status code

[来源：ISO/IEC 10746-3:2009, 信息技术 —— 开放分布式处理 —— 参考模型：架构, 6.1.2]

3.4216

例会 status meeting

定期举行的会议，旨在交流和分析项目当前进度及其绩效的相关信息。

参见：Scrum 会议。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4217

状态报告 status report

项目当前状态的报告。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4218

指导委员会 steering committee

由高级利益相关者的咨询机构，为项目团队提供指导和支持，并在项目团队权限之外做出决策。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4219

步骤 step

程序中的一个元素（编号列表项），用于告知用户执行一个或多个操作。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.48]

3.4220

分步说明 step-by-step instructions

由一系列步骤组成的程序，用于安全、有效、高效地使用被支持产品。

[来源：IEC/IEEE 82079-1:2019, 3.38]

3.4221

步进操作 step-by-step operation

一种调试技术，在其中，单计算机指令或指令的一部分执行以响应外部信号。

参见：单步操作（single-step operation）。

[来源：GB/T 11457:2006, 2.1586]

3.4222

逐步细化 stepwise refinement

一种软件开发技术，首先，广泛地定义数据和处理步骤，然后逐步详细。

参见：以数据结构为中心的设计、输入-处理-输出、模块分解、面向对象的设计、快速原型、结构设计、事务分析、转换分析

[来源：GB/T 11457-2006, 2.1587]

3.4223

衍型 stereotype

一种扩展元模型语义的建模元素的新类型。衍型必须基于元模型中现有的某些类型或类。衍型可扩展已存在的类型和类的语义，但不可扩展其结构。某些衍型在 UML 中是预定义的，另一些可由用户定义。衍型是 UML 中三种可扩展性机制之一。

参见：约束、加标值

[来源：GB/T 11457-2006, 2.1588]

3.4224

激励 stimulus

任何导致单元在其行为模式库中呈现特定行为模式的因素；即引发或被视为引发反应的因素。

注：刺激可为影响单元的外部控制交互事件（外部刺激），亦可为内部控制事件（内部刺激）。

[来源：ISO/IEC/IEEE 24765i:2020, X.X]

3.4225

库存量单位 stock keeping unit; SKU

特定产品的标识，通常为字母数字，用于库存和软件授权跟踪。

注：通常与用于销售目的的独特产品相关联，如软件授权。它可以唯一对应特定的软件产品，或代表具有特定条款和条件的软件包，例如标识对象为完整产品、升级产品，或是现有产品的维护服务。

[来源：ISO/IEC 19770-5:2015, 3.48]

3.4226

停止 stop

终止计算机程序的执行。

参见：暂停。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4546]

3.4227

存储器 storage

可存入数据、保存数据并检索数据的功能单元。

[来源：ISO/IEC 2382:2015, 词汇]

3.4228

存储分配 storage allocation

计算机资源分配的组成部分，包括为特定作业分配存储区域，并执行相关过程（如在主存储器与辅助存储器之间传输数据）以支持所做的分配。

参见：缓冲器、邻接分配、循环搜索、内存紧缩、覆盖、调页、虚拟存储器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4548]

3.4229

存储断点 storage breakpoint

当访问指定数据项时触发的断点。另请参阅代码断点、动态断点、尾声断点、可编程断点、序言断点、静态断点。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1138]

3.4230

存储容量 storage capacity

给定存储设备可容纳的最大项目数，通常以字或字节为单位计量。

参见：通道容量、内存容量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4549]

3.4231

存储效率 storage efficiency

系统或部件以最少可用存储消耗执行指定功能的程度。

参见：执行效率。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4550]

3.4232

存储 storage

存储设备 storage device

可以存放数据、存储数据并从中检索数据的功能单元。

同义词：存储设备。

3.4233 [[来源：ISO/IEC 2382:2015, X.X]

存储 store

将数据存入或保存在存储设备中。

2. 将计算机指令或数据从寄存器复制到内部存储器，或从内部存储器复制到外部存储器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4551]

3.4234

故事地图 story map

给定产品所需的所有特性和功能的可视化模型，旨在让团队对其正在构建的内容及其原因有一个全面的了解。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4235

故事点 story point

相对于项目团队认为的典型用户故事而言，开发用户故事所需工作量的相对衡量标准。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4553]

3.4236

故事映射 story-mapping

制作待办事项列表中部分内容的二维表示图，作为用户直观辅助工具，以便在待办事项列表中逻辑地切割、分组和排列故事，从而概述所有用户故事之间的关系。

参见：故事地图。

[来源：ISO/IEC TR 7052:2023, 3.43]

3.4237

软件测试计划 STP

软件测试计划。

[来源：ISO/IEC/IEEE 24748-8:2019, part 8, 3.2]

3.4238

直线代码 straight-line code

无循环结构的计算机指令序列。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4556]

3.4239

直线编码 straight-line coding

一种编程技术, 通过显式完整列出循环执行时涉及的所有指令, 从而避免使用循环结构。

参见: 展开。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4557]

3.4240

战略资产管理计划 strategic asset management plan; SAMP

规定如何将组织目标转化为资产管理目标、制定资产管理计划的方法, 以及资产管理体系在支撑实现资产管理目标中所起作用的成文信息。

[来源: ISO/IEC 19770-1:2017, 3.53]

3.4241

战略性 IT 资产管理计划 strategic IT asset management plan

规定组织目标如何转化为 IT 资产管理目标、制定 IT 资产管理计划的方法以及 IT 资产管理体系在支持实现 IT 资产管理目标中的作用的文件化信息。

同义词: 战略信息技术资产管理计划。

[来源: ISO/IEC 19770-1:2017, 3.54]

3.4242

战略信息技术资产管理计划 strategic IT asset management plan.

规定如何将组织目标转化为信息技术资产管理目标、制定信息技术资产管理计划的方法, 以及信息技术资产管理体系在支持实现信息技术资产管理目标中作用的文件化信息。

[来源: ISO/IEC 19770-1:2017, part 1, 3.54]

3.4243

战略计划 strategic plan

高级别的文件, 阐释了组织的愿景和使命, 以及为实现这一使命和愿景所采取的方法途径, 包含在本文件所覆盖的周期内需要达成的具体指标与目标。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4244

战略制品 strategy artifacts

在项目开始前或开始时创建的, 涉及项目战略、业务或高层信息的文件。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4245

战略文件 Strategy documents

所有适用生存周期过程的策略, 包括业务或任务分析策略、利益相关方需求及需求定义策略、系统需求定义策略、架构定义策略、设计定义策略、系统分析策略、实施策略、集成策略、验证策略、移交策略、确认策略、运行策略、维护策略、处置策略、项目评估与控制策略、决策管理策略、风险管理策略、配置管理策略、信息管理策略、测量策略、采购策略及供应策略。

[来源: INCOSE 系统工程手册 第四版, 附录 E]

3.4246

分层语言 stratified language

不能作为自身元语言的语言。

示例: FORTRAN、COBOL。

参见：非分层语言

[来源：ISO/IEC/IEEE FDIS 24765, 3.4563]

3.4247

流接口 **stream interface**

所有交互均以流的形式进行的接口。

[来源：ISO/IEC 10746-3:2009 7.1.8]

3.4248

流接口签名 **stream interface signature**

流接口对应的接口特征。

注：流接口包含一组有限的动作模板，每个流类型对应一个动作模板。每个流的动作模板包含流的名称、流的信息类型，以及相对于实例化该模板的对象的流因果关系指示（即生产者或消费者，二者仅居其一）。短语“与 X 互补的接口特征”（其中 X 本身为接口特征）指的是除因果关系与 X 相反外，其他所有方面均与 X 相同的接口特征。许多接口定义语言（IDL）仅捕获特征的动作模板，并依赖 IDL 的使用上下文确定要应用的因果关系。

[来源：ISO/IEC 10746-3:2009]

3.4249

优势 **strengths**

劣势 **weaknesses**

机会和威胁 **opportunities, and threats; SWOT**

优势、劣势、机会和威胁。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4642]

3.4250

优势、劣势、机遇和威胁分析 **strengths, weaknesses, opportunities, and threats analysis; SWOT analysis**

对组织、项目或备选方案的优势、劣势、机遇和威胁的分析。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4251

压力测试 **stress testing**

性能效率测试的一种类型。用于评价测试项在高于预期或指定容量负载需求，或低于最少需求资源的条件下的行为。

[来源：GB/T 38634.1-2020, 3.43]

3.4252

强类型法 **strong typing**

某些编程语言的特性要求声明每个数据项的类型，禁止将运算符应用于不适当的数据类型，并防止不兼容类型的数据项进行交互。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4568]

3.4253

结构性特征 **structural feature**

某一模型元素的一种静态特征，例如一个属性。

[来源：GB/T 11457-2006, 2.1602]

3.4254

结构性模型方面 **structural model aspect**

侧重于系统中各对象的结构的一种模型方面，它包括对象的类型、类、联系、属性和操作。

[来源: GB/T 11457—2006, 2.1603]

3.4255

结构测试 structural testing

一种性能效率测试类型,用于评估测试项在负载超过预期或规定容量要求、或资源可用性低于规定最低要求条件下的行为。

参见: 边界值。

[来源: ISO/IEC/IEEE 29119-1:2022, part 1, 3.79]

3.4256

结构测试 structural testing; glass-box testing; white-box testing

动态测试,其测试用例源自对被测试项结构的检查。

参见: 结构测试示例: 分支测试、判定测试、语句测试。

注: 基于结构的测试不仅限于组件级别,还可用于所有测试层级,例如作为系统测试组成部分的菜单项覆盖测试。

[来源: ISO/IEC/IEEE 29119-1:2022, 3.80]

3.4257

结构测试 structural testing; white-box testing

基于结构的测试 structure-based testing

白盒测试 glass-box testing

由测试项结构导出测试用例的动态测试。

注1: 基于结构的测试并不局限于在组件级别的使用,其可以被用于所有级别,如系统测试中菜单项覆盖。

注2: 技术包括分支测试、判定测试和语句测试。

注3: 基于结构的测试的同义词包括结构测试,白盒测试。

[来源: GB/T 38634.1-2020, 3.45]

3.4258

结构图 structure chart

标识系统或计算机程序中的模块、活动或其他实体,并展示较大或较通用的实体如何分解为较小、较具体实体的图。

参见: 调用图、层次图。

条目注释1: 其结果不一定与调用图所示内容相同。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4570]

3.4259

结构图 structure chart

层次结构图 hierarchy chart

程序结构图 program structure chart

用于标识系统或计算机程序中的模块、活动或其他实体,并展示较大或更通用的实体如何分解为更小、更具体的实体的图表。

参见: 调用图、层次结构图。

注: 其结果未必与调用图所示相同。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4570]

3.4260

结构相搅 structure clash

在软件设计中,某一模块需处理两个或多个数据结构不兼容的数据集的情形。

参见: 以数据结构为中心的设计、次序冲突。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4571]

3.4261

结构模型 structure model

从空间域上定义系统的组成。

注：它用于描述系统所包括的所有构件的集合以及构件之间的关系，但不包括时间和同步信息。

[来源: GB/T 36457-2018, 3.8]

3.4262

基于结构的测试 structure-based testing

由测试项结构导出测试用例的动态测试。

注1：基于结构的测试并不局限于在组件级别的使用，其可以被用于所有级别，如系统测试中菜单项覆盖。

注2：技术包括分支测试、判定测试和语句测试。

注3：基于结构的测试的同义词包括结构测试，白盒测试。

[来源: GB/T 38634.1-2020, 第1部分: 概念和定义]

3.4263

结构化编写 structured authoring

根据预定义且强制实施的内容元素组织结构来开发内容元素，包括指定模板中的元数据。

注：在结构化写作中，内容元素根据其所含内容的性质进行标记。结构化写作还允许使用准语义标记（如 Heading1 或 NestedList）来表明内容元素的层级位置和功能。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4578]

3.4264

结构化设计 structured design

一种规范化的软件设计方法，遵循基于模块化、自顶向下设计以及数据、系统结构和处理步骤逐步细化等原则的既定规则。

参见：以数据结构为中心的设计、输入 - 处理 - 输出、模块分解、面向对象设计、快速原型法。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4573]

3.4265

结构化程序 structured program

由一组基本控制结构构成的计算机程序，每个控制结构均具有单一入口和单一出口。

参见：结构化设计。

条目注释 1：该组控制结构通常包括：两条或多条指令的序列、从两条或多条指令序列中进行条件选择、指令序列的重复执行。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4574]

3.4266

结构化程序设计 structured programming

包含结构化设计，并最终产出结构化程序的一种软件开发技术。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4575]

3.4267

结构化程序设计语言 structured programming language

提供结构化程序构造（即单入口 - 单出口序列、分支和循环）并便于开发结构化程序的程序设计语言。

参见：块结构语言。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4576]

3.4268

结构化走查 structured walkthrough

由合格人员对系统的需求、设计、实现或其任意部分开展的系统性审查。

3.4269 [[来源: ISO/IEC 2382:2015, X. X]

结构化写作 structured writing; structured authoring

根据预定义且强制执行的内容元素组织结构, 开发内容元素, 包括指定模板中的元数据。

注: 在结构化写作中, 内容元素根据其所包含内容的性质进行标记。结构化写作还允许使用准语义标记, 如标题或嵌套列表, 以指示内容元素的层级位置和功能。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.29]

3.4270

桩[模块] stub

软件模块的框架性或专用性实现, 用于开发或测试调用该模块或依赖该模块的其他模块。

为支持软件研发和测试而构建的计算机程序和数据文件, 不打算纳入最终产品。

[来源: ISO/IEC/IEEE 24748-6:2023, 3.1.7]

3.4271

样式 style

一组特定于语言的编辑约定, 涵盖语法、术语、标点、大小写、用法、词汇选择、结构化编写元素和文档中的图像。

[来源: ISO/IEC/IEEE 26513:2017, 3.37]

3.4272

子活动 subactivity

本身是活动且满足以下条件的活动的子图: 对于父活动中的任何一对 分叉-连接 操作, 如果这些操作中的一个包含在子图中, 则两者都必须包含在子图中。

[来源: ISO/IEC 10746-2:2009, 13.1.8]

3.4273

子活动状态 subactivity state

在活动图中, 一种表示执行具有持续期的非原子步骤的序列的状态。

[来源: GB/T 11457-2006, 2.1612]

3.4274

子类 subclass

一个或多个超类的特化。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.186]

3.4275

子类簇 subclass cluster

同一通用实体的一个或多个互斥特化的集合。

注: 当超类的一个实例只能是该集合中子类之一的实例, 且每个子类的实例都是超类的实例时, 就存在一个簇。同义词: 类别簇。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.187]

3.4276

子类簇 subclass cluster

类别簇 category cluster

一个或多个泛化结构的集合, 其中子类共享相同的超类, 且超类的实例最多只能是一个子类的实例。

注：当超类的实例只能是集合中某一个子类的实例，且每个子类的实例都是超类的实例时，就存在聚类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.187]

3.4277

子类职能 subclass responsibility

表示类的某个属性在其子类中被重写的标识。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.188]

3.4278

子域 subdomain

给定域的子集。

[来源：ISO/IEC 10746-2:2009, 10.4]

3.4279

主体 subject

与系统交互（无论自愿或非自愿，无论是否知情）的人，或其个人数据已经、正在或将要被获取以用于与系统交互或训练系统的人。

参见：数据主体。

[来源：IEEE 7014-2024, 3.1]

3.4280

主题领域 subject area

元对象实例定义的相关集合。

注：主题领域用于定义范围内的兴趣领域。主题区域重叠以确保整体元模型的集成，但工具只需要使用与要导出或导入的数据相关的那些主题区域。

[来源：ISO/IEC 15474-1:2002 Part 1: 4.2]

3.4281

主题域 subject domain

关注或专长的领域。

注：主题域的职责是对一组现有或潜在命名类的职责的聚合；主题域也可包含其他主题域；主题域封装某一视图的细节内容。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.189]

3.4282

主题域职能 subject domain responsibility

分析人员通过提出问题“一般而言，该主题域中的实例需要具备哪些能力或掌握哪些信息？”所提炼的广义概念。

注：某一主题域内的各类与子主题域共同提供构成该领域的知识、行为和规则，这些内容统称该主题域的职能。在分析初期，主题域职能无需区分其所属子域或类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.190]

3.4283

目标软件 subject software

计算机系统工具或计算机辅助软件工程（CASE）工具中，为这类应用程序开发描述性信息的、已存在或待开发的计算（软件）应用程序。

[来源：IEEE 1175.3-2004, X.X]

3.4284

目标工具 subject tool

特定计算系统工具或 CASE 工具，它是互连和工具内容描述的重点。

[来源：IEEE 1175.2-2006, 3.14]。

3.4285

目标单元 subject unit

作为行为建模主题的单位。

注：主题单元与构成其行为被征求和展示的环境。

[来源：IEEE 1175.4-2008, 3.28]

3.4286

领域专家 subject-matter expert; SME

负责为用户提供技术信息或检查信息草案技术准确性的相关人员。

[来源：ISO/IEC/IEEE 26514:2022, 3.1.50]

3.4287

子机[器]状态 submachine state

如下状态机中的一种状态：等效于某一组合状态，但其内容由另一状态机来描述。

[来源：GB/T 11457-2006, 2.1616]

3.4288

子网络 subnetwork

项目进度网络图的细分部分（片段），通常代表一个子项目或一个工作包。

注：常用于说明或分析一些潜在的或建议的进度条件，例如优先进度逻辑的变更或项目范围的变更。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4594]

3.4289

子包 subpackage

一种包含在另一包中的包。

[来源：GB/T 11457-2006, 2.1617]

3.4290

子路径 sub-path

作为较大路径一部分的路径。

[来源：ISO/IEC/IEEE 29119-4:2021 Part 4, 3.47] 同义词：subpth

3.4291

子阶段 subphase

阶段的细分。

[来源：IEEE_CS_SE_Vocabulary-2025]

3.4292

子程序 subprogram

计算机程序中可独立编译、可执行的组成组件。

注：术语「例程」「子程序」和「子例程」在不同编程语言中的定义和使用方式存在差异。参见：协程、主程序、例程、子例程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4595]

3.4293

子项目 subproject

将项目拆分为更易于管理的组成部分后，所形成的整体项目的组成部分。

[来源：ISO/IEC/IEEE 24765-2025 fdis, 3.4596]

3.4294

子例程 `subroutine`

执行后将控制权返还至调用它的程序或子程序的例程。

参见：协程、闭型子例程、开型子例程。

注：术语「例程」「子程序」和「子例程」在不同编程语言中的定义和使用方式存在差异。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4597]

3.4295

子例程跟踪 `subroutine trace`

记录计算机程序执行期间执行的所有或选定的子程序或函数调用，以及可选的，传递给每个子程序或函数并由每个子程序或函数返回的参数值。

计算机程序执行过程中，所有或选定子程序、函数调用的记录；还可选择性地记录各子程序、函数的传入参数值与返回参数值。

参见：执行跟踪、追溯跟踪、子程序跟踪、符号跟踪、变量跟踪。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4598]

[来源：同义词：调用跟踪]

3.4296

子程序跟踪 `subroutine trace`调用跟踪 `call trace`

计算机程序执行期间所有或选定子程序/函数调用的记录，可选择性包含传递给每个子程序/函数的参数值及其返回值。

同义词：调用追踪。

参见：执行追踪、回溯追踪、子程序追踪、符号追踪、变量追踪。

[来源：IEEE_CS_SE_Vocabulary-2025]

3.4297

订阅式许可证 `subscription-based license`

限时授权许可。

注1：不是永久性的许可证并需要更新才能保持有效。

同义词：基于期限的许可，基于服务的许可。

[来源：ISO/IEC 19770-3:2016 Part3, 3.1.39]

3.4298

基于订阅的许可 `subscription-based license`基于服务的许可 `service-based license`基于期限的许可 `term-based license`

一种针对有时限权利的许可授权。

注1：该许可并非永久授权，需通过续期维持效力。

同义词：基于期限的许可、基于服务的许可。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4599]

3.4299

订阅式许可证 `subscription-based license`服务式许可证 `service-based license`期限式许可证 `term-based license`

在限定时间内有效的授权许可证。

[来源: ISO/IEC 19770-3:2016, 3.1.39]

3.4300

子集选取 **subsetting**

定义或创建数据子集, 该子集仅包含数据源中的部分而非全部数据。

[来源: IEEE 982:2024, 3.1]

3.4301

子状态 **substate**

作为某一组合状态一部分的状态。

参见: 并发子状态、互斥子状态

[来源: GB/T 11457-2006, 2.1621]

3.4302

可替代性 **substitutability**

因子类的每个实例均为超类的实例, 故在超类实例适用的任意语境中, 子类实例均具备适用性的设计原则。

注: 向某一实例发送的任一请求, 无论接收方是子类实例还是超类实例, 均会返回符合要求的响应。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4601]

3.4303

次级子特征 **sub-subcharacteristic**

质量模型用户建立的质量子特性的细分, 以更细致地表示目标实体的质量属性。

[来源: ISO/IEC 25002:2024, 3.28]

3.4304

子系统 **subsystem**

较大系统中的二级或从属系统。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4602]

3.4305

子类型 **subtype**

数据类型的子集, 通过约束数据类型的可能值集合获得。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.4306

成功标准 **success criteria**

过程实例在完成时要满足的一组条件。

注: 过程产生的信息项和人工产物实例必须满足成功标准。成功标准是根据相应生存周期过程的结果、过程实例所贡献的系统元素的要求以及其他过程实例中的决策产生的要求和约束来建立的。

[来源: ISO/IEC 30103:2015, 3.9]

3.4307

后续活动 **successor activity**

在进度计划中, 逻辑上位于另一活动之后的从属活动。

[来源: IEEE_CS_SE_Vocabulary-2025]

3.4308

适合性 **suitability**

在使用产品时, 其行为和(或)结果满足规定质量要求的程度。

[来源: ISO/IEC 25019:2023, 3.2.1.3]

3.4309

总结活动 summary activity

作为单个活动来展示的，一组相关的进度活动的集合。

同义词：汇总活动 参见：子项目、子网络。

[来源：IEEE_CS_SE_Vocabulary-2025]

3.4310

总结性评价 summative evaluation

为总结体现评价客体的优点或价值而设计的评价。

注1：评价结果能用于针对是否宜保留、更改或删除评价客体形成推荐性意见。

注2：设计一种兼具形成性和总结性的评价方法是可能的。

注3：采用总结性测试方法来实施总结性评价。

[来源：GB/T 25000.22-2019]

3.4311

沉没成本 sunk cost

无法通过未来行动收回的成本。

注：沉没成本会产生心理影响，但在商业决策中无关紧要。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4312

超类 superclass

其实例专门化为一个或多个子类的类。

[来源：ISO/IEC/IEEE 24765n: 2025, 3.1.193]

3.4313

超级计算机 supercomputer

在特定时间内具有最高处理速度的计算机类别。

3.4314 [[来源：ISO/IEC 2382:2015, X.X]

超类型 supertype

类型 B 和类型 A 之间的关系，其中满足 A 的每个 <X> 也满足 B。

参见：通用实体、子类型、超类。

[来源：ISO/IEC 10746-2:2009, 9.11]

3.4315

监督学习 supervised learning

基于带标签的输入输出示例对，学习将输入映射到输出的函数的任务。

参见：无监督学习。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4613]

3.4316

监督程序 supervisor

计算机程序，通常作为操作系统的一部分，用于控制其他计算机程序的执行并调节计算机系统的工作流程。参见：监督状态、内核。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4615]

3.4317

监督状态 supervisor state

(1) 在计算机系统运行中，监控程序正在执行的状态。

参见：问题状态。

注 1：该状态通常具有比应用程序更高的优先级，并会阻止应用程序的执行。

同义词：执行状态，特权状态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4614]

3.4318

监督状态 supervisor state

执行状态 executive state

主控状态 master state

特权状态 privileged state

在计算机系统运行中，一种监督程序正在执行的状态。

注 1：此状态通常具有比应用程序更高的优先级，并阻止应用程序的执行。

同义词：执行状态、特权状态。

另请参阅：问题状态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4614]

3.4319

监督程序 supervisory program

控制程序 control program

执行 executive

执行程序 executive program

主管 supervisor

计算机程序，通常作为操作系统的一部分，用于控制其他计算机程序的执行并调节计算机系统的工作流程。同义词：控制程序、执行程序、管理程序、监督程序。

参见：监控状态、内核。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4615]

3.4320

补充授权实体 supplemental Ent

补充授权模式 supplemental entitlement schema

补充授权：具有 Supplemental 类型的 Ent。

注：补充授权提供关于主授权的扩展信息，并通过 <linkedToPrimaryEntId> 属性链接到主授权。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4616]

3.4321

补充权利项 supplemental Ent; supplemental entitlement schema

<entType>（权利项类型）取值为 Supplemental（补充型）的权利项（Ent）。

注：补充权利项用于提供关于主权利项（primary Ent）的扩展信息，并通过 <linkedToPrimaryEntId>（关联主权权利项标识）属性与主权利项建立关联。

[来源：ISO/IEC 19770-3:2016 Part 3: 3.1.40]

3.4322

补充信息结构 supplemental information structure

具有从属关系且能扩展另一信息结构内容的信息结构。

同义词：补充性信息结构。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4617]

3.4323

补充运行 supplementary run

测量程序的时间间隔，从测量结果达到要求的统计显著性到评级间隔内提交的所有任务完成的时间。

[来源：ISO/IEC 14756:1999, 4.17]

3.4324

供应系统 Supplied system

系统或系统要素（产品或服务）按照供应协议的交付条件从供方交付给采购方。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4325

供方 supplier

与需方订立协议，为其提供产品或服务的组织或个人。

[来源：ISO/IEC/IEEE 12207:2017 3.1.60；ISO/IEC/IEEE 15288:2023，3.45]

3.4326

Supplier management procedure

3.4327

供应商 supplier

承包商 contractor

生产商 producer

卖方 vendor

提供产品的个人或组织。

注：采购方和供应商有时属于同一组织。应用管理组织可能拥有内部或外部供应商。供应商可以是另一家应用管理组织，也可以是 IT 基础设施管理组织或咨询公司。同义词：承包商、分销商、生产商、零售商、销售商、SUP、供应商。

[来源：IEC/IEEE 82079-1:2019, 3.39]

3.4328

供应协议 Supply agreement

对项目组织与采购方之间关系及承诺的理解。该协议形式多样，既可以是正式合同，也可以是非正式的组织间工作指令。正式协议通常包含条款与条件。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4329

供应付款 Supply payment

对所提供系统的付款或其他补偿。包括收据和确认。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4330

供应记录 Supply record

与供应相关的永久、可读形式的数据、信息或知识。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4331

供应报告 Supply report

为相关方编制的报告，用于通报供应活动的状态、进展与成果。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4332

供应响应 Supply response

组织对供应请求的响应。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4333

供应策略 Supply strategy

确定供管理层考虑的候选项目所需的方法、时间表、资源和具体考虑因素。还可能包括用于确定供应限制的输入。还应涵盖潜在采购方的识别。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4334

支持活动组 support activity group

为确保项目成功完成所必需的活动组，但。

其由辅助活动组成，而非直接面向开发工作的活动。

[来源：ISO/IEC/IEEE 24765:2017 系统与软件工程——术语]

3.4335

支持经理 support manager

微型实体中负责生产支持的角色，承担客户关系管理，并监督生产环境内的变更管理工作。

[来源：ISO/IEC 29110-5-1-2:2025, 3.36]

3.4336

支持手册 support manual

提供在整个生存周期内为运行中的系统或组件提供维修和维护所需信息的文件。

参见：诊断手册、安装手册、维护手册、操作手册、程序员手册、用户手册。

注：通常会描述构成系统或组件的硬件和软件，以及维修、修理或重新编程的程序。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4623]

3.4337

支持软件 support software

对其他软件的开发或维护提供辅助的软件。

参见：应用软件、系统软件。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4338

支持人员小时 support staff-hour

工作人员投入的工作小时数，该人员不直接定义或创建软件产品，但协助从事此类工作的人员。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4339

支持阶段 support stage

为系统提供服务、使其能够持续运行的阶段。可提出修改方案，以解决可保障性问题和降低运行成本或延长系统使用寿命。此类变更需经过系统工程（SE）评估，避免系统在运行过程中出现能力损失。

[来源：INCOSE 系统工程手册 第四版，3.3.5]

3.4340

支持向量机 support vector machine; SVM

支持向量机。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4636]

3.4341

可支持性 supportability

为建立、维护和修订在役系统持续使用所需的条件和资源而采取的支持行动的频次、范围和一致性。

注：作为可信性的一个定量组成部分，可支持性可以表示为一个需求参数（阈值）、一个概率（预测或估计）或在测试或实际运行中获得的测量值（观测值）。

[来源：IEEE 982:2024, 3.1]

3.4342

支持的产品 supported product

与使用信息相关的产品。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4627]

3.4343

监督 surveillance

为保持合格声明的有效性，而系统地重复开展合格评定活动。

[来源：ISO/IEC 29110-1-2:2024, Part1-2, 3.10]

3.4344

生存性 survivability

产品在遭受攻击的情况下，仍能及时提供关键服务的能力。

[来源：ISO/IEC/IEEE 247651:2024]

3.4345

挂起准则 suspension criteria

用于（暂时）停止全部或部分测试活动的标准。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.81]

3.4346

保障 sustainment

为确保产品或服务保持可运行状态而开展的活动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4631]

3.4347

交换 swap

交换两个存储区域的内容，通常是一个主存储区域和一个辅助存储区域。

参见：. 卷入，卷出。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4639]

3.4348

泳道 swimlane

在活动图上，一种用于组织动作职责的分割。在典型情形下，泳道对应于业务模型中的组织单元。

参见：2.1113 条 分区（partition）。

[来源：GB/T 11457-2006, 2.1634]

3.4349

象征 symbol

在特定语境中具有特定含义的概念的图形表示。

3.4350 [[来源：ISO/IEC 2382:2015, X.X]

符号表 symbol table

表示程序符号及其对应的地址、值和其他属性的表。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4646]

3.4351

符号地址 `symbolic address`

以名称或标签表示的地址，必须转换为要访问的设备或存储位置的绝对地址。

参见: 绝对地址。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4647]

3.4352

符号执行 `symbolic execution`

软件分析技术，其中使用符号（例如变量名称）而不是输入数据的实际值来模拟程序执行，程序输出表示为涉及这些符号的逻辑或数学表达式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4648]

3.4353

符号语言 `symbolic language`

以人类方便的符号而不是机器语言来表达操作和地址的编程语言。

参见: 机器语言。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4649]

3.4354

符号踪迹 `symbolic trace`

当计算机程序用符号而不是用输入数据的实际值运行时，遇到的源语句或分支出口的记录。

参见: 执行跟踪、追溯跟踪、子程序跟踪、变量跟踪。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4650]

3.4355

对称网 `symmetric net`

高级佩特里网，其类型有限且仅允许内置操作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4651]

3.4356

同步状态 `synch state`

在状态机中，用于对其中的并发区域同步化的一种顶点。

[来源: GB/T 11457-2006, 2.1640]

3.4357

同步 `synchronize`

将父分支中所做的更改拉入其（发展中的）子（例如，功能）分支。

将父分支的变更拉取至动态迭代的子分支（如功能分支）。

使用其相应分支中文件的当前版本更新视图。

[来源: ISO/IEC/IEEE 24765:2017, X.X]

3.4358

同步的 `synchronous`

与两个或多个依赖于特定事件发生的过程有关，例如共同时间。

3.4359 [[来源: ISO/IEC 2382:2015, X.X]

同步动态随机存取存储器 `synchronous dynamic random access memory; SDRAM`

一种动态随机存取存储器（DRAM），其存储器访问由与处理器存储器总线时钟同步的时钟驱动。

注: SDRAM 可同时访问不同存储体中的数据片段。

[来源: ISO/IEC/IEEE 24765c:2014; ISO/IEC/IEEE FDIS 24765, 3.4654]

3.4360

同步图形随机存取存储器 **synchronous graphics random access memory (SGRAM)**; SGRAM
为电脑显卡设计的 SDRAM。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4655]

3.4361

同步消息通信 **synchronous message communication**
生产者任务向消费者任务发送消息并等待确认的通信形式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4656]

3.4362

带应答的同步消息通信 **synchronous message communication with reply**
生产者（或客户端）任务向消费者（或服务器）任务发送消息并等待应答的通信形式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4657]

3.4363

带有回复的同步消息通信 **synchronous message communication with reply**
回复紧密耦合的消息通信 **tightly coupled message communication with reply**
生产者（或客户端）任务向消费者（或服务器）任务发送消息并等待回复的通信形式。
同义词: 带回复的紧耦合消息通信。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4657]

3.4364

无应答的同步消息通信 **synchronous message communication without reply**
生产者任务向消费者任务发送消息并等待消费者接受消息的一种通信形式。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4658]

3.4365

无回复的同步消息通信 **synchronous message communication without reply**
无回复的紧耦合消息通信 **tightly coupled message communication without reply**
生产者任务向消费者任务发送消息并等待消费者接受该消息的通信形式。
同义词: 无回复的紧耦合消息通信。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4658]

3.4366

同步消息通信 **synchronous message communication**
紧耦合消息通信 **tightly coupled message communication**
生产者任务向消费者任务发送消息并等待确认的通信形式。

同义词: 紧耦合消息通信。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4656]

3.4367

同步请求 **synchronous request**
客户端暂停等待请求完成的请求。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4659]

3.4368

语法一致 **syntactic agreement**

3.4369

语法错误 syntactic error

违反为语言定义的结构或语法规则。

参见：语义错误。

例如：FORTRAN 中，使用 $B+C=A$ 说明而不是正确的 $A=B+C$ 。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4660]

[来源：同义词：语法错误]

3.4370

语法错误 syntactic error**句法差错** syntax error违反为语言定义的结构或文法规则。例如，在 FORTRAN 中用语句 $B+C=A$ ，而不是正确的 $A=B+C$ 。

相对：语义错误。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4660]

3.4371

语法 syntax

(1) 用于规定语言中的符号如何组合以构成词、短语、表达式及其他合法构造的结构规则或语法规则。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

(2) 对 CDIF 传输中信息格式的定义。

参见：语义。

[来源：ISO/IEC 15474-1:2002 Part 1, 4.2]

3.4372

句法测试 syntax testing

一种基于规格说明的测试用例设计技术，通过执行测试项输入的形式化定义中的元素实现。

注：巴科斯-诺尔范式通常用于定义测试项输入的语法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4662]

3.4373

综合 synthesis

将独立的物理实体、逻辑实体或两者兼有的构成元素组合成单一或统一实体的结果。

[来源：ISO/IEC/IEEE 24748-6:2023, 3.1.8]

3.4374

系统 system

为达到一个或多个规定目的而组织起来的、相互作用的元素的组合体。

注1：一个系统可被认为是一种产品或者是一种它所提供的服务。。

注2：实际中，对系统含义的解释通常通过使用一个联合名词来阐述，例如飞行器系统。有时候“系统”也可简单地由依赖语境的同义词来替代，例如飞行器，虽然这可能会使系统的视角不太明显。。

注3：完整的系统包括相关装置、设备、原料、计算机程序、固件、技术文档、服务和运行、支持所必需的人力，在预期的环境中实现某种程度的自给自足。。

[来源：GB/T 22032-2021; ISO/IEC 25000:2014, 3.1.61]

3.4375

系统分析记录 System analysis record

与系统分析相关的永久、可读形式的数据、信息或知识。

[来源: INCOSE 系统工程手册 第四版, 附录 E]

3.4376

系统分析报告 **System analysis report**

为相关方编制的报告, 用于通报系统分析活动的状态、进展与成果。包括成本分析、风险分析、有效性分析和其他关键特征分析的结果。还包括为分析而开发的所有模型或模拟。

[来源: INCOSE 系统工程手册 第四版, 附录 E]

3.4377

系统分析策略 **System analysis strategy**

完成要进行的各种分析所需的方法、时间表、资源和具体考虑因素, 包括方法、程序、评估标准或参数。

[来源: INCOSE 系统工程手册 第四版, 附录 E]

3.4378

系统与软件产品质量 **system and software product quality**

系统和/或软件在规定条件下使用时满足明确和隐含需要的能力。

注: 产品质量模型是指 GB/T 25000.10 中定义的系统与软件产品质量模型。

[来源: GB/T 25000.20-2021, 3.4]

3.4379

系统架构师 **system architect**

负责系统的架构设计, 协调工程团队制定解决复杂问题的方案, 并监督方案实现的人员、团队或组织。

[来源: INCOSE 系统工程手册, 第五版.]

3.4380

系统体系结构 **system architecture**

系统各部件之间的结构和关系。系统体系结构也可以包括系统和它的运行环境之间的界面。

[来源: GB/T 11457-2006, 2.1647]

3.4381

系统架构描述 **System architecture description**

所选系统架构的描述, 通常以一组架构视图(例如, 来自架构框架的视图)、模型(例如, 逻辑和物理模型, 尽管还有其他类型的模型可能有用)和架构特征(例如, 物理尺寸、环境抵抗力、执行效率、可操作性、可靠性、可维护性、模块化、鲁棒性、保障性、可理解性等)。

[来源: ISO/IEC/IEEE 42010, 2010]。在这个工件中, 在某种程度上识别和定义了架构上重要的系统元素。(随着设计的充实, 在设计定义过程中可能需要添加其他系统元素)

3.4382

系统架构依据 **system architecture rationale**

选择架构、选择技术/工艺系统的元素以及在系统需求和架构实体之间进行分配的理由和依据(例如, 功能、输入/输出流、系统元素、物理接口、架构特性、信息/数据元素、容器、节点、链接、通信资源)。

[来源: INCOSE 系统工程手册 第四版, 附录 E]

3.4383

系统边界 **system boundary**

被度量软件与用户或其他系统之间的界限。

[来源: GB/T 36964-2018, 3.11]

3. 4384

系统分解结构 system breakdown structure; SBS

包含已识别的使能系统及人员的系统层级结构，通常用于分配开发团队、支持技术评审，并将已分配的工作及相关的资源划分到为完成项目技术目标所必须的各项任务中。

注：它还为成本跟踪和控制提供基础。

[来源：ISO/IEC/IEEE 24748-4:2016, 4.12]

3. 4385

系统特性 system characteristic

与系统相关的属性或区分特征。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4672]

3. 4386

系统描述 system description

系统设计产生的文件，定义系统的组织、基本特征和硬件和软件要求[来源：ISO/IEC 2382:2015]。

3. 4387

系统设计 system design

定义系统的硬件和软件架构、组件、模块、接口和数据以满足特定要求的过程。

3. 4388 [[来源：ISO/IEC 2382:2015, X.X]

系统设计描述 System design description

所选系统设计的描述。识别和定义系统要素。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3. 4389

系统设计原理 system design rationale

设计选择、系统要素选择以及系统需求和系统要素之间分配的基本原理。包括主要选定实施方案和促成因素的基本原理。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3. 4390

系统开发 system development

通常包括需求分析、系统设计、实施、文档编制和质量保证的过程。

3. 4391 [[来源：ISO/IEC 2382:2015, X.X]

系统开发周期 system development cycle

从决定开发系统开始，到系统交付给最终用户为止的这段时期。

参见：系统生存周期、软件开发周期。

注：该术语有时也用于指代更长的时间段，即系统不再进行增强时结束的时期，或整个系统生存周期。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3. 4392

系统文档 system documentation

描述信息处理系统的要求、能力、限制、设计、操作和维护的文件集合。

3. 4393 [[来源：ISO/IEC 2382:2015, X.X]

系统效能 system effectiveness

系统在其预期使用条件下满足预期使用要求的程度的测量，取决于系统在预期环境条件下的执行情况，以及在其整个生存周期内生产、测试、分发、运行、支持、培训和处置系统的能力。

[来源：IEEE 1220-2005, 3.1.40]

3. 4394

系统效能分析 system effectiveness analysis

用于确定系统在其预期使用环境中性能表现的分析方法。

[来源：ISO/IEC/IEEE 24748-7:2019, 3.1]

3. 4395

系统要素 system element

系统中可被实现以满足规定要求的独立组成部分。

[来源：ISO/IEC/IEEE 15288:2023, 3.47]

构成系统的一组要素中的成员。

注：系统要素是系统中可被实现以满足规定要求的独立组成部分。

[来源：ISO/IEC/IEEE 12207:2026, 3.1.62]

满足特定需求的一个系统的离散部分。

示例：硬件、软件、数据、人、过程（例如为用户提供服务的过程）、规程（例如操作指南）、设备、材料、自然存在的实体或其任意组合。

[来源：ISO/IEC/IEEE 15288:2023, 3.47]

[来源：GB/T 45803-2025, X.X]

组成系统的一组元素中的一个成员。

示例：硬件、软件、数据、人、过程（例如提供给用户服务的过程）、规程（例如操作指南）、设备、原料、自然存在的实体或其任意组合。

注：系统元素是系统中离散的部分，通过实现它可以完成规定的要求。

[来源：GB/T 22032-2021, X.X]

3. 4396

系统要素描述 system element description

对系统中包含的系统要素的设计特性描述；该描述取决于实现技术（例如：数据表、数据库、文档、可导出数据文件）。

[来源：INCOSE-TP-2003-002-04 2015]

3. 4397

系统要素文档 system element documentation

详细图纸、代码及材料规格。因采购、合规要求或纠正措施所需的修订设计文档。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3. 4398

系统工程组 system engineering group

负责以下工作的人员的集合（包括经理和技术人员）：规定系统需求；将系统需求分配给硬件、软件和其他成分；规定硬件、软件和其他成分间的接口；以及为保证与相应规范的符合性而监控这些成分的设计和开发。

[来源：GB/T 11457-2006, 2.1654]

3. 4399

系统实体 system entity

在 MkII 功能点分析中，将一个应用程序包含的所有非主实体组合在一起标记的实体。

[来源：GB/T 42566-2023, 1.788]

3. 4400

系统跟踪 system follow-up

系统达到稳定运行状态后的效果研究。

[来源：ISO/IEC 2382:2015, X. X]

3.4401

系统跟进 system follow-up

实施后审查 post-implementation review

开发后审查 post-development review

对系统达到稳定运行状态后的效果研究。

[来源：ISO/IEC 2382:2015, X. X]

3.4402

系统跟踪 system follow-up

实施后评审 post-implementation review

开发后评审 post-development review

对系统在达到稳定运行状态后的影响研究。

[来源：ISO/IEC 2382:2015, X. X]

3.4403

系统功能定义 system function definition

定义系统的功能边界和系统必须执行的功能。

[来源：INCOSE 系统工程手册 第四版, 附录 E]

3.4404

系统功能识别 system function identification

系统功能的识别。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4405

系统功能接口识别 system functional interface identification

识别和记录与边界外部系统的功能接口以及相应的信息交换要求。

[来源：INCOSE 系统工程手册 第四版, 附录 E]

3.4406

系统危害 system hazard

事故发生的前提系统状态。

[来源：IEEE 1228-1994, 3.1.8]

3.4407

系统集成 system integration

将系统组件逐步组装成整个系统。

[来源：ISO/IEC 2382:2015, X. X]

3.4408

系统集成模块 system integration module; SIM

控制系统功能单元的微控制器单元（MCU）中的模块。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4685]

3.4409

系统接口任务 system interface task

隐藏与外部系统或子系统的接口并与之通信的任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4686]

3.4410

系统库 system library

包含系统驻留软件的软件库，可以被访问使用或通过引用将其合并到其他程序中。

参见：主库、生产库、软件开发库、软件存储库。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4687]

3.4411

系统生存周期 system life cycle

系统从构思到停止使用期间经历的开发变化过程。

[来源：ISO/IEC 2382:2015, X.X]

3.4412

系统维护 system maintenance

修改系统以纠正故障、提高性能或使系统适应变化的环境或变化的要求。

[来源：ISO/IEC 2382:2015, X.X]

3.4413

系统模型 system model

在计算机性能评估中，对系统的一种描述，用于展示系统中工作负载与性能指标之间的关系。

[来源：ISO/IEC/IEEE 24765:2017 系统与软件工程——术语]。

参见：工作负载模型。

3.4414

所关注系统 system of interest; SOI

正在考虑其生存周期的系统。

[来源：INCOSE 系统工程手册 第四版 Appendix C]

3.4415

系统之系统 system of systems; SOS

一组系统，它们通过集成或互操作提供一种独特能力，而任何一个组成系统都无法单独实现这种能力。

注：通常，系统之系统（SoS）用于处理包含多个、异构、分布式系统的大规模跨学科问题。每个组成系统本身都是一个独立可用的系统，拥有自身的管理、目标和资源，但会在系统之系统内进行协同，以提供系统之系统的独特能力。

[来源：ISO/IEC/IEEE 12207:2026, 3.1.64]

3.4416

系统之系统工程 system of systems engineering; SoSE

对现有系统和新系统组合的能力进行规划、分析、组织、开发与集成，并将系统间基础设施、设施及总体流程纳入其中，使之形成体系能力，且该能力大于各组成系统能力之和的过程。

注：系统之系统工程（SoSE）还包括测试、修改、维护及其他集成后活动。

[来源：ISO/IEC/IEEE 21841:2019, 3.1.5]

3.4417

片上系统 system on a chip; SoC

片上系统。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4302]

3.4418

可编程芯片系统 system on a programmable chip; SoPC

可编程芯片系统。

[来源：ISO/IEC/IEEE FDIS 24765 3.4412]

3.4419

系统配置文件 system profile

计算机性能评估中使用的指标集，描述计算机系统中各主要资源的忙碌时间占比，除以该资源的可用时间。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4420

系统可靠性 system reliability

包括全部硬件和软件子系统在内的某个系统，在规定的环境及时间里正确执行所要求的任务或使命的概率。

参见：运行可靠性、软件可靠性

[来源：GB/T 11457-2006, 2.1662]

3.4421

系统需求 system requirement

为了满足用户解决问题需要的条件或能力，系统或系统成分必须满足或具有的条件或能力。

[来源：GB/T 11457-2006, 2.1663]

3.4422

系统需求评审 system requirement review; system requirements review; SRR

一种评审行为，它评价为系统定义的需求是否完整和适当、评价产生这些需求的系统工程过程、评定系统工程试作的结果和评价系统工程计划。

参见：软件需求评审

[来源：GB/T 11457-2006, 2.1665]

3.4423

系统需求规格 system requirements

系统及其运行环境和外部接口的结构化需求集合。

[来源：GB/T 45802-2025, X.X]

3.4424

分配给软件的系统需求 system requirements allocated to software

拟由系统的软件成分实现的系统需求的子集。分配需求是软件开发计划的主要输入。软件需求分析将精心加工和细化分配需求，并产生文档化的软件需求。简称分配需求。

[来源：GB/T 11457-2006, 2.1664]

3.4425

系统需求定义记录 system requirements definition record

与系统需求定义相关的永久、可读形式的数据、信息或知识。

[来源：INCOSE 系统工程手册 第四版，附录 E]

3.4426

系统需求定义策略 system requirements definition strategy

用于识别和定义系统需求并在整个生存周期中管理需求所需的方法、技术、资源和特定注意事项。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4427

系统需求审查 system Requirements Review; SRR

对一个或多个软件配置项所规定需求的评审，目的是评估这些需求对系统需求的响应性与解读正确性，并确定这些需求是否构成开展配置项初步设计的满意依据。

注：美国国防部将此类评审称为软件规范评审。

[来源：ISO/IEC/IEEE 24765:2017]

3.4428

系统需求规范 system requirements specification; SyRS

系统及其运行环境、外部接口的需求（包括功能、性能、设计约束及属性）的结构化集合。

参见：软件需求规格说明书，SRS。

[来源：ISO/IEC/IEEE 29148:2018, 4.1.29]

3.4429

系统需求可追溯性 System requirements traceability

系统需求的双向可追溯性。

[来源：INCOSE 系统工程手册 第四版 附录 E]

3.4430

系统安全 system safety

系统免受不可接受风险的能力。

参见：软件安全。

注：基于系统的安全方法，是运用科学、技术和管理技能在系统、计划、项目、活动或产品的整个生存周期中识别危害、分析危害，并消除、控制或管理危害。

[来源：ISO/IEC 23643:2020, 3.25]

3.4431

系统软件 system software

旨在促进计算机系统及其相关程序的操作和维护的软件。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4432

系统规范 system specification

系统的一套强制性要求文件。

示例：系统性能规范[SPS]、系统需求文档[SRD]和系统/子系统规范[SSS]。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.1]

3.4433

系统结构 system structure

将所关注的系统分解为一组相互作用的子系统与系统要素。

注：系统结构在 系统分解结构（SBS）中描述。

[来源：ISO/IEC 29110-5-6-4:2025 Part 5-6-4 : 3.22]

3.4434

系统支持 system support

持续提供使用和改进已实施系统所需的服务和材料。

[来源：ISO/IEC 2382:2015, X.X]

3.4435

系统测试 system testing

在完整、集成的系统上开展的测试，旨在评估系统是否符合规定的要求。

[来源：ISO/IEC 29110-5-1-2:2025, 3.35]

3.4436

被测系统 **system under test; SUT**

基于计算机的软件系统（CBSS）中需要被测试的部分。

注：待测试系统（SUT）由硬件、系统软件、数据通信功能或应用软件或其组合构成。

[来源：ISO/IEC 14756:1999, 4.18]

3.4437

系统性失效 **systematic failure**

故障以确定性方式与特定原因相关，只能通过修改设计或制造过程、操作程序、文档或其他相关因素来消除。

一种与特定原因以确定性方式相关的失效，只有通过修改设计、制造工艺、操作规程、文档或其他相关因素才能消除。

[来源：ISO/IEC/IEEE 15026-1:2019, 3.4.10]

3.4438

系统化重用 **systematic reuse**

根据定义明确、可重复的过程进行重用实践。

[来源：IEEE 1517-2010 IEEE, 3]。

3.4439

目标系统 **system-of-interest**

其生存周期在本标准语境中考量的系统。

[来源：GB/T 45615-2025, XXXXX]

3.4440

系统分析 **systems analysis**

对实在的或计划好和系统所进行的一种系统性调研，其目的是确定该系统的信息要求和过程，它们之间的相互关系，以及与其他系统的关系。

[来源：GB/T 11457-2006, 2.1646]

3.4441

系统开发生存周期 **systems development life cycle**

与系统开发相关的活动，包括系统需求、设计、验证、部署、维护及生存周期终止。

同义词：SDLC（系统开发生存周期）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4704]

3.4442

系统工程 **systems engineering**

一种跨学科方法，用于控制将一系列利益相关方的要求、期望、约束转化为解决方案，并在整个生存周期中支持此解决方案所需要的全部技术工作和管理工作。

注：System engineering 用于单一所关注系统的场景；systems engineering 泛指该学科。它包括技术性能指标定义、为构建体系结构而开展的各工程专业集成，以及平衡成本、性能和进度目标的支撑性生存周期过程定义。

[来源：ISO/IEC/IEEE 15288:2023, 3.50]

3.4443

系统 eng 程工作 **systems engineering effort**

系统 eng 程工作将多个学科和专业组集成为一组，从概念推进到生产再到运行。系统 eng 程考虑所有利益相关方的业务和技术需求，旨在提供满足其需求的高质量系统。

[来源：INCOSE 系统 eng 程手册 第四版, APPENDIX C]

3.4444

系统工程计划 Systems Engineering Plan; SEP; SEMP

系统工程管理计划 Systems Engineering Management Plan

用于管理系统工程工作的顶层技术计划，该计划定义项目技术层面如何组织、构建与实施，以及如何管控系统工程过程，以提供满足相关方需求的产品。

[来源：ISO/IEC/IEEE 24748-4:2016, 4.14]

3.4445

系统集成测试 systems integration testing

对多个完整的集成系统进行测试，以评估它们相互成功通信的能力以及满足整个集成系统的特定要求的能力。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4707]

3.4446

测试与评估 T&E

测试和评估。

[来源：IEEE 15288.2:2014 IEEE, 3.2]

3.4447

表[格] table

实体的更具体表示。

[来源：ISO/IEC 15476-4:2005, 6.3, 有修改]

3.4448

表头 table heading

从其他文档中引用决策表的符号名称或其他方式。

注：或是对表格的清晰描述。

[来源：ISO 5806:1984, 3.12]

3.4449

表驱动法 table-driven method

通过查表方式获取信息、替代逻辑判断的机制。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4717]_x000D_

3.4450

标签 tag

2. 提供关于软件资产的权威信息，以支持其管理的信息架构。

注：为开发方和最终用户提供与其正在使用的代码库对应的唯一参考标识。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4719]

3.4451

标签遍历 tag slide

将相同的标签应用于文件的更改版本，以更正发布中的临期错误。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4720]

3.4452

标签 ID tagid

一种全局唯一值，对所生成的每个软件标识（SWID）标签均保持全局唯一。

[来源：ISO/IEC 19770-2:2015, 3.1.3]_x000D_

3.4453

定制化过程 tailored process

通过对标准过程进行裁剪而形成的过程。

[来源：ISO/IEC 33001:2015, 3.1.10]

3.4454

裁剪 tailoring

通过选用（特殊情况下可修改现有需求或新增需求），对规范、标准及相关文件中的各项单独需求进行评估，并使其适用于特定项目的过程。

注：定制可应用于项目的各个方面，包括项目文档；在每个生存周期阶段执行的过程和活动；或符合适用法定要求的审查、分析和决策的时间和范围。

[来源：ISO/IEC/IEEE 26513:2017, 3.38]

3.4455

剪裁指南 tailoring guideline

使组织能够适当地调整标准过程，以满足特定需求的指导说明。

注：对过程的调整是为特定目的而对过程描述进行的适应性修改。例如，一个项目通过调整组织的一套标准过程来满足项目的目标、约束条件和环境要求。组织的一套标准过程是以一般性水平描述的，不能直接用于执行该过程。调整指南有助于那些为特定需求制定定义过程的人。调整指南描述了哪些内容可以修改、哪些内容不能修改，并指出了可以进行修改的过程组件。

[来源：ISO/IEC 33001:2015, 3.1.11]

3.4456

有形性 tangibility

IT 服务的有形方面有效沟通和支持服务的程度。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4725]

3.4457

目标受众 target audience

供应商预期的信息使用对象群体。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4726]

3.4458

目标能力 target capability

由过程能力确定发起人鉴定的过程能力，它代表对成功地实施特定要求的一种可接受的过程风险。

[来源：GB/T 11457-2006, 2.1674]

3.4459

目标计算机 target computer

拟运行交付软件的计算机。

参见：2.709 条 宿主计算机 (host computer)。

3.4460

目标实体 target entity

与用户有关，持有自身信息并需加以测量的基础事物。

[来源：GB/T 25000.21-2019, 4.17]_x000D_

3.4461

目标语言 target language

源语言内容被翻译成的语言。

同义词：对象语言。

参见：源语言。

示例：汇编器或编译器生成的语言。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.32, 有修改]

3.4462

目标语言 目标语言 target language object language

将源语言内容翻译为其形式的语言。

参见：源语言（source language）。

示例：汇编程序或编译器输出的语言。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.32]

3.4463

目标机 target machine

正在被另一台计算机模拟的计算机。

参见：主机。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4729, 有修改]

3.4464

目标节点 target node

弧终点关联节点。

[来源：ISO/IEC 15909-2:2011, 4.1.18]

3.4465

过程目标 target of process

测量或评价过程所作用的系统、软件产品或由系统或软件产品所执行的任务。

[来源：GB/T 25000.1-2021, 3.48]_x000D_

3.4466

验证目标 target of verification; TOV

待验证的软件、一组软件项或软件单元，例如从安全性与保密性方面进行验证。

[来源：ISO/IEC 23643:2020, 3.26]

3.4467

目标过程概要 target process profile

描述所需过程属性及其对每个所需过程属性所要求的评定等级。

[来源：ISO/IEC 33001:2015, 3.2.20]

3.4468

目标风险 target risk

预期达到的风险。

[来源：ISO/IEC/IEEE 15026-1:2019 《系统与软件工程 —系统与软件保障 第1部分：概念和术语》 3.3.13]

3.4469

任务 task

随管理可审核性而定的最小工作单元；或者，指派给一个或多个项目成员并明确界定的工作。

注：通常将有关任务分组，以此形成各种活动。

[来源：GB/T 41866.1-2022, 3.20, 有修改]

3.4470

任务行为规格说明 task behaviour specification

描述并发任务的接口、结构、时序特征、相对优先级、检测到的错误和任务事件排序逻辑的规格说明。

[来源：ISO/IEC/IEEE 24765:2017；ISO/IEC/IEEE FDIS 24765, 3.4737]

3.4471

任务看板 task board

对计划工作进度的可视化表示，使每个人都能看到任务的状态。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4472

任务完成 task completion

对于特定任务，总输出字符串或一组输出字符串的所有部分被模拟用户或另一实例完全接收的及时事件。

注：任务完成时间定义了前面准备时间的结束时间和后续任务执行时间的开始时间。

[来源：ISO/IEC 14756:1999, 4.2]

3.4473

任务接口 task interface

输入或输出、信号化的事件（输入或输出）、外部输入或输出，或对被动对象的访问。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4740]

3.4474

任务倒置 task inversion

系统中的任务可以被系统地合并的优化概念。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4741]

3.4475

任务启动会 task kick-off meeting

在项目任务开始时举行的会议，目的使有关人员为有效地执行任务的各项活动做好准备。

[来源：现有 11457 定义]

3.4476

任务模式 task mode

用户的准备时间是从前一任务提交时立即开始（值为 0，即“无等待”），还是在前一任务完成后（任务完成）才开始（值为 1，即“等待”）的指示。

注：基于 UNIX 的系统中的“对话”或“批处理”模式。

[来源：ISO/IEC 14756:1999, 4.21]

3.4477

任务优先级准则 task priority criteria

衡量给定任务执行优先级的任务构建准则分类。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4743]

3.4478

任务构建 task structuring

将并发应用程序构建为并发任务并定义任务接口的软件设计阶段。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4744]

3.4479

任务提交 task submission

当模拟用户将输入字符串完全提交至 SUT（被测系统）且任务可以开始的及时事件，无论 SUT 是否立即开始执行。

注：通常任务提交由提交特殊字符（例如回车）或者输入字符串末尾或输入字符串若干部分末尾的字符序列来内部定义。此外，任务提交事件也常由字符串中指定数量的最后一个字符提交来定义。对于经典批处理任务，任务提交由批处理命令序列中最后一个字符串的最后一个字符的提交来定义。

[来源：ISO/IEC 14756:1999, 4.22]

3.4480

任务类型 task type

任务的分类由以下因素的组合所(1) 活动类型，或一组属于相同及时性函数和任务模式的活动类型(2) 及时性函数；任务模式。

[来源：ISO/IEC 14756:1999, 4.23]。

注：模拟用户仅向 SUT（被测系统）提交此类任务。

3.4481

任务板 taskboard

对敏捷团队待完成任务及团队近期进展的可视化展示。

[来源：ISO/IEC TR 24587:2021, 3.17]

3.4482

任务聚类准则 task-clustering criteria

衡量是否及如何将对象聚合成并发任务的判定准则类别。

[来源：ISO/IEC/IEEE FDIS24765, 3.4747]

3.4483

任务构建准则 task-structuring criteria

帮助设计者将系统构建为并发任务的一组启发式规则。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4748]

3.4484

分类学 taxonomy

用于明确引用概要文件或概要文件集的分类方案。

注：该分类方案用于对知识体系进行分类和理解。

[来源：ISO/IEC 29110-2-1:2015, 4.63, 有修改]

3.4485

TCPI TCPI; TCPI

完工尚需绩效指数。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4756]

3.4486

测试驱动开发 TDD

测试驱动开发。

[来源：IEEE 2675-2021, 3.2]

3.4487

团队 team

一组人，这些人通常来自各个不同的但相关的组，受指派去执行组织或项目的一个妥善定义的职能。

团队成员可以是兼职的，而且可有其他主要职责。

[来源：GB/T 11457-2006, 2.1683]

3.4488

团队绩效域 team performance domain

涉及与负责生产实现业务成果的项目可交付物的人员相关的活动和职能的绩效领域。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4489

团队选拔计划 team selection plan

说明项目人员的资格、经验和培训需求的文件[无来源]。

3.4490

技术复杂性调整 technical complexity adjustment

一个将技术要求和质量要求对应用程序规模的影响纳入考虑的调整系数，使用该系数后将得出调整规模。

注：如果执行了此操作，得到的是调整规模，而不是功能规模。

[来源：GB/T 42566-2023, 3.29]

3.4491

技术复杂性调整因素 technical complexity adjustment factors

技术复杂度调整中要考虑的 19 个系数的集合。每个系数的影响程度在 1 到 5 之间。

[来源：GB/T 42566-2023]

3.4492

技术债务 technical debt

为在后续需要时提供能力而推迟工作所产生的成本。

[来源：ISO/IEC/IEEE FDIS 24765, X.X]

3.4493

技术独立 technical independence

软件质量保证（SQA）领域中，SQA 活动使用未参与系统或其组成部分开发人员的情况。

[来源：IEEE 730-2014, 3.2]

3.4494

技术管理 technical management

为计划、组织和控制工程功能，技术的和管理的资源的应用。

[来源：GB/T 11475 2006, 2.1684]

3.4495

技术性能测量 technical performance measure; TPM

通过技术性能的可量化度量，帮助确保系统组成部分满足技术要求。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4496

技术产品质量需求 technical product quality requirement

对于产品开发和维护过程中用到，技术上已标识的特性的产品质量需求。

[来源：GB/T 25000.30-2021, 3.19]

3.4497

技术需求 technical requirement; TR

与用于软件开发、维护、支持和执行的技术及环境相关的需求。

示例：编程语言、测试工具、操作系统、数据库技术、用户接口技术。

[来源：ISO/IEC/IEEE 41062:2024, 3.1.20]

3.4498

技术审查 technical review

在系统生存周期的逻辑节点上开展的一系列系统工程活动，通过一套双方商定的评判标准，对照项目技术要求评估项目进展。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.1]

3.4499

技术标准 technical standard

描述在产品开发或提供服务时应用积累的技术或管理技能和方法的特征的标准。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4774, 有修改]

3.4500

技术路线图 technology roadmap

对技术方面必要且预期的变更进行概要说明，并附预计时间节点，用以支撑产品或产品族的实现或转型。

[来源：ISO/IEC 26560:2019《软件与系统工程 — 产品线产品管理的工具与方法》，3.6]

对技术方面必要且预期的变更进行概要说明，并附预计时间节点，用以支撑系统的实现或转型。

参见：产品路线图（product roadmap）。

[来源：ISO/IEC/IEEE 24641:2023, 3.1.38]

3.4501

技术视图 technology viewpoint

聚焦于系统及其环境中技术选型的开放分布式处理（ODP）系统视图。

[来源：ISO/IEC 10746-3:2009, 4.1.1.5]

3.4502

温度 TEMP

测试与评估主计划。

[来源：ISO/IEC/IEEE 24748-8:2019, 3.2]

3.4503

模板 template

可用于构建实例化资产的可配置参数资产。

[来源：ISO/IEC/IEEE 24765j:2021, 有修改]

为收集、整理和呈现信息及数据提供明确结构的，预定义格式的半成品文档。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4779, 有修改]

一组<X>的公共特性规格说明，足以支撑<X>的实例化。

注：<X>代表可被建模的任意实体类型。当<X>被替换为具体实体（如对象、接口）时，即构成该实体的模板类型定义。

[来源：ISO/IEC 10746-2:2009, 9.13, 有修改]

3.4504

模板 <X> 模板 template <X> template

具有可用于构建实例化资产的参数的资产。

采用预定义格式的部分完成的文档，为信息与数据的收集、整理和呈现提供规定的结构。

[来源：《项目管理知识体系指南（PMBOK® 指南）》第七版]

对一组<X>的共同特征所做的详细规范，其详细程度足以实例化一个 X。

（参考模型：基础，9.13）。

注：<X>代表可被建模的任意实体类型。当<X>被替换为具体实体（如对象、接口）时，即构成该实体的模板类型定义。

参见：构造

3.4505

模板类 `template class`

针对某一<X>而言，指所有满足<X>模板类型的<X>实例构成的集合。

注：<X>代表可被建模的任意实体类型。当<X>被替换为具体实体（如对象、接口）时，即构成该实体的模板类型定义。

[来源：ISO/IEC 10746-2:2009, 9.23]

3.4506

模板评级模块 `template rating module`

一种质量评级模块，其中测量环境和评级方法已被参数化，以供未来使用。

[来源：ISO/IEC 25040:2024, 3.9]

3.4507

模板类型 `template type`

针对某一<X>而言，指在模板中定义的一个谓词，该谓词对模板的所有实例均成立，并表达模板实例旨在满足的需求。

注：<X>代表可被建模的任意实体类型。当<X>被替换为具体实体（如对象、接口）时，即构成该实体的模板类型定义。

[来源：ISO/IEC 10746-2:2009, 9.22]

3.4508

时间聚类 `temporal clustering`

一种任务结构化准则：将无先后顺序依赖关系、但由同一事件触发的活动归并为一个任务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4783]

3.4509

时间内聚 `temporal cohesion`

内聚的类型，其中软件模块执行的任务都需要在程序执行的特定阶段进行，参见：偶然内聚、通信内聚、功能内聚、逻辑内聚、过程内聚、顺序内聚。

示例：包含程序所有初始化任务的模块。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4784, 有修改]

3.4510

项 `term`

由签名和一组排序变量构建的，常量、变量和运算符组成的表达式。

[来源：ISO/IEC 15909-1:2004, 2.1.24]

用于指代一个实体的语言构造。

[来源：ISO/IEC 10746-2:2009, 7.1]

3.4511

任期评价 `term evaluation`

系统或通信网络中可输入或检索数据的功能单元。

[来源：ISO/IEC 2382:2015, X.X]

3.4512 终端 terminal

系统或通信网络中可输入或检索数据的功能单元。

[来源: ISO/IEC 2382:2015, X.X]

3.4513**终结符 terminal symbol**

在层次结构中不进一步分解的语法的层次结构定义的一部分。

[来源: ISO/IEC 15474-1:2002, 4.2]

3.4514**终止行为 terminating behaviour**

终止联络并拒绝相应契约上下文及相应契约的行为。

[来源: ISO/IEC 10746-2:2009, 13.2.5]

3.4515**终止构造 termination construct**

导致停机或退出的程序构造。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4789, 有修改]

3.4516**终止交付 termination deliver**

客户端计算对象的隐式定义信号接口中的信号, 其名称和参数与原始操作接口中的询问终端之一相同。

[来源: ISO/IEC 14752:2000, 3.3.13]

3.4517**终止证明 termination proof**

专门设计用于收集、维护和访问术语数据的软件工具。

[来源: ISO/IEC/IEEE 26511:2018 系统和软件工程—对系统、软件和服务用户信息管理人员的要求, 3.1.34]

3.4518**终止提交 termination submit**

服务器计算对象的隐式定义信号接口中的信号, 其名称和参数与原始操作接口中的询问终端之一相同。

参见: 调用提交, 调用传递。

[来源: ISO/IEC 14752:2000, 3.3.14]

3.4519**术语管理系统 terminology management system**

(1) 专门设计用于收集、维护和访问术语数据的软件工具。

[来源: ISO/IEC/IEEE 26511:2018 系统和软件工程—对系统、软件和服务用户信息管理人员的要求, 3.1.34]

3.4520**测试方法 test approach**

用于确定特定测试用例输入值的具体方法。

参见: 测试实践。

注: 从非常通用(例如, 黑盒或白盒)到非常具体(例如, 最小和最大边界值)的特异性各不相同。

[来源: ISO/IEC/IEEE FDIS FDIS 24765, 3.4794]

3.4521

测试依据 test basis

作为测试分析和测试用例设计基础的知识体系。

注：测试依据可以采用文件的形式，例如需求规格说明、设计规格说明或模块规格说明，但也可以是非书面形式对需求行为的理解。

[来源：GB/T 38634.2-2020]

3.4522

测试台 test bed

包含进行测试所需的硬件、仪器、模拟器、软件工具和其他支持元素的环境。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4796, 有修改]

3.4523

测试用例 test case

前置条件、输入（包括操作，如果适用）和预期结果的集合，用于驱动测试项的执行以满足测试目标，测试目标包括正确实现、错误识别、检查质量和其他有价值的信息。

注1：测试用例是测试子过程的最低测试输入级别。（即，测试用例无法再划分为更细的测试用例）。

注2：测试用例的前置条件包括测试环境、已有数据（如数据库）、被测软件、硬件等。

注3：输入是用于驱动测试执行的数据信息。

注4：预期结果包含通过的准则、失效的校核。

[来源：GB/T 38634.1-2020, 3.48, 有修改]_x000D_

3.4524

测试用例生成器 test case generator

接受源代码、测试标准、规范或数据结构定义作为输入的软件工具；使用这些输入生成测试输入数据；有时确定预期结果。

同义词：测试数据生成器、测试生成器。

[来源：ISO/IEC/IEEE 24765:2017, 有修改]

3.4525

测试用例生成器 test case generator**测试数据生成器 test data generator****测试生成器 test generator**

一种软件工具，接受源代码、测试准则、规格说明或数据结构定义作为输入，利用这些输入生成测试输入数据，并可确定预期结果。

同义词：测试数据生成器（test data generator）、测试生成器（test generator）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4798]

3.4526

测试用例规格说明 test case specification

一个或多个测试用例组成的文档集。

[来源：GB/T 38634.1-2020, 3.49, 有修改]_x000D_

3.4527

测试类 test class

指定的测试用例分组。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4800]

3.4528

测试完成过程 test completion process

测试管理过程的子过程。用于确保有用的测试资产可供以后使用、测试环境保持在令人满意的状态、测试结果被记录并传达给利益相关方。

[来源：GB/T 38634.1-2020]

3.4529

测试完成报告 test completion report

描述已完成测试的总结报告。

注：也被称为测试总结报告。

[来源：GB/T 38634.1-2020, 3.51]_x000D_

3.4530

测试完成报告 test completion report

测试总结报告 test summary report

对已执行的测试工作进行汇总的报告。

注：也被称为测试总结报告。

[来源：GB/T 38634.1-2020, 3.51]

3.4531

测试条件 test condition

组件或系统可测的方面，如作为测试依据的功能、事务、特征、质量属性或者结构元素。

注：测试条件常用来导出覆盖项，或由其本身构成覆盖项。

[来源：GB/T 38634.1-2020, 3.52]_x000D_

3.4532

测试覆盖（范围） test coverage

以百分比表示的、用以表示一个或多个测试用例实现指定测试覆盖项的程度。

[来源：GB/T 38634.1-2020, 3.53, 有修改]_x000D_

3.4533

测试覆盖项 test coverage item

使用测试设计技术从一个或多个测试条件导出的属性或属性组合，可以用于测量测试执行的充分性。

[来源：GB/T 38634.1-2020, 3.54]_x000D_

3.4534

测试覆盖项 test coverage item

覆盖项 coverage item

使用测试设计技术从一个或多个测试条件导出的属性或属性组合，可以用于测量测试执行的充分性。

[来源：GB/T 38634.1-2020, 3.54]

3.4535

测试准则 test criteria

系统或组件为通过某项测试所必须满足的准则。

参见：验收准则，通过/失败准则。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4806, 有修改]

3.4536

测试数据 test data

为满足执行一个或多个测试用例的输入需求而创建或选择的数据，该数据可在测试计划、测试用例和测试规程中定义。

注：测试数据可以存储在被测的产品中（例如阵列、平面文件或数据库），也可以从外部源获得或由外部源提供，如其他系统、其他系统组件、硬件设备或人员提供。

[来源：GB/T 38634.1-2020, 3.55, 有修改]_x000D_

3.4537

测试数据生成器 test data generator

一种以源代码、测试准则、规格说明或数据结构定义为输入，利用这些输入生成测试输入数据，并有时可确定预期结果的软件工具。

3.4538

测试数据准备报告 test data readiness report

描述每个测试数据需求准备状态的文档。

[来源：GB/T 38634.1-2020, 3.56]_x000D_

3.4539

测试描述 test description

对一组或多组测试用例进行记录的文档。

参见：测试用例规格说明（test case specification）。

[来源：ISO/IEC/IEEE 29119-2:2021《软件与系统工程 — 软件测试 第2部分：测试过程》，3.24]

3.4540

测试设计 test design

规定系统、软件或硬件特性或特性组合的测试方法细节并识别相关测试的文档。

注：通常包括测试的分组组织方式。

[来源：IEEE 1012-2024, 3.1.32, 有修改]

3.4541

测试设计和实现过程 test design and implementation process

生成和确定测试用例和测试规程的测试过程。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4810]

3.4542

测试设计规格说明 test design specification

规定测试项的被测特征及其相应测试条件的文档。

[来源：GB/T 38634.1-2020, 3.58]_x000D_

3.4543

测试设计技术 test design technique

用于构建测试模型的活动、概念、过程和模式，该模型用于识别测试项的测试条件，导出相应的测试覆盖项，并导出或选择测试用例。

[来源：GB/T 38634.1-2020, 3.59]_x000D_

3.4544

测试设计 test design

实施过程 implementation process

生成和确定测试用例和测试规程的测试过程。

[来源: GB/T 38634.1-2020]

3.4545

测试文档集 test documentation

测试活动固有文档的集合。

注: 类型包括测试用例规范、测试事件报告、测试日志、测试计划、测试程序、测试报告。

[来源: ISO/IEC 25051:2014, 4.1.19, 有修改]

3.4546

测试驱动程序 test driver

用于调用被测模块, 并通常提供测试输入、控制和监控执行过程、报告测试结果的软件模块。

参见: 测试装具。

[来源: ISO/IEC/IEEE FDIS FDIS 24765, 3.4813, 有修改]

3.4547

测试活动 test effort

执行一项或多项测试任务的活动。

[来源: ISO/IEC/IEEE FDIS FDIS 24765, 3.4814]

3.4548

测试环境 test environment

用于执行软件测试的设施、硬件、软件、固件、规程和文档集。

注: 测试环境可包括多种环境以适应指定的测试子过程(例如, 单元测试环境、性能测试环境等)。

[来源: GB/T 38634.1-2020, 3.60]_x000D_

3.4549

测试环境与数据管理过程 test environment and data management process

建立和维护所需测试环境及对应测试数据的测试过程。

[来源: ISO/IEC/IEEE 29119-2:2021, 3.37]

3.4550

测试环境项 test environment item

可与测试环境其他部分分开考虑的测试环境元素。

[来源: ISO/IEC/IEEE 29119-3:2021, 3.11]

3.4551

测试环境准备报告 test environment readiness report

描述每个测试环境需求实现程度的文档。

[来源: GB/T 38634.1-2020, 3.61]_x000D_

3.4552

测试环境需求 test environment requirements

测试环境必要性质的描述。

注: 所有或部分的测试环境需求可以参考可获取的信息, 例如合适的组织级测试策略、测试计划和/或测试规格说明。

[来源: GB/T 38634.1-2020, 3.62]_x000D_

3.4553

测试环境构建过程 test environment set-up process

建立和维护所需的测试环境的过程。

[来源: GB/T 38634.1-2020, 3.63]_x000D_

3.4554

测试执行 test execution

在测试项上执行测试并产生实测结果的过程。

[来源: GB/T 38634.1-2020]

3.4555

测试执行引擎 test execution engine

操控测试项执行测试用例的工具。

注1: 测试执行引擎多由软件实现, 有时也会由硬件实现。

注2: 一个典型的测试执行引擎包含单元测试工具框架、激励命令系统、捕捉回放工具, 或硬件机器人以及控制它们的软件。

[来源: GB/T 38634.5-2024, 3.11]

3.4556

测试执行日志 test execution log

记录一个或多个测试规程执行细节的文档。

注: 也称为测试记录。

[来源: GB/T 38634.1-2020, 3.65]_x000D_

3.4557

测试执行过程 test execution process

动态测试过程的子过程。用于在准备好的测试环境中执行测试设计和实现过程中创建的测试规程, 并记录其结果。

[来源: GB/T 38634.1-2020]

3.4558

测试框架 test framework

用于组织、选择和交流测试的一组结构化的原则、指导方针和实践。

[来源: GB/T 38634.5-2024, 3.12]

3.4559

测试发生器 test generator

一种软件工具, 它接收源代码、测试准则、规格说明或数据结构定义作为输入, 利用这些输入生成测试输入数据, 有时还会确定预期结果。

3.4560

测试事件 test incident

在测试执行过程中发生的、需要进行调查的事件。

[来源: ISO/IEC/IEEE 29119-3:2021, 3.15]

3.4561

测试待查事件报告 test incident report

描述在测试期间发生要求进一步调查的事件的文档。

参见: 测试用例规格说明、测试项传输报告、测试日志、测试计划、测试规程和测试报告。

[来源: GB/T 11457-2006, 2.1708]_x000D_

3.4562

测试事件报告过程 test incident reporting process

动态测试过程的子过程。用于向利益相关方报告在测试执行过程中确定的、需要进一步处理的问题。

[来源: GB/T 38634.1-2020, 3.67]_x000D_

3.4563

测试独立性 test independence

执行测试人员与开发测试项的人员承担不同职责的程度。

[来源：ISO/IEC/IEEE 29119-1:2022, 3.106]

3.4564

测试接口 test interface

与测试项的接口，用于激励测试项和/或获取测试响应。

注1：典型的测试接口如 GUI、API 或 SOA 接口。

注2：激励测试项可能涉及通过计算机接口或附加硬件将数据传递到测试项中。

注3：获取响应包括从被测的测试项或其关联的硬件获取信息。

注4：典型的测试响应如实测结果。

[来源：GB/T 38634.5-2024, 3.13]

3.4565

测试接口层 test interface layer

关键字在最低层级上的抽象，直接与测试项交互，并在测试接口上封装了原子级（最低层级）的交互。

[来源：GB/T 38634.5-2024, 3.14]

3.4566

测试项 test item

作为测试对象的工作产品。

示例：系统、软件项、需求文档、设计规格说明、用户指南。

[来源：GB/T 38634.1-2020, 3.68, 有修改]_x000D_

3.4567

测试项传输报告 test item transmittal report

测试项标识文档。

注：包含当前状态和位置信息。

[来源：ISO/IEC/IEEE FDIS FDIS 24765, 3.4832, 有修改]

3.4568

测试项 test item

测试对象 test object

作为测试对象的工作产品。

示例：系统、软件项、需求文档、设计规格说明、用户指南。

[来源：GB/T 38634.1-2020, 3.68]

3.4569

测试负责人 test lead

领导整个测试过程的人。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4833]

3.4570

测试级别 test level

测试子过程的具体实例化。

示例：以下是常用的测试级别，可以实例化为测试子过程：组件测试级别/子过程，集成测试级别/子过程，系统测试级别/子过程，验收测试级别/子过程。

注：测试级别与测试阶段同义。

[来源：GB/T 38634.1-2020，3.69]_x000D_

3.4571

测试级别 test level

测试阶段 test phase

测试子过程的具体实例化。

示例：以下是常用的测试级别，可以实例化为测试子过程：组件测试级别/子过程，集成测试级别/子过程，系统测试级别/子过程，验收测试级别/子过程。

注：测试级别与测试阶段同义。

[来源：GB/T 38634.1-2020，3.69]

3.4572

测试日志 test log

测试执行的相关细节的时序记录。

[来源：ISO/IEC/IEEE FDIS FDIS 24765，3.4835，有修改]

3.4573

测试管理 test management

测试活动的策划、安排、预估、监测、报告、控制和完成。

[来源：GB/T 38634.1-2020，3.70]_x000D_

3.4574

测试管理流程 test management process

包含测试项目管理所需子过程的测试过程。

注：见测试策划过程、测试监测和控制过程、测试完成过程。

[来源：GB/T 38634.1-2020]

3.4575

测试模型 test model

在测试用例设计过程中使用的测试项的表示。

[来源：GB/T 38634.4-2020，4.27]_x000D_

3.4576

测试模型规格说明 test model specification

(1) 规定测试模型的文档。

[来源：ISO/IEC/IEEE 29119-2:2021，3.47]

(2) 规定测试模型的文档集。

[来源：ISO/IEC/IEEE 29119-3:2021，3.17]

[来源：<由总体组会议裁定>_x000D_]_x000D_

3.4577

测试监控过程 test monitoring and control process

测试管理过程的子过程。用以确保测试按照测试计划和组织级测试规格说明执行。

[来源：GB/T 38634.1-2020]

3.4578

测试目标 test objective

(1) 进行测试的原因。

[来源：ISO/IEC/IEEE 29119-2:2021，3.49]

(2)在特定条件下通过将实际行为与所需行为进行比较来衡量的一组已识别的软件特性。

[来源：ISO/IEC 25051:2014, 4.1.20, 有修改]

3.4579

测试预言 test oracle

用于判断测试是否通过的信息来源。

注：测试预言通常是用于为单个测试用例生成预期结果的规格说明，但也可使用其他来源，例如将实际结果与另一个类似程序或系统的结果进行比对，或由领域专家进行判定。

[来源：ISO/IEC TR 29119-11:2020, 3.1.76]

3.4580

测试预言问题 test oracle problem

针对给定的一组测试输入和状态，判断测试是否通过的挑战。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4843]

3.4581

测试组织 test organization

组织内部负责测试工作的管理结构。

注：测试组织通常在技术、管理和财务上独立于开发组织。

[来源：ISO/IEC/IEEE 29119-3:2021, 3.18]

3.4582

测试参与者 test participant

验证软件文档中的信息是否能正确完成任务，并形成对系统的概念性理解，提供反馈并配合数据收集的人员。

[来源：ISO/IEC/IEEE 26513:2017, 3.40]

3.4583

测试阶段 test phase

测试子过程的具体实例化。

[来源：GB/T 38634.1-2020, 3.74]

3.4584

测试计划 test plan

描述需要达到的测试目标以及实现该测试目标的方法和安排的文档，用于协调测试项的测试活动。

[来源：GB/T 38634.1-2020, 3.75, 有修改]

3.4585

测试策划过程 test planning process

测试管理过程的子过程。用于完成测试策划和开发测试计划。

[来源：GB/T 38634.1-2020, 3.76]

3.4586

测试方针 test policy, organizational test policy

组织进行测试的目的、目标、原则和整体范围。

注1：测试方针定义了执行什么测试，以及期望实现什么，但没有详细说明如何执行测试。

注2：测试方针可以为建立、评审和持续改进组织测试提供一个框架。

[来源：GB/T 38634.2-2020, 4.1.55]

3.4587

测试方针 test policy

组织级测试方针 **organizational test policy**

组织进行测试的目的、目标、原则和整体范围。

注 1：测试方针定义了执行什么测试，以及期望实现什么，但没有详细说明如何执行测试。

注 2：测试方针可以为建立、评审和持续改进组织测试提供一个框架。

[来源：GB/T 38634.2-2020，4.1.55]

3.4588

测试实践 test practice

便于实施测试的概念框架，该框架可应用于组织级测试过程、测试管理过程和/或动态测试过程。

[来源：GB/T 38634.1-2020，3.77]

3.4589

测试规程 test procedure

测试用例的执行序列，以及任何与构建初始前置条件所需的相关动作和执行后的收尾活动。

注：测试规程包括如何连续运行一个或多个测试用例的详细说明，包括设置通用的前提条件，为每个测试用例提供输入并评价实际结果。

[来源：GB/T 38634.1-2020，3.78]

3.4590

测试过程 test process

为一个软件产品提供质量信息的过程，通常由多个活动组成，分为一个或多个测试子过程。

注：特定项目的测试过程可能包含多个子过程，如系统测试子过程、测试计划子过程（较大测试管理过程的一部分）或静态测试子过程。

[来源：GB/T 38634.1-2020]

3.4591

测试规则 test protocol

测试中应遵循的步骤清单。

[来源：ISO/IEC/IEEE 26513:2017，3.41]

3.4592

测试就绪评审 test readiness review; TRR

为评估一个或多个配置项的初步测试结果；验证各配置项的测试规程完整、符合测试计划与测试说明、满足测试要求；并验证项目已具备开展配置项正式测试的条件而开展的评审。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS，3.4852，有修改]

3.4593

测试可重复性 test repeatability

测试的一种属性，表明每次执行该测试时均能产生相同的结果。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS，3.4853]

3.4594

测试报告 test report

描述对系统或部件进行测试产生的行为及结果的文件。

参见：测试用例规格说明 **test case specification**(2.1696)，测试待查事件报告 **test incident report**(2.1708)，测试项传输报告 **test item transmittal report**(2.1710)，测试日志 **test log**(2.1711)、测试计划 **test plan**(2.1714)和测试规程 **test procedure**(2.1715)。

[来源：GB/T 11457-2006，2.1719]

3.4595

测试需求 test requirement

组件或系统可测的方面，如作为测试依据的功能、事务、特征、质量属性或者结构元素。

注：测试条件常用来导出覆盖项，或由其本身构成覆盖项。

[来源：GB/T 38634.1-2020，4.52]

3.4596

测试结果 test result

指定的测试用例是否通过的标示，即观察到测试项输出的实测结果是否与预期结果一致或有偏差。

[来源：GB/T 38634.1-2020，3.82]

3.4597

测试场景 test scenario

作为生成测试用例基础的、针对测试项的场景或设定。

[来源：ISO/IEC TR 29119-11:2020，3.1.78]

3.4598

测试脚本 test script, test procedure specification

人工测试或自动化测试的测试规程规格说明。

[来源：GB/T 38634.1-2020，3.83]

3.4599

测试集 test set

一个或多个测试用例的集合，其执行时具有共同的约束。

示例：特定的测试环境、专业的领域知识或特定的目的。

[来源：GB/T 38634.1-2020，3.84]

3.4600

测试集体系结构 test set architecture

测试用例集之间的嵌套关系，该关系直接体现测试目标的层次化分解。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS，3.4858]

3.4601

测试规格说明 test specification

包含针对特定测试项的测试设计、测试用例和测试规程的全部文档集。

注：测试规格说明具体可以是一个文档、文档集合或以其他方式，例如文档和数据库条目的混合。

[来源：GB/T 38634.1-2020，3.85]

3.4602

测试规格说明技术 test specification technique

用于构建测试模型的活动、概念、过程和模式，该模型用于识别测试项的测试条件，导出相应的测试覆盖项，并导出或选择测试用例。

[来源：GB/T 38634.1-2020，3.59]

3.4603

测试状态报告 test status report

提供在指定报告期间所执行测试的状态信息的报告。

[来源：GB/T 38634.1-2020，3.86]

3.4604

测试策略 test strategy

测试计划的一部分。描述对特定测试项目或测试子过程进行测试的方法。

注 1：测试策略和组织级测试策略是不同的。

注 2：测试策略通常描述以下部分或全部：使用的测试实践、实现的测试子过程、采用的复测和回归测试、使用的测试设计技术和相应的测试完成准则、测试数据、测试环境和测试工具需求、测试交付物的预期。

[来源：GB/T 38634.1-2020，3.87]

3.4605

测试策略与计划过程 test strategy and planning process

用于设计测试策略、完成测试规划、创建并维护测试计划的测试管理过程。

[来源：ISO/IEC/IEEE 29119-2:2021，3.51]

3.4606

测试子过程 test sub-process

通常在测试项目的整体测试过程的周境中，用于执行特定的测试级别（例如系统测试、验收测试）或测试类型（例如易用性测试、性能测试）的测试管理和动态（和静态）测试过程。

注：测试子过程可以包含一个或多个测试类型。根据使用的生存周期模型，测试子过程也通常被称为测试阶段或测试任务。

[来源：GB/T 38634.1-2020，3.88]

3.4607

测试用例套 test suite

一组测试用例或测试程序。

注：测试套件的分组通常基于测试的执行时间。

[来源：ISO/IEC/IEEE 29119-1:2022，3.129]

3.4608

测试总结报告 test summary report

总结测试活动和结果的文件。它也包含相应测试项的评价。

[来源：ISO/IEC/IEEE 29119-2:2021，3.26]

3.4609

测试对象版本 test target version

用于执行一次动态测试或代码分析的测试对象的特定版本。

[来源：GB/T 41905-2022]

3.4610

测试技术 test technique

用于构建测试模型的活动、概念、过程和模式，该模型用于识别测试项的测试条件，导出相应的测试覆盖项，并导出或选择测试用例。

[来源：GB/T 38634.1-2020，3.59]

3.4611

测试追踪矩阵 test traceability matrix

用于识别文档集和软件中相关项（如测试相关的需求）的文档、电子表格或其他自动化工具。

注 1：测试追踪矩阵也称为验证交叉引用矩阵、需求测试矩阵、需求验证表等。

注 2：不同的测试追踪矩阵可能有不同的信息、格式和细节程度。

[来源：GB/T 38634.1-2020，3.90]

3.4612

测试追踪矩阵 test traceability matrix

需求测试矩阵 requirements test matrix

需求验证表 requirements verification table

验证交叉引用矩阵 verification cross reference matrix

用于识别文档集和软件中相关项（如测试相关的需求）的文档、电子表格或其他自动化工具。

注1：也称为验证交叉引用矩阵、需求测试矩阵,需求验证表等。

注2：不同的测试追踪矩阵可能有不同的信息、格式和细节程度。

[来源：GB/T 38634.1-2020, 3.90]

3.4613

测试类型 test type

一组专注于特定质量特性的测试活动。

示例：信息安全性测试、功能性测试、易用性测试和性能测试。

注：一种测试类型可以在单个测试子过程中执行或跨多个测试子过程执行（性能测试在组件测试子过程中完成，也在系统测试子过程中完成）。

[来源：GB/T 38634.1-2020, 3.91]

3.4614

测试单元 test unit

一个或多个计算机程序模块与相关的控制数据（例如，表）、用法规程和操作规程一起的集合，这些集合满足下列条件：a)所有模块来自单个计算机程序、b)在集中至少有一个新的或更改的模块未完成单元测试、c)模块与它相关的数据和规程是测试过程的唯一目标。

[来源：GB/T 11457-2006, 2.1724]

3.4615

可测试性 testability

产品所具备的、可设计并执行客观且可行的测试以判定是否满足某项要求的能力。

[来源：ISO/IEC 25010:2023, 3.7.5]

3.4616

测试驱动的开发 test-driven development; TDD

2. 在工作开始前定义测试，以便持续验证进行中的工作，从而以零缺陷思维开展工作的技术。

[来源：ISO/IEC 29110-1-2:2024, 3.113]

3.4617

测试 testing

为发现和/或评价一个或多个测试项的属性而进行的一系列活动。

注：测试活动可包括针对测试的计划、准备、执行、报告和管理活动，其均与测试直接相关。

[来源：GB/T 38634.1-2020, 3.92]

3.4618

测试说明 testing description

对测试执行条件的描述（即测试规程）。

[来源：ISO/IEC 25051:2014, 4.11]

3.4619

测试金字塔 testing pyramid

一系列的分级测试，包含较多的简单且自动化的测试（单元测试），频率较低的集成测试，较少且耗时较长的端到端测试或手动测试。

[来源：ISO/IEC 33202:2024, 3.26]

3.4620

测试任务迭代 testing task iteration

在开发阶段最初执行后，于维护阶段重新执行的测试任务。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4874]

3.4621

测试工具 testing tool

用于测试执行和测试管理的特定或通用工具，如测试结果记录、测试结果显示、测试结果解释、测试数据生成、测试步骤生成、测试脚本生成、测试建模等。

[来源：GB/T 41905-2022]

3.4622

测试有效性 test-validity

完成测试规定目标的程度。

[来源：GB/T 11457:2006, X.X]

3.4623

测试件 testware

计划、设计和执行测试所需的、在测试过程中产生的制品。

注：测试件可以包括文档、脚本文件、输入、预期结果、文件、数据库、环境以及任何用于测试过程中附加的软件或设备。

[来源：GB/T 38634.1-2020, 3.93]

3.4624

文本 text

以字符、符号、字、短语、段落、语句、表格或其他字符排列形式呈现、旨在传递含义的数据，其解读主要依赖读者对某种自然语言或人工语言的掌握。

[来源：ISO/IEC 2382:2015]

3.4625

文本编辑程序 text editor editor

一种计算机程序，它通常作为字处理系统的一部分，允许用户输入、替换和浏览文本。

[来源：ISO/IEC/IEEE 24765:2017《系统和软件工程 术语》]

3.4626

文本页 text page

包含与特定图表相关文字内容的模型页。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4879]

3.4627

文本处理 text processing

对文本进行的数据处理操作，例如输入、编辑、合并、检索、存储、显示或打印。

3.4628 [[来源：ISO/IEC 2382:2015, X.X]

文本处理 / 文档处理 text processing; word processing

对文本进行数据处理操作，例如输入、编辑、合并、检索、存储、显示或打印。

3.4629 [[来源：ISO/IEC 2382:2015, X.X]

文本处理文字处理 text processing word processing

对文本的数据处理操作，如输入、编辑、合并、检索、存储、显示或打印。

[来源：ISO/IEC 2382:2015, 《信息技术—术语》]

3.4630

构造 texture, architectural texture

架构性构造 architectural texture。

实现产品线应用的通用开发规则和约束条件的集合。

[来源: GB/T 45804-2025]

3.4631

构造 texture

架构性构造 architectural texture

为实现产品线应用而定义的通用开发规则与约束的集合。

[来源: ISO/IEC 26551:2016 《软件与系统工程 — 产品线需求工程的工具与方法》，3.19)]

[架构]针对产品线架构的共性与可变部分而制定的通用开发规则、指南与约束的集合。

[来源: ISO/IEC 26552:2019 《软件与系统工程 — 产品线架构设计的工具与方法》，3.2]

3.4632

主题 theme

由共同因素（如功能、数据源、安全级别）相互关联的用户故事集合。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4882]

3.4633

思考时间 think time

交互系统发出提示或消息结束，至人类用户开始响应之间的经过时间。

参见：端口间时间、响应时间、周转时间。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4883]

3.4634

第三范式 third normal form

一种规范化过程的结果：该过程对数据分组进行转换，使每个非键属性不依赖于任何其他非键属性。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4884]

3.4635

系统颠簸 thrashing

计算机系统的一种状态：系统将大部分或全部资源消耗在开销操作（如主存与辅存之间的数据换页 / 交换）上，而非执行预期的计算功能。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4886]

3.4636

线程 thread

由一系列动作构成的动作链，其中至少有一个对象参与该链中的全部动作。

[来源: ISO/IEC 10746-1:1998, 13.1.2]

3.4637

[控制的]线程 thread [of control]

贯穿某一程序、动态模型或其他控制流中表示的一种单一执行路径。也是作为轻量级进程实现的主对象的一种衍型。

参见：进程

[来源: GB/T 11457:2006, X.X]

3.4638

威胁建模 threat modeling

一种系统性探查技术，用于识别任何可能以破坏、泄露、数据篡改或拒绝服务的形式对系统造成危害的情形或事件。

注：该活动最终形成脆弱性评估结果。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4890]

3.4639

三地址指令 **three-address instruction**

包含三个地址字段的计算机指令。

参见：一地址指令、二地址指令、四地址指令、零地址指令。

示例：将存储单元 A 与 B 的内容相加，并将结果存入存储单元 C 的指令。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4891]

3.4640

三加一地址指令 **three-plus-one address instruction**

包含四个地址字段的计算机指令，其第四个地址字段包含下一条要执行的指令的地址。例如，一指令加单元 A 和 B 的内容，把结果放至单元 C，然后执行在单元 D 中的指令。

相对：加一地址指令 **one-plus-one address instruction** (2.1050)，二加一地址指令 **two-plus-one address instruction** (2.1780) 和四加地址指令 **four-plus-one address instruction** (2.653)。

[来源：GB/T 11457-2006, 2.1734]

3.4641

阈值 **threshold**

可测量项目变量的预定值，该值代表一个限值，一旦达到即需采取相应措施。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4893]

3.4642

阈值覆盖 **threshold coverage**

(1) 在一组测试中，激活值超过阈值的神经元数量与神经网络中神经元总数的比值（通常以百分比表示）。

注：选取介于 0 与 1 之间的激活阈值作为该阈值。

[来源：ISO/IEC TR 29119-11:2020 《软件与系统工程 — 软件测试 第 11 部分：基于人工智能的系统测试指南》，3.1.79]

3.4643

吞吐量 **throughput**

对计算机系统或部件在给定的时间周期内规定执行的工作总量的度量。例如，每天的作业数。

参见：往返时间 **turnaround time** (2.1775) 和工作负载模型 **workload model** (2.1857)。

[来源：GB/T 11457-2006, 2.1735]

3.4644

吞吐量图表 **throughput chart**

显示随时间推移的可交付成果已接受情况的图表。

[来源：项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4645

吞吐率 **throughput rate**

系统在单位时间内可处理的样本数量。

注：吞吐率取决于系统特性和样本特性。

[来源：ISO/IEC TR 29119-13:2022, 3.1.32]

3.4646

吞吐量评级值 throughput rating value

实际吞吐量与吞吐量参考值的商。

[来源: ISO/IEC 14756:1999, 4.25]

3.4647

吞吐量参考值 throughput reference value

一组模拟用户所要求的最小吞吐量。

[来源: ISO/IEC 14756:1999, 4.26]

3.4648

缩略图 thumbnail

用于快速识别较大图像或视频文件的微型图像文件。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.22]

3.4649

节拍 tick

绝对时间或相对时间中的一个时间点, 或被计为自然单位的一次事件发生。

[来源: IEEE 982:2024, 3.1]

3.4650

排列图 tier chart

排列图 (tier chart): 按一定顺序排列的图表, 用于展示数据的分布或优先级等情况, 通常以条形图形式呈现, 便于识别关键因素或主要部分。

参见: 结构图 (structure chart)、控制流图 (control flow diagram)、数据流图 (data flow diagram)、数据结构图 (data structure diagram)、状态图 (state diagram)。

注: 其呈现结果不一定与结构图一致。

3.4651

工料合同 time and materials contract; T&M contract

一种混合型合同, 兼具成本补偿型合同和固定价格合同的特征。

注: 协议的总价值基于工作小时数和所供材料, 在授予合同时并未确定。但协议中设定了人工和材料的单位费率。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4652

时间行为 time behaviour

产品在指定条件下执行其指定功能的能力, 且响应时间和吞吐率满足要求。

[来源: ISO/IEC 25010:2023, 3.2.1]

3.4653

时间类别 time class

一种时间限制, 并配有一个相对频率; 该频率是指 (某特定任务类型中) 执行时间小于或等于该时间限制的任务数量, 与该任务类型总任务数量的比值。用于与该任务类型的任务执行时间进行比较。

[来源: ISO/IEC 14756:1999 《信息技术 —— 基于计算机的软件系统性能的测量与评定》, 4.27]

3.4654

用时事件 time event

用时事件 (time event): 在系统工程中, 指与时间相关的事件, 用于触发或控制系统的特定行为或状态变化, 例如定时事件、超时事件等。(来源: 11457.0)。

3.4655

时间表达 time expression

一种决定时间的绝对值或相对值的表达式。

3.4656

超时 time out

当达到预先确定的消逝的时间总数而未发生预料的事件时产生的条件。例如，若在规定的周期内未接收到用户的输入，引起联机进程终止的条件。

[来源：ISO/IEC/IEEE 24765:2017《系统和软件工程 术语》]

3.4657

时间 Petri 网 time Petri net

(1) 时间约束与转换相关的 Petri 网。

[来源：ISO/IEC 15909-1:2019《系统和软件工程 — 高级 Petri 网 第 1 部分：概念、定义和图形表示法》，3.36]

3.4658

分时 time sharing

一种操作方式，它用程序的交叉执行，允许两个或多个用户在同一计算机系统上并发执行计算机程序。

注 1：分时可以用时间片、基于优先权的中断或其他调度方法实现。

注 2：分时可通过时间片、基于优先级的中断或其他调度方法实现。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4659

时间分片 time slicing

一种操作方式，其中的两个或多个进程，每一个都赋予一小的、在同一处理器上固定的连续处理时间，而处理器以轮转方式执行，每一个执行为它分配的时间，直至全部完成。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4660

时间盒 timebox

为项目任务规定的时长上限。

[来源：ISO/IEC 29110-1-2:2024, 3.114]

3.4661

时间关键任务 time-critical task

具有明确截止期限的任务。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4662

及时性 timeliness

IT 服务在规定时限内交付结果的程度。

[来源：ISO/IEC TS 25011:2017, 3.2.6.1]

3.4663

时效功能 timeliness function

针对特定任务类型，对任务执行时间相关的用户需求的描述。

注：它由一个或多个时间类别组成。

[来源：ISO/IEC 14756:1999, 4.29]

3.4664

及时性评价值 **timeliness rating value**

及时吞吐量与总吞吐量的商（对应第 j 个任务类型）。

[来源：ISO/IEC 14756:1999, 4.28]

3.4665

及时吞吐量 **timely throughput**

所有执行时间在及时性函数约束下可被接受的任务的吞吐量。

[来源：ISO/IEC 14756:1999, 4.3]

3.4666

定时器事件 **timer event**

用于周期性激活任务的激励信号。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4667

定时脉冲单元 **timer pulse unit; TPU**

用于生成定时脉冲的微控制器单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4920]

3.4668

时标进度网络图 **time-scaled schedule network diagram**

一种项目进度网络图，其进度活动的位置与长度代表该活动的持续时间。

注：本质上，它是包含进度网络逻辑的横道图（条形图）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4913]

3.4669

计时 **timing**

估计或测量软件系统或部件要求的总执行时间的过程。

参见：规模估算（sizing）。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4670

计时分析程序 **timing analyser**

一种软件工具，通过累加指定路径上指令的执行时间，或在程序指定位置插入探测点并测量探测点间的执行时间，来估算或测量计算机程序（或程序片段）的执行时间。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4924]

3.4671

时序分析 **timing analysis**

用于评估逻辑信号的时间序列或数字电路速度的方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4923]

3.4672

时序图 **timing diagram**

展示一组任务按时间顺序执行序列的图表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4925]

3.4673

电信信息网络体系结构 **TINA; TINA**

电信信息网络体系结构。

[来源: ISO/IEC 10746-1:1998]

3.4674

自动构建与回归测试工具 tinderbox

一种自动化构建与回归测试工具。

注: 该工具通常定期从各支持分支获取软件最新版本, 针对不同平台执行构建, 并反馈构建与回归测试结果。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4927]

3.4675

完工尚需绩效指数 to-complete performance index; TCPI

完成剩余工作所需成本与剩余预算的比率。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4932]

3.4676

评估目标 TOE

评估目标。

[来源: ISO/IEC 23643:2020, 3.26]

3.4677

令牌 token

与位置相关联、且从该位置的类型中选取的数据项。

[来源: ISO/IEC 15909-1:2004, 2.1.25]

终结符。

[来源: ISO/IEC 15475-2:2002, SYNTAX.1, 6.1]

3.4678

可接受风险 tolerable risk

在给定情境下, 基于当前社会价值观所接受的风险水平。

[来源: ISO/IEC/IEEE 15026-1:2019, 3.3.15]

3.4679

容差 tolerance

对质量要求的可接受偏差的量化描述。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4937]

3.4680

工具 tool

为软件和系统生存周期过程提供支持的软件产品。

[来源: ISO/IEC 154741:2002, 4.2]

3.4681

工具桥接器 tool bridge

将关键字映射和转换为可由测试执行引擎执行的代码和数据的组件。

[来源: ISO/IEC/IEEE 29119-5:2024, 3.28]

3.4682

工具箱 toolbox

在能力上相互补充, 以覆盖其预期用途更大范围的一组工具。

参见: 工具套件 (tool suite)。

[来源: ISO/IEC 23643:2020 《软件和系统工程 — 软件安全与安保验证工具的能力》, 3.29]

3.4683

工具专用信息 tool-specific information

与网图对象或网图本身相关、专属于某一特定工具且不供其他工具使用的信息。

[来源: ISO/IEC 15909-2:2011, 4.1.19]

3.4684

顶级 top level

指称包容层次结构中最顶的包的一种包的衍型。顶级衍型在命名空间向外“看”时,定义查阅名称的外层界限。例如:顶级子系统表示子系统包容层次结构的顶部。(来源:11457.0)。

3.4685

自顶向下的 top-down

用于说明从层次的最高级部件处开始,逐步推进到较低级的活动。例如,自顶向下设计、自顶向下程序设计和自顶向下测试。

参见:关键部分优先 critical piece first(2.368)。

相对:自底向上 bottom-up(2.153)。

[来源: GB/T 11457-2006, 2.1746]

3.4686

自顶向下设计 top-down design, top-down decomposition

通过识别系统主要组件,将其分解为低层组件,并迭代细化直至达到所需详细程度的系统设计过程。

[来源: ISO/IEC/IEEE FDIS 24765 FDIS, 3.4944]

3.4687

自上而下的设计 top-down design**自上而下的分解 top-down decomposition**

将系统的功能从高级概念分解为低级部分的设计方法。

2. 通过识别主要组件、将它们分解为低级组件并迭代直到所需级别的系统设计过程实现细节。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4944]

3.4688

自上而下的测试 top-down testing

自顶向下测试(top-down testing):从系统的顶层开始,逐步向下进行测试,先测试上层模块,再逐步测试下层模块,通常使用驱动模块来模拟下层模块的调用。

[来源: 现有11457定义]

3.4689

主题 topic

可与一项或多项信息相关联的唯一标识或标识符。

[来源: ISO/IEC/IEEE 26531:2023, 3.1.31]

3.4690

龙卷风图 tornado diagram

敏感性分析中用于比较变量相对重要性的一种特殊条形图[项目管理知识体系指南(PMBOK®指南)第五版]。

3.4691

完全映射 total

完全映射。

参见: partial(部分的)、mandatory(强制的)、mapping completeness(映射完备性)。

注：从集合 D 到集合 R 的映射 M 是“完全（全域）的，当且仅当对于 D 中的每一个 X，在 R 中至少存在一个 Y，使得序偶 [X, Y] 属于 M。一个类的属性是完全（全域）”的，指该类的每一个实例都具有该属性的值，除非被显式声明为部分（partial）。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4946]

3.4692

完全簇 / 完备簇 total cluster complete cluster

一种子类簇，其中超类的每个实例至少是该簇中某一个子类的实例。

参见：非完备簇（incomplete cluster）、部分簇（partial cluster）、超类（superclass）。

[来源：IEEE 1320.2-1998 (R2004)《IDEF1X97 (IDEFobject) 概念建模语言语法和语义 IEEE 标准》，3.1.197]

3.4693

总簇 total cluster, complete cluster

一种子类簇，其中超类的每个实例都是该簇中至少一个子类的实例。

参见：不完备簇、部分簇、超类。

[来源：ISO/IEC/IEEE FDIS 24765 FDIS, 3.4947]

3.4694

全簇 total cluster

完备簇 complete cluster

子类集群，其中超类的每个实例均为集群子类中至少一个子类的实例。

参见不完全簇、部分簇、超类。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4947]

3.4695

完全正确性 total correctness

在正确性证明中，一种标识，表明程序的输出断言可从其输入断言和处理步骤逻辑推导得出。

参见：部分正确性。

[来源 ISO/IEC/IEEE FDIS 24765 FDIS, 3.4948]。

3.4696

全面质量管理 total quality management; TQM

所有生存周期阶段质量改进的整体方法。

[来源：ISO/IEC/IEEE 24765:2017, X.X]

3.4697

传输轮廓 T-profile

传输轮廓。

[来源：ISO/IEC 10746-1:1998：概述]

3.4698

全面质量管理 TQM

与 1113 重复。

3.4699

威胁与风险评估 TRA

威胁与风险评估。

[来源：IEEE 1012-2024, 3.2]

3.4700

踪迹 trace

a) 计算机程序执行的记录，它显示执行的指令的顺序、变量的名和值或两者。类型包括执行踪迹、回顾的踪迹、子例程踪迹、符号踪迹、变量踪迹。b) 如像 a) 产生的记录。c) 建立在两个或多个开发过程的产品之间的关系。例如，建立在给定的需求和实现此需求的设计元素之间的关系。

[来源：GB/T 11457-2006, 2.1751]

3.4701

跟踪链接 trace link

两个追踪制品之间，或可变性模型元素与开发制品之间的关联。

[来源：ISO/IEC 26559:2017《系统和软件工程 — 软件与系统产品线中可变性追踪的方法和工具》，

3.6]

3.4702

跟踪链接语义 trace link semantics

追踪链接的用途或含义，在追踪链接类型中予以规定。

[来源：ISO/IEC 26559:2017《软件工程与系统工程 —— 软件和系统产品线中可变性追踪的方法与工具》，3.7]

3.4703

可追踪性 traceability

两个或多个逻辑实体（如需求、系统元素、验证或任务）之间可辨识的关联。

[来源：ISO/IEC/IEEE 15288:2023, 3.52]

3.4704

测试追踪矩阵 traceability matrix

记录开发过程中两个或多个产品之间关系的矩阵。

示例：记录给定软件组件的需求和设计之间关系的矩阵。

[来源：ISO/IEC/IEEE 24765:2017《系统和软件工程 术语》]

3.4705

具有不同抽象级别的可追溯性 traceability with different abstraction level

既定可追溯性的详细程度。

[来源：ISO/IEC 26559:2017 软件和系统工程—软件和系统产品线可变性可追溯性的方法和工具，3.5]

3.4706

可追溯 traceable

具有可确定来源的组件。

[来源：ISO/IEC/IEEE 15289:2015 系统和软件工程 生存周期信息产品的内容（文档），5.25]

3.4707

追踪程序 tracer

用于追踪的软件工具。

[来源：11457.0]

3.4708

权衡研究 trade study

基于准则和系统分析评估备选方案，以选出能实现既定目标的最佳方案。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4967]

3.4709

权衡 trade-off

根据利益相关方的净效益，从各种需求和备选解决方案中选择决策行动。

[来源：GB/T 45802-2025; ISO/IEC/IEEE 12207:2017, 3.1.68]

3.4710

权衡分析 trade-off analysis

对照性能、目标成本设计及全生存周期质量要素，对设计方案 / 备选方案开展的分析评估。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4970]

3.4711

尾部 trailer

置于文件或报文末尾的标识或控制信息。

参见：头部（header）。

[来源：ISO/IEC/IEEE 24765:2017 《》]

3.4712

尾随判定 trailing decision

在循环体之后执行的循环控制。

参见：“直至”语句 UNTIL(2.1795)。

相对：先导判定 leading decision(2.808)。

[来源：GB/T 11457-2006, 2.1756]

3.4713

训练数据 training data

(1) 用于训练机器学习模型的数据集。

[来源：ISO/IEC TR 29119-11:2020 《软件工程与系统工程 — 软件测试 第 11 部分：基于人工智能系统的测试指南》，3.1.80]

3.4714

训练组 training group

负责为组织协调和安排培训活动的人员的集合（包括经理和职员）。这个组准备和讲授大多数的培训课程，并协调其他培训工具的使用。

[来源：11457.0]

3.4715

培训计划 training program

重点阐述组织的培训需求的一组相关元素。它包括组织的培训计划、培训教材、培训的开发、培训的执行、培训设施、培训的评价以及培训记录的维护。

[来源：现有 11457 定义]

3.4716

事务 transaction

软件工程中，指能够引起、触发或启动一个动作或一系列动作的数据单元、控制单元、信号、事件或状态变更。

导致一组对象变更与动态模式（及其约束性不变模式）保持一致的活动。

[来源：ISO/IEC 10746-3:2009, 13.7.1.1]

3.4717

事务分析 transaction analysis, transaction-centred design

一种软件开发技术，在这种技术中，系统的结构来自系统要求处理的事务的分析。

参见：以数据结构为中心的设计、输入-处理-输出、模块分解、面向对象的设计、快速原型、逐步细化、结构设计、转换分析

[来源：GB/T 11457-2006，2.1760]

3.4718

事务分析 transaction analysis

以事务为中心的设计 transaction-centered design

软件开发技术，其中系统的结构源自分析系统处理所需的事务。

参见：以数据结构为中心的设计、输入-处理-输出、模块化分解、面向对象设计、快速原型法、逐步求精、结构化设计、变换分析

[来源：ISO/IEC/IEEE FDIS 24765，3.4977]

3.4719

事务文件 transaction file

用于收集近期数据变更，并定期更新主文件的文件。

注：仅被读取一次，其数据即被使用。

[来源：ISO/IEC/IEEE FDIS 24765，3.4978]

3.4720

事务矩阵 transaction matrix

一种矩阵，用于标识数据库访问的可能请求，并将每个请求与数据库中的信息类别或信息项相关联。

[来源：ISO/IEC/IEEE FDIS 24765，3.4979]

3.4721

事务模式 transaction schema

定义事务及其依赖关系的动态模式和不变模式。

[来源：ISO/IEC 10746-3:2009，16.8.1.1]

3.4722

事务透明性 transaction transparency

一种分布透明性，它屏蔽对象配置之间的活动协调，以实现一致性。

[来源：ISO/IEC 10746-3:2009，4.4.1.8]

3.4723

事务功能 transactional function

提供给用户用以处理数据功能的基本过程。

注：事务功能是外部输入、外部输出和外部查询。

[来源：GB/T 42449-2023，X.X]

3.4724

事务功能类型 transactional function type

NESMA FPA 方法分配给事务功能的三个类别之一：外部输入、外部输出和外部查询。

[来源：GB/T 42588-2023，X.X]

3.4725

事务中心设计 transaction-centered design

一种软件开发技术，通过分析系统所需处理的事务来推导系统的结构。

参见：面向数据结构设计、输入-处理-输出、模块化分解、面向对象设计、快速原型法、逐步求精、结构化设计、变换分析、以事务为中心的设计

[来源: ISO/IEC/IEEE 24765:2017, 3.4363]

3.4726

传送 transfer

转移

将数据从一处发送并在另一处接收。

参见: 跳转

[来源: ISO/IEC/IEEE FDIS 24765, 3.4983]

3.4727

传输文件 transfer file

包含待交换数据的文件。

注: 由一个头部和若干组件构成。组件包含数据或数据定义数据。

[来源: ISO/IEC 15474-1:2002, 4.1]

3.4728

权利转移 transfer of an entitlement

将既定权利授予独立法人实体的过程。

注: 当大型机构将其部分业务剥离为独立法人实体时, 通常会发生权利转移。该权利模式支持对权利转移进行记录。

转移须遵循软件许可方与最终用户之间的合同条款与条件。

[来源: ISO/IEC 19770-3:2016, 3.1.41]

3.4729

转换分析 transform analysis; transform-centred design

转换中心设计 transformation analysis

一种软件开发技术: 通过分析流经系统的数据流及必须对数据执行的转换处理, 推导系统结构。

参见: 面向数据结构设计 (data structure-centered design)、输入-处理-输出 (input-process-output)、模块化分解 (modular decomposition)、面向对象设计 (object-oriented design)、快速原型法 (rapid prototyping)、逐步求精 (stepwise refinement)、结构化设计 (structured design)、事务分析 (transaction analysis)。

同义词: 转换分析 (transformation analysis)、面向转换设计 (transform-centered design)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4986]

3.4730

转换分析 transform analysis

变换分析 transformation analysis

以转换为中心的设计 transform-centered design

一种软件开发技术: 通过分析流经系统的数据流及必须对数据执行的转换处理, 推导系统结构。

参见: 面向数据结构设计 (data structure-centered design)、输入-处理-输出 (input-process-output)、模块化分解 (modular decomposition)、面向对象设计 (object-oriented design)、快速原型法 (rapid prototyping)、逐步求精 (stepwise refinement)、结构化设计 (structured design)、事务分析 (transaction analysis)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4986]

3.4731

以转换为为中心的设计 transform-centered design

一种软件开发技术: 通过分析流经系统的数据流及必须对数据执行的转换处理, 推导系统结构。

参见: 面向数据结构设计 (data structure-centered design)、输入-处理-输出 (input-process-output)。

output)、模块化分解(modular decomposition)、面向对象设计(object-oriented design)、快速原型法(rapid prototyping)、逐步求精(stepwise refinement)、结构化设计(structured design)、事务分析(transaction analysis)。

[来源: ISO/IEC/IEEE 24765:2017, 3.4373]

3.4732

瞬态错误 transient error

仅出现一次或以不可预测的时间间隔发生的错误。

参见: 间歇性故障(intermittent fault)、随机失效(random failure)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.4987]

3.4733

瞬时对象 transient object

一种仅在创建它的进程或线程的执行期间存在的对象。

[来源: GB/T 11457-2006, 2.1768]

3.4734

转移 transition

涉及将新的或变更的服务、系统或组件移入或移出环境的活动。

[来源: ISO/IEC/IEEE 14764:2021, 3.1.14]

3.4735

转移发生转移规则 transition occurrence transition rule

若某变迁在某模式下具备使能条件, 则该变迁可在此模式下触发。

注1: 变迁触发时, 将原子不可分地执行以下动作:

对该变迁的每一个输入库所, 从输入库所的标识中, 扣除该模式下输入弧对应的使能托肯;

对该变迁的每一个输出库所, 将经求值后的输出弧表达式对应的托肯多重集, 加入输出库所的标识中。

一个库所可同时作为同一变迁的输入库所与输出库所。

[来源: ISO/IEC/IEEE 24765:2017, 3.4378]。

3.4736

过渡场景 transition scenario

切换场景

迁移场景

规定产品线迁移策略部署过程中, 相关主体、实施内容、流程顺序、角色与职责履行方式的场景。

[来源: ISO/IEC 26562:2019 3.6]

3.4737

过渡变量 transition variables

出现在与变迁相关联表达式中的变量。

注: 这类变量包含变迁条件以及变迁周边各弧的标注。

[来源: ISO/IEC/IEEE 24765:2017, 3.4379]

3.4738

翻译程序 translator

转换程序

一种计算机程序, 它把一种语言表达的语句序列变换为另一种语言表达的等价的语句序列。

参见: 2 汇编程序(assembly), 编译程序(compiler)。

[来源: GB/T 11457-2006, 2.1760, 有修改]

3.4739

透明度 transparency

向利益相关方传递信息的特性：信息真实，包含与行为、决策或动因相关的内容，并以利益相关方可理解的技术程度与形式呈现。

注：组织信息系统向用户披露其处理信息及内部运行流程时，即具备透明性。在需求工程中，透明性通常视为非功能需求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4993]

3.4740

陷入 trap

条件跳转至异常或中断处理例程，此过程通常由硬件自动触发，并会记录跳转的起始位置。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4994]

3.4741

树状图 tree diagram

用于将一组系统规则可视化父子关系的分解层次结构图。

[来源：ISO/IEC/IEEE 24765-2025 fdis, 3.4995]

3.4742

树结构图 tree-structured chart

描述 ISO/IEC 8631 中定义的程序结构并具有树结构的图表。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4996]

3.4743

趋势分析 trend analysis

基于历史结果，运用数学模型预测未来结果的分析方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4997]

注：趋势分析是确定预算、成本、进度或范围参数相对基准偏离程度的一种方法。它利用以往各进度报告期的数据，在项目执行不做任何调整的前提下，预测该参数在项目未来某一时间点相对于基准的偏差。

3.4744

触发条件 trigger condition

表明风险即将发生的事件或情况。[项目管理知识体系指南（PMBOK®指南）第五版]。

3.4745

触发事件 triggering event

能够引发软件功能用户启动（触发）一个或多个功能过程的事件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4998]

3.4746

触发事件 triggering event

触发事件类型 triggering event type

能够引发软件功能用户启动（触发）一个或多个功能过程的事件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.4998]

3.4747

风险触发器 triggers

表明风险已发生或即将发生的迹象。触发因素可在风险识别过程中发现，并在风险监控过程中予以监测。触发因素有时亦称风险征兆或预警信号。

[来源：IEEE_CS_SE_Vocabulary-2025]

3.4748

三重底线 triple bottom line

通过从利润、人员和地球环境的角度评估公司的底线来考虑经营的全部成本的框架。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4749

三重约束 triple constraint

评估竞争需求的框架，例如进度、成本和质量。

注：三重约束通常被描述为一个三角形，其中一个边或一个角代表项目团队管理的参数之一。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5000]

3.4750

真反例 true negative

当故障确实存在时，应正确报告故障。

参见：假阴性（false negative）、真阳性（true positive）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5002]

3.4751

真正例 true positive

正确报告通过时通过。

参见：假阳性（false positive）、真阴性（true negative）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5003]

3.4752

主干线 trunk

中继线

软件的开发主线；大多数分支的主要起点。

注：通常可以通过用于标识其文件的版本号将主干与其他分支区分开来，该版本号比所有其他分支的版本号短。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5004]

3.4753

可信度 trustworthiness

表征人们对系统可靠程度信任度的质量特性。

注：对可信度有贡献的方面包括：可靠性、可用性、弹性、安全性、隐私性、问责性、透明度、完整性、真实性、质量和可用性。

[来源：IEEE 982:2024, 3.1]

3.4754

可信原则 trustworthiness principle

由一组程序活动产生的证据来支持的，用于保证可靠的通用要求。

[来源：GB/T 37970-2019]

3.4755

值得信赖的数据 trustworthy data

为完成任务所需，向授权用户提供的准确、完整、相关、易于理解且可获取的数据及相关信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5006]

3.4756

可信过程域 trustworthy process area

以可信保护目标为导向的一类可信原则。

[来源: GB/T 37970-2019]

3.4757

T 恤尺码估算法 T-shirt sizing

通过比较用户故事并将其划分为特小、小、中、大和特大等级别来开展相对估算的方法。

注: T 恤尺码估算法明确了相互关系, 避免了在虚假精度上浪费时间。

[来源: ISO/IEC TR 7052:2023, 3.47]

3.4758

隧道符号 tunnel notation

一对短浅的弧线, 形似左右圆括号, 用于括住箭头线段的箭首或箭尾。

[来源: ISO/IEC/IEEE 24765m:2024, 2.1.107]

3.4759

穿透箭头 tunneled arrow

从祖先功能框的连接点到其作为边界箭头出现在某直接下级子图之间的未绘制箭头段。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5009, 有修改]

3.4760

隧道 tunneling

对箭头线段施加隧道表示法的操作。

[来源: ISO/IEC/IEEE 24765m:2024, 2.1.106]

3.4761

元组 tuple

一组字段或数据项。

注: 可以使用元组代替记录。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5011]

3.4762

往返时间 turnaround time

在作业的提交至批处理系统和完全的输出返回之间消逝的时间。

参见: 2.1148 条 端口至端口时间 (port-to-port time)、2.1377 条 响应时间 (response time) 和 2.1729 条 思考时间 (think time)。

[来源: GB/T 11457-2006, 2.1775]

3.4763

交钥匙的 turnkey

指硬件或软件系统以完整、可直接运行的状态交付。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5013]

3.4764

交钥匙系统 turnkey system

数据处理系统, 安装后即可供用户使用, 处于可运行状态, 可能针对特定用户或应用进行定制。

注: 用户数据可能需要进行某些准备工作。

[来源: ISO/IEC 2382:2015 信息技术——术语]

3.4765

操作实训 tutorial

用户使用随软件或文档提供的示例数据练习软件功能的指导程序。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5015, 有修改]

3.4766

两地址指令 two-address instruction

包含两个地址字段的计算机指令。

同义词：双操作数指令。

参见：单地址指令、三地址指令、四地址指令、零地址指令。

[来源:GB/T 11457-2017, 2.1777]

3.4767

二地址指令 two-address instruction

双操作数指令 double-operand instruction

包含两个地址字段的计算机指令。

参见：单地址指令（one-address instruction）、三地址指令（three-address instruction）、四地址指令（four-address instruction）、零地址指令（zero-address instruction）。

示例：将 A 的内容添加到 B 的内容的指令。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5016]

3.4768

两级地址 two-level address

一种间接地址，它规定包含所希望的操作数的地址的存储单元。

参见：2.1007 条 n 级地址（n-level address）。

[来源:GB/T 11457-2006, 2.1778]

3.4769

两级编码 two-level encoding

一种微程序技术，在此种技术中，不同的微操作同时编码在微指令的同一字段中，且一个执行依赖于在微指令内部或外部的另一字段的值。

[来源:GB/T 11457-2006, 2.1779]

3.4770

两阶段获得 two-phase acquisition

一种项目管理方法，将项目划分为两个明确阶段：早期阶段聚焦需求获取、关键风险应对及项目规划；后期阶段仅在早期阶段成果满足预期时启动全面实施。

注：最终是否推进完整项目的决策，从不确定性最高的立项阶段推迟至不确定性显著降低的后续阶段。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5019]

3.4771

二加一地址指令 two-plus-one address instruction

包含三个地址字段，第三个包含下一条要执行的指令的地址的计算机指令。例如，把 A 单元的内容加至 B 单元的内容上，然后执行在 C 单元的指令。

参见：2.1050 条 一加一地址指令（one-plus-one address instruction）、2.1734 条 三加一地址指令（three-plus-one address instruction）和 2.653 条 四加一地址指令（four-plus-one address instruction）相对。

[来源:GB/T 11457-2006, 2.1780]

3.4772

类型 type

一个可标识实体，其关联的谓词是一个定义在实体上的单参数布尔函数。

示例：“类”。

[来源:ISO/IEC 19500-1:2012, 5.3.4, 有修改]

3.4773

类型表达式 **type expression**

一种求出对一个或多个类型的引用的表达式。

[来源:GB/T 11457-2006, 2.1782]

3.4774

上层架构 **ULA**

上层体系架构。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5030]

3.4775

解绑行为 **unbinding behavior**

终止绑定的行为。

[来源:ISO/IEC/IEEE 24765:2017, 3.4411]

3.4776

解除绑定行为 **unbinding behaviour**

解除绑定行为。

[来源:ISO/IEC 10746-2:2009, 13.5.4]

3.4777

解绑 **unbundle**

箭头含义分离:通过分支箭头段所表示的、对箭头含义的划分。

注:即从对象类型集合中分离出对象类型。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5034]

3.4778

不确定 **uncertainty**

对某一事件、其后果或发生概率缺乏认知或了解的状态(即使是部分状态)。

参见:风险(risk)。

注:不确定性通常是风险因素的根本原因。在业务决策中,不确定性指无法量化的波动;这类波动的概率不能用于决策分析。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5035]

3.4779

不确定性绩效域 **uncertainty performance domain**

针对与风险和不确定性相关的活动与职能的绩效域。

[来源:项目管理知识体系指南(PMBOK® 指南) — 第七版]

3.4780

无条件分支 **unconditional branch**

不考虑执行条件即可发生的跳转。

参见:条件跳转(conditional jump)。

[来源:GB/T 11457-2006, 2.1784]

3.4781

无条件跳转 **unconditional jump**

不考虑执行条件即可发生的跳转。

参见:条件跳转(conditional jump)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5037]

3.4782

欠拟合 underfitting

指机器学习模型未能学习到训练数据的内在规律与趋势,致使模型难以进行准确预测。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5038]

3.4783

下溢异常 underflow exception

算术运算结果的数值过小,无法在指定存储单元中表示时所产生的异常。

参见: 寻址异常 (addressing exception)、数据异常 (data exception)、操作异常 (operation exception)、上溢异常 (overflow exception)、保护异常 (protection exception)。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5039]

3.4784

基础许可证 underlying license

软件最初购买或获取时所附带的使用许可,且通常可直接关联至具体购买记录 (ISO/IEC 19770-1:2012 信息技术 软件资产管理 第1部分: 符合性流程和分层评估, 3.15)。

注: 基础许可可能包含相关条件,要求其需与其他一个或多个许可证结合使用,以形成有效的完整许可。它也可能包含使用未来版本的容量或权限,或规定其如何通过新版本升级/替换的限制,或如何通过与另一购买记录直接关联的许可证结合实现许可升级。

3.4785

可理解性 understandability

一个系统在系统组织层面和详细陈述层面都容易被理解。

可理解性与可读性相比,在更普遍的层面上与系统的连贯性有关。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5040]

3.4786

未展开论证项 undeveloped argument

待由具体参数替换的占位参数。

[来源: ISO/IEC/IEEE 15026-2:2022, 3.1.7]

3.4787

无向图 undirected graph

在节点的相互连接中无隐含方向的图.与 2.485 条 定向图 (directed graph) 相对。

[来源: GB/T 11457-2006, 2.1787]

3.4788

一维性 unidimensionality

指一组度量指标背后存在单一属性或结构。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5043]

3.4789

统一建模语言 Unified Modeling Language; UML

用于可视化、指定、构建和记录面向对象的软件密集型系统工件的图形语言。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5044, 有修改]

3.4790

统一建模语言 unified modeling language™; UML

用于可视化、规范、构建和记录面向对象软件密集型系统制品的图形化语言。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5044]

3.4791

统一资源标识符 uniform resource identifier; URI

标识互联网上可用的抽象或物理资源的紧凑字符序列。

注: 用于 URI 的语法在 IETF RFC 3986 中定义。

[来源: ISO/IEC 19770-5:2015, 3.51]

3.4792

统一资源定位符 Uniform Resource Locator; URL

通过指定资源地址及所用访问协议来标识互联网资源(如网页)的机制。

注: IETF 指定的术语是统一资源标识符(URI), URL 是其子集。

[来源: ISO/IEC/IEEE 23026:2023, 3.1.23]

3.4793

非预期用户 unintended user

指自主 / 智能系统(A/IS)的设计者并未将其作为使用对象, 但仍与该系统产生交互的人员。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5047]

3.4794

未解释的 uninterpreted

用于说明对于其实现未由 UML 规定的某一或某些类型的一种占位符。每一未解释值都有一个对应的串表示。

[来源: GB/T 11457-2006, 2.1788]

3.4795

独特的功能 unique function

一个在形式和/或逻辑处理上不同于应用程序提供的其他任何功能的功能。

[来源: ISO/IEC/IEEE 24765:2017, 3.4423]

3.4796

唯一性约束 uniqueness constraint

一种约束, 规定在唯一性约束中命名的所有属性上, 同一个类的任意两个不同实例不得具有相同取值。

[来源: ISO/IEC/IEEE FDIS 24765, 3.1.199]

3.4797

单元 unit

在计算机软部件设计中规定的一独立的可测试的元素。

2. 计算机程序的逻辑独立的部分。

3. 不可分为其他部件的软部件。

4. 见: 2.1724 条 测试单元(test unit)。

注: “模块(module)”、“部件(component)”和“单元(unit)”常常可以交互使用或在不同的情况下相互可以定义为子元素, 取决于上下文。这些的关系尚未标准化。

[来源: GB/T 11457-2006, 2.1789]

3.4798

(测量) 单位 unit (of measure)

按约定定义并采用的特定量词, 其他同类量词与这个量词进行比较, 以表示它们相对于这个量词的大小。

注 1：只有以相同测量单位表达的量才可直接比较。单位的例子有故障数和失效数。小时（h）和米（m）也都是测量单位。

注 2：测量单位依惯例都赋予名称和符号。

注 3：本术语基于 GB/T 25000.1-2010 中对测量单位的定义。

[来源：GB/T 25000.21-2019, 4.18]

3.4799

测量单位 unit of measurement

按约定定义和采用的具体量，其他同类量与这个量进行比较，用以表示它们相对于这个量的大小。

注：改写 1993 年版计量学基本与通用术语国际词汇。

[来源：ISO/IEC 15939:2007]

3.4800

规模综合单价 unit price of size

单位规模的直接人力成本与间接成本之和。

注：单位通常为元每功能点。

[来源：GB/T 36964-2018]

3.4801

单元需求文档集 unit requirements documentation

对于测试单元，设置功能、接口、性能和设计约束的文档。

[来源：GB/T 11457-2006, 2.1791]

3.4802

单元测试 unit test

独立的硬件或软件单元或相关单元组的测试。

参见：2.264 条 部件测试（component testing）、2.788 条 集成测试（integration testing）、2.800 条 接口测试（interface testing）和 2.1669 条 系统测试（system testing）。

同义词：unit testing。

[来源：GB/T 11457-2006, 2.1792, 有修改]

3.4803

单元测试框架 unit test framework

便于单元测试的环境。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5054]

3.4804

单元测试 unit testing

独立的硬件或软件单元或相关单元组的测试。

参见：部件测试（component testing）、集成测试（integration testing）、接口测试（interface testing）、系统测试（system testing）。

[来源：GB/T 11457-2006, 2.1792]

3.4805

通用异步收发器 universal asynchronous receiver/transmitter; UART

用于在并行和串行格式之间转换数据的串行通信的电子单元。

注：数据格式和传输速度是可配置的。通常是微控制器的一部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5055]

3.4806

通用串行总线 universal serial bus; USB

计算机与外设之间的串行通信接口，包含两条数据线和两条电源线。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5056]

3.4807

解除紧缩 unpack

从紧缩的数据，恢复一个或多个数据项的原始形式。

参见：紧缩 (pack)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5057]

3.4808

无脚本测试 unscripted testing

测试者的动作不是由测试用例中的书面指令所规定的动态测试。

[来源：GB/T 38634.1-2020, 3.94]

3.4809

取消特化 unspecialize

泛化

将某实例从属于簇内当前子类的实例，变更为不属于该簇内任何子类的实例。

参见：再特化 (respecialize)、特化 (specialize)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.1.200]

3.4810

非分层语言 unstratified language

能用作其他元语言的语言。

示例：英语 (English)、德语 (German)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5061]

3.4811

无监督学习 unsupervised learning

学习一种函数的任务，该函数将无标记输入数据映射到隐式表示。

参见：监督学习 (supervised learning)。

[来源：ISO/IEC TR 29119-11:2020 3.1.85]

3.4812

“直至”语句 UNTIL

测试后迭代 post-tested iteration

单入口、单出口循环，在此循环中，控制循环条件在循环体后执行。

参见：闭合循环 (closed loop)、“当”语句 (WHILE)、尾随判定 (trailing decision)。

3.4813

展开 unwind

在程序设计中，显式地且充分地说明在循环的多次执行中引用的所有指令。

参见：直线编码 (straightline coding)。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5064]

3.4814

运行中 up

用于说明可操作的正在服务中的系统或部件。

参见：忙（busy）、关机的（down）、空闲的（idle）。

注：这样的系统是忙的或空闲的。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5066]

3.4815

开机时间 up time

系统或部件是可操作的和正在服务中的时间周期。即忙时间和空闲时间之和。

参见：关机时间（down time）、忙时间（busy time）、空闲时间（idle time）、平均失效时间间隔（mean time between failures）、设置时间（set-up time）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5067]

3.4816

可更新参数 updatable argument

赋予操作参数的一种标识，该标识用于指明可向其发送请求的实例，且发送的请求将改变该实例的状态。

注：未被指定为“可更新（updatable）”的参数意味着：无法发送任何可改变该参数所标识实例状态的请求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5068]

3.4817

上传 upload

将程序或数据从一台相连的计算机上传至资源更充裕的计算机。

注：通常指从个人计算机传输至服务器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5069]

3.4818

向上兼容的 upward compatible

关于与后一个或它自己的更复杂的版本兼容的硬件或软件。例如，能处理由它自己的后续版本创建的文件。与 2.504 条 向下兼容的（downward compatible）相对。

[来源：GB/T 11457-2006, 2.1799]

3.4819

向上压缩 upward compression

在软件设计中，一种反模块化形式：将从属模块直接以代码内联方式复制到上级模块的主体中。

参见：横向压缩（lateral compression）、向下压缩（downward compression）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5071]

3.4820

易用性 usability

在指定使用情境下，特定用户能够有效、高效且满意地使用某一系统、产品或服务以达成既定目标的程度。

注：可用性可以根据其子特性作为产品质量特性来指定或测量，或者直接通过作为使用质量子集的度量来指定或测量。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5077]

3.4821

可用性缺陷 usability defect

导致用户意图或用户操作与系统属性及行为不匹配的产品特性。

例如：无需作为完成任务一部分的额外不必要的步骤、误导性信息、用户界面中信息不足或质量低劣、意外的系统响应、导航方面的局限性、无效的错误恢复机制、用户界面的物理特性不适合用户的物

理特征、产品特性与既定标准的偏差。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5079]

3.4822

可用性发现 usability finding

已识别的可用性缺陷、可用性问题或与可用性相关的积极特性。

示例:交互系统属性与指定标准(如用户需求、原则、设计指南或既定惯例)的偏离也是可用性发现。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5080]

3.4823

可用性检查 usability inspection

由评估人员依据既定准则,对界面的可用性相关方面进行审慎判断后开展的评估。

注:可用性审查是对多种方法的统称,包括启发式评估、认知走查、标准审查、多方参与走查以及一致性审查。审查人员可包括可用性专家、开发人员、最终用户或其他专业人员。评估准则可包括良好实践或已成文的原则、指南、需求或标准。该评估可在参考文件辅助下开展,也可独立开展。

[来源:ISO TR 25060:2023, 3.4.2]

3.4824

可用性实验室 usability laboratory

通常指一组配备音视频设备、用于记录用户反馈的评估室和观察室。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5082]

3.4825

可用性目标 usability objective

在特定使用情境下以有效性、效率和满意度的形式所表达的可用性水平,该水平可予以验证。

注:可用性目标可以表述为用户需求,在这种情况下,要达到的水平即为可用性要求;或者也可以表述为期望的目标水平,这取决于其在设计和评估中的使用方式。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5083]

3.4826

可用性问题 usability problem

使用过程中导致效果、效率或满意度差的情况。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5084]

3.4827

可用性测试 usability test

为确定已实现的系统是否达到用户确定的功能目标所作的一种测试。

[来源:GB/T 11457-2006, 2.1802])

3.4828

可使用性测试 usability test

适用性测试 fitness-for-use test

验证已实现系统是否满足用户规定的功能目标的测试。

参见:可用性测试(usability testing)。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5085]

3.4829

易用性测试 usability testing

由代表性用户使用系统执行特定任务,以便对效率、有效性和/或用户满意度进行评估的活动。

[来源:ISO/IEC TR 25060:2010, 2.17]

3. 4830

可用性演练 usability walkthrough

可用性评估中，一个或多个评估人员扮演用户的角色逐步通过一个场景，并识别与成功完成场景相关的可用性问题。

注：评估人员可以包括可用性专家、开发人员、最终用户或其他类型的专业人员。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5087]

3. 4831

使用方式 usage mode

文档开发人员期望文档被使用的主要方式。

[来源:ISO/IEC/IEEE 26514:2022, 4. 53]

3. 4832

用例 use case

在 UML 中，系统为参与者提供可衡量的价值结果的一项完整任务。

[来源:ISO/IEC/IEEE FDIS 24765, 有修改]

3. 4833

用例图 use case diagram

显示参与者、用例及其关系的 UML 图。

注：用例描述用户的目标和需求，包括交互的顺序、用户和系统之间。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5091]

3. 4834

用例实例 use case instance

用例中规定的动作序列的执行。用例的一个实例。

[来源: [GB/T 11457-2006, 2. 1806]]

3. 4835

用例模型 use case model

借助用例描述系统功能需求的模型。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5092]

3. 4836

用例规范 use case specification

描述用例的文档。

注：用例规范的基本部分是用例名称、简要描述、前提条件、基本流程、后置条件和替代流程。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5093]

3. 4837

使用错误 use error

用户在使用交互系统时的作为或不作为，导致系统响应偏离预期。

注：使用错误源于用户、界面、任务或环境的不匹配，不含系统故障。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5094, 有修改]

3. 4838

IT 的使用 use of IT

为满足组织需求而规划、设计、开发、部署、运营、管理和应用 IT。

注：包括内部团队、IT 专业部门或外部供应商及公共服务（如 SaaS 提供商）对 IT 服务的供需。

[来源:ISO/IEC/IEEE FDIS 24765, 3. 5095]

3.4839

用处 usefulness

用户对实用目标的感知实现的满意程度，包括使用结果和使用后果。[来源：ISO/IEC 25010:2011 系统和软件工程-系统和软件质量要求和评估 (SQuaRE)-系统和软件质量模型，4.1.3.1]。

3.4840

用户 user

使用可操作的系统完成一项特定的功能的个人或机构。(可以是买主或需方的同义词。)

[来源:GB/T 11457-2006, 2.1808]

3.4841

用户帮助 user assistance

产品在规定使用环境下，供具有最广泛特征与能力的人群使用，以实现指定目标的能力。

参见：可访问性。

[来源：ISO/IEC 25010:2023, 3.4.7]

3.4842

用户可控性 user controllability

用户能够及时适当干预 AI 系统运行的程度。

[来源：ISO/IEC 25059:2023, 3.2.1]

3.4843

用户文档集 user documentation

描述、解释或指导用户如何使用系统的信息，通常随软件提供。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5100, 有修改]

3.4844

用户参与度 user engagement

产品以友好且具有激励性的方式呈现功能和信息，鼓励持续交互的能力。

注：产品的属性能够提升用户的愉悦感与满意度，例如色彩协调、界面直观、语音引导友好。

[来源：ISO/IEC 25010:2023, 3.4.5]

3.4845

用户错误保护 user error protection

产品防止操作错误的能力，或 IT 服务保护用户免于犯错的程度。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5102, 有修改]

3.4846

用户体验 user experience

因使用或预期使用产品、系统或服务而产生的用户的感知和响应。

注：用户体验受品牌形象、呈现方式、功能、系统性能、交互行为及辅助能力影响；也受用户先前经验、态度、技能、个性和使用情境影响。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5103]

3.4847

用户组 user group

目标用户群体中因年龄、文化或专业知识等因素而与其他目标用户区分开来的子集，这些因素可能影响其可用性。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5104]

3. 4848

用户指南 user guide

提供使用系统或组件以获得预期结果所需信息的文档。

[来源: ISO/IEC 2382:2015 信息技术 — 词汇]

3. 4849

用户交互 user interaction

用户与交互系统之间通过用户界面交换信息以完成预期任务。

注: 用户交互规范聚焦于用户交互本身, 不涉及实现细节。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5105]

3. 4850

用户界面 user interface

用户接口

允许信息在个人用户与计算机系统的硬件或软件部件间传送的接口。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5106]

3. 4851

用户界面美学 user interface aesthetics

用户界面为用户带来愉悦和令人满意的交互的程度。[来源: ISO/IEC 25010:2011 系统和软件工程—系统和软件质量要求和评估 (SQuaRE)—系统和软件质量模型, 4. 2. 4. 5]。

注: 指增加用户愉悦和满意度的产品或系统的特性, 例如颜色的使用和图形设计的性质。

3. 4852

用户界面元素 user interface element

软件呈现给用户的用户界面实体。

注 1: 用户界面元素可以是交互式的, 也可以不是交互式的, 与任务相关的实体或用户界面的实体。

同义词: 用户界面对象。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5107]

3. 4853

用户界面元素 用户界面对象 user interface element user interface object

软件呈现给用户的用户界面实体。

[来源: ISO/IEC TR 25060:2010]]

3. 4854

用户界面元素 user interface element

用户界面对象 user interface object

软件向用户呈现的用户界面实体。

示例: 任务对象 (如信件、销售订单、电子部件或接线图) 或系统对象 (如打印机、硬盘或网络连接) 的可视化表示或交互机制; 基本对象 (如窗口标题栏、菜单项、按钮、图像映射和可编辑文本字段) 或容器 (如窗口、分组框、菜单栏、菜单、互斥选项按钮组以及由若干较小图像组成的复合图像); 消息和操作提示。

注: 用户界面元素可以是交互式的或非交互式的, 既可以是与任务相关的实体, 也可以是用户界面的实体。

[来源: ISO TR 25060:2023, 3. 2. 7]

3. 4855

用户界面任务 user interface task

一种隐藏界面细节、与人类用户顺序交互的任务。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5108]

3.4856

用户手册 user manual

为使用系统或部件以获得预期结果而提供必要信息的文档，也可包含用户、所有者及供应方的权利和责任说明。

参见：数据输入表、诊断手册、安装手册、操作员手册、程序员手册、支持手册。

注 1：通常描述系统或部件的功能、限制、选项、允许的输入、预期输出、可能的错误消息及特殊指令。当区分计算机系统操作人员（如装磁带等）与系统最终使用者时，用户手册有别于操作员手册。

同义词：用户指南。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5109]

3.4857

用户需要 user need

用户需要系统满足的用户功能需求和用户非功能需求的集合。

注：由一个系统或一个软件段提供的所有功能服务来自于用户功能需求。

[来源:GB/T 42452-2023, 1.683]

3.4858

用户通知 User notification

3.4859

用户资料 user profile

特定用户或用户组独有的一组属性，例如工作职能或服务订阅，用于控制用户可以访问的系统或网页部分。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5111]

3.4860

用户需求 user requirements; UR

为满足已识别的用户需要，为交互系统的设计和评估提供依据的一组使用需求。

注 1：用户需求规定了使用系统时应满足的用户需要和能力程度，来源于用户需要和能力，包含功能性用户需求和非功能性用户需求。

同义词：使用要求。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5112, 有修改]

3.4861

用户故事点数 user story point

敏捷开发方法中用于度量用户故事范围大小的相对度量。

[来源:ISO/IEC TR 7052:2023, 3.51]

3.4862

用户系统交互要求 user system interaction requirements

规定用户为实现目标所需交互（如信息识别、输入操作、选择操作、接收输出）的用户要求。

[来源:ISO TR 25060:2023, 3.3.5]

3.4863

用户终端 user terminal

使用户能够与计算机通信的终端。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5116]

3.4864

用户类型 user type

模拟用户的一种分类，由以下两项组合。

- 1) 使用链类型的相对频率；
- 2) 准备时间（平均值及其标准差）。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5117]

3.4865

用户视图 user view

用户描述的功能用户需求。

注：开发人员将用户视角转换为软件，以便提供解决方案。

[来源:GB/T 42566-2023, 1.760]

[来源：同义词：user perspective]

[来源：用户所感知的功能性用户需求。]

[来源：1：开发人员依据用户视角转化为软件提供解决方案。]

[来源:ISO/IEC/IEEE FDIS 24765, 3.5118]

3.4866

基于用户的评估 user-based evaluation

由代表性用户执行系统任务的评估，旨在识别可用性问题或测量效率、有效性、用户满意度及其他用户体验。

[来源：ISO/IEC 25066:2016, 3.22]

3.4867

用户定义的可靠性类 user-defined reliability class

对特定类型系统行为的书面化集合，其失效具有足够重要性，值得在安全、保密性、功能性、性能和利用率这些常规可靠性类别之外，单独进行规定、观测和评估。

示例：人工智能幻觉。

[来源：IEEE 982:2024, 3.1]

3.4868

使用相关质量要求 use-related quality requirements

规定交互式系统使用预期结果和相关质量标准的用户要求。

[来源：ISO TR 25060:2023, 3.3.6]

3.4869

用户友好的 user-friendly

用于修饰或说明人类使用的易用性和方便性。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5121]

3.4870

用户可识别的 user-recognizable

由用户和软件开发人员商定和理解的流程或数据要求。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5122]

3.4871

用户-系统交互 user-system interaction

用户与交互式系统之间通过用户界面进行信息交换。

[来源：ISO TR 25060:2023, 3.2.1]

3.4872

功利主义 utilitarianism

采用伦理决策方法考量系统设计与部署的后果（危害与效益）。

注：功利主义旨在最大化行为的积极后果并最小化消极后果，从而在长期内实现直接与间接利益相关者在生活中的最大满足感与幸福感。

[来源：IEEE 7000:2021, 3.1]

3.4873

公用事业 utility

设计用于执行某些常用支持功能的软件工具。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5126]]

3.4874

公共设施软件 utility software

计算机程序或例程序，其设计目的是为其他应用软件、操作系统或系统用户提供他们所要求的某些通用支持功能。

[来源：[GB/T 11457-2006, 2.1816]]

3.4875

利用率 utilization

在计算机性能评估中，表示系统或组件忙碌的时间除以可用时间的比率。忙碌时间，空闲时间，正常工作时间。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5127]]

3.4876

利用界限定理 utilization bound theorem

实时调度定理，说明由速率单调算法调度的一组 n 个独立周期性任务将始终满足其截止日期的条件。

[来源：ISO/IEC/IEEE 24765:2017, X.X]]

3.4877

基准结果应用活动 utilize benchmarking results activity; URa

利用基准比对结果活动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5073]

3.4878

有效的 valid

遵循指定 XML 模式文档且从 XML 角度来看有效的信息结构的状态。

[来源：ISO/IEC 19770-5:2015]]

3.4879

范围确认 validate scope

对已完成的项目可交付成果进行正式验收的过程。

[来源：ISO/IEC/IEEE 24765:2017, 3.4497]

3.4880

经过确认的交付件 validated deliverables

作为执行质量控制过程以确定正确性的结果的可交付成果 [项目管理知识体系指南第五版]。

[来源：ISO/IEC/IEEE 24765:2017, 3.4498]]

3.4881

已验证测量 validated metric

测量值已通过统计方式与相应质量因子数值建立关联。

[来源：IEEE 1061-1998 (R2004)，2.25]

3.4882

确认 validation

通过提供客观证据来证实针对某一特定预期用途的需求已经得到满足。

注：在生存周期周境中，确认是确保并确信系统能够实现其预期用途、目标和目的的一系列活动。

[来源：GB/T 25000.1-2021，3.52]

通过自动化技术（如可能），提供客观证据，适时的确认特定预期用途或应用需求已得到满足。

参见：验证。

注：在系统生存周期周境中，确认是在预期运行环境中确信系统能够实现其预期用途、目标和目的（满足相关方要求）的一系列活动。构建了正确的系统，或系统运行满足业务目标，确认证实系统可供用户完成其特定任务。

“已确认”用于表示相应状态。

[来源：IEEE 2675-2021，3.1]

3.4883

确认动作 validation action

描述要验证的内容（作为参考的元素）、执行操作的项目、执行操作的预期结果、要应用的验证技术以及在哪个级别分解的操作感兴趣的系统。

[来源：ISO/IEC TS 24748-6:2016]

3.4884

确认数据 validation data

用于在调整候选机器学习模型时对其进行评估的数据集。

[来源：ISO/IEC TR 29119-11:2020]

3.4885

验证程序 Validation procedure

一种验证过程，包括一组验证操作，使用特定的验证技术，使用特定的验证使能器执行。

3.4886

验证记录 Validation report

为相关方准备的帐户，用于交流验证活动的状态、结果和结果。包括验证结果和确认系统是否满足涉众需求和业务需求的客观证据。是否还应沟通评估结果或结果的置信度。

3.4887

确认测试 validation test

用于确定已实现的系统是否满足其规定需求的测试。

[来源：ISO/IEC 2382:2015，X.X]

3.4888

风险价值 value at risk

被视为受到破坏或威胁的价值。

[来源：IEEE 7000:2021，3.2]

3.4889

价值基准 value baseline

在优化前对一组资产进行的衡量，为每组被追踪的资产分配相关价值。

[来源：ISO/IEC 19770-5:2015，3.53]

3.4890

价值承载者 value bearer

承载价值的系统、人、事物、行为或关系。

[来源：IEEE 7000:2021, 3.1]

3.4891

价值收益 value benefit

培养价值的积极状态或活动。

[来源：IEEE 7000:2021, 3.2]

3.4892

价值链分析 value chain analysis

以产品或服务形式提供或接收价值的整个活动序列或参与方。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.42]

3.4893

价值变化覆盖率 value change coverage

在一系列测试中，激活值差异超过某个阈值的神经元数量占神经网络中神经元总数的比例（通常以百分比表示）。

[来源：ISO/IEC TR 29119-11:2020]

3.4894

价值等级 value class

表示由纯值构成的实例的类。

注：值类所包含的实例不会产生或消失，且不能改变状态。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5142]

3.4895

价值集群 value cluster

一组包含一个核心价值 and 几个有助于核心价值或与之相关的价值。

[来源：IEEE 7000:2021, 3.2]

3.4896

价值交付办公室 value delivery office; VDO

项目交付支撑体系，专注于指导团队、在组织内构建敏捷技能与能力，并辅导发起方和产品负责人更有效地履行其职责。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4897

价值交付系统 value delivery system

旨在建立、维持和/或发展组织的一系列战略性业务活动。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4898

价值展示者 value demonstrator

核心价值的潜在体现，它要么有助于核心价值，要么破坏核心价值。

[来源：IEEE 7000:2021, 3.2]

3.4899

价值处置 value disposition

系统特性，它可以促进或阻碍一个或多个价值观的实现。

[来源：IEEE 7000:2021, 3.1]

3.4900

价值工程 value engineering; VE

对功能特性、系统架构、基础设施及材料选型进行分析，以在满足规定性能、质量、可靠性和安全要求的前提下，以最低生存周期成本实现核心功能。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5148]

3.4901

价值伤害 value harm

破坏价值的负面状态或活动。

[来源：[IEEE 7000:2021, 3.2]]

3.4902

价值领导者 value lead

被指派负责通过需求和设计工件协调和执行与道德价值观的启发和优先排序以及价值观的可追溯性相关的任务的人员。

[来源：[IEEE 7000:2021, 3.2]]

3.4903

值列表约束 value list constraint

调整语序及补充来源。

用于指定某个值类的所有可接受实例值集合的约束。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5151]

3.4904

价值主张 value proposition

组织向其客户传递的产品或服务价值。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4905

取值范围约束 value range constraint

用于指定值类中所有可接受实例值的集合，其中实例值受下限和/或上限边界约束的约束条件。

示例：方位角，要求在 -180° 至 $+180^{\circ}$ 之间。

注：范围约束仅在指定线性排序时才有效。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5153]

3.4906

值寄存器 value register

出于透明度和可追溯性原因而创建的信息存储，其中包含在道德价值获取和优先排序以及对道德价值要求的可追溯性中获得的数据和决策。

[来源：[IEEE 7000:2021, 3.1]]

3.4907

价值流图 value stream map

显示过程中的关键步骤及各步骤耗时，以识别浪费。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4908

价值流图法 value stream mapping

精益企业方法，用于记录、分析和改进为客户生产产品或提供服务所需的信息或物料流。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4909

增值经销商 value-added reseller; VAR

获得许可对现有产品进行重新包装并提供支持的公司，例如组合软件包。

[来源：ISO/IEC/IEEE 24765g:2018]

[来源：同义词：增值经销商（value added reseller）。]

3.4910

基于价值的系统需求 value-based system requirement

一种系统需求，可追溯至伦理价值需求、价值群组和核心价值观。

[来源：[IEEE 7000:2021, 3.1]]

3.4911

无效测度 vanity metric

表面上能够显示某些结果，但无法为决策提供有用信息的测度。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.4912

可变性 variability

可区分产品线成员的特性。

注 1：成员产品间的差异包含多个方面，例如功能性、质量属性、成员产品的使用环境、用户、约束条件以及实现功能性和质量属性的内部机制。

注 2：区分系统和软件可变性与产品线可变性的概念很重要。由于软件系统具有内在可塑性、可扩展性或特定使用周境的可配置性，因此部分或全部由软件构成的系统都被认为具有软件可变性。产品线可变性关注的是产品管理明确定义的可变性。本文件主要涉及产品线可变性。

[来源：GB/T 41865-2022, 1.57]

3.4913

可变性结合 variability binding

确定变异模型中定义的变异点变异的行为。

[来源：ISO/IEC 26555:2015, 3.8]]

3.4914

可变性约束 variability constraint

变体与可变点之间，两个变体之间，以及两个可变点之间的约束关系。

[来源：GB/T 41865-2022 软件与系统工程 产品线工程与管理参考模型，1.57]

3.4915

可变性依赖 variability dependency

变点与变体集之间的关系，表示变点暗示着变体的决定。

[来源：ISO/IEC 26550:2015 , 3.26]]

3.4916

可变性文件 variability documentation

产品线内跨成员产品使用的可变性模型的详细描述。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5165]

3.4917

可变性实施 variability implementation

源代码或可执行模块的可变性开发。

[来源：ISO/IEC 26553:2018, 3.19]]

3.4918

需求的可变性 variability in requirements

需求工程中的外部和内部可变性。

[来源: ISO/IEC 26551:2016, 3.20]]

3.4919

空间的可变性 variability in space

不同形态同时发生的变异。

[来源: ISO/IEC 26555:2015, 3.9]]

3.4920

测试用例的可变性 variability in test cases

域测试用例中包含的可变性, 在应用程序测试期间将绑定这些可变性, 以导出具体的测试用例。

[来源: ISO/IEC 26554:2018, 3.12]]

3.4921

时间的可变性 variability in time

不同时期发生的变异。

[来源: ISO/IEC 26555:2015, 3.11]]

3.4922

可变性指数 variability index

用于捕捉特征、子系统、部件、产品或产品线之间变异程度的指标。

注: 变异指数是通过公式计算得出的, 如单个指标或多个指标的组合(函数)。计算变异指数的公式反映了特征、子系统、部件、产品或产品线的相对大小、工作量、复杂性或重要性作为权重。

[来源: ISO/IEC 26564:2022, 3.6]

3.4923

可变性管理 variability management

管理任务与可变性相关, 有两个维度: 可变性维度和资产维度。

[来源: ISO/IEC 26550:2015, 3.27]]

3.4924

可变性机制 variability mechanism

支持领域资产组装的产品线可变性实施方法。

[来源: ISO/IEC 26555:2015, 3.12]]

3.4925

可变性机制可操作化 variability mechanism operationalization; VMO

在各特定领域或应用工程生存周期的每个阶段, 对可变性机制进行充分的规定或约束。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5237]

3.4926

可变机制操作化(VMO) variability mechanism operationalization (VMO)

在每个特定领域或应用工程生存周期阶段充分提供或绑定可变性机制。

[来源: ISO/IEC 26557:2016, 3.15]]

3.4927

可变性模型 variability model

产品线可变性的明确定义。

[来源: ISO/IEC 26550:2015, 3.28]]

3.4928

可变性建模计划 variability modeling plan

包含时间表、定义的角色和职责以及定义的质量保证措施的文档，这些措施将应用于可变性建模。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5176]

3.4929

可变性建模策略 variability modeling strategy

可变性建模方法论、可变性模型验证的严格程度、规则、约束条件及其他细节，以支持可变性模型在整个可变性管理中的作用。

[来源: [[来源: ISO/IEC 26558:2017, 3.10]]]

3.4930

可变性建模计划 variability modelling plan

包含时间表、定义的角色与职责以及应用于可变性建模的的质量保证措施的文档。

[来源: ISO/IEC 26558:2017, 3.9]

3.4931

可变性建模策略 variability modelling strategy

可变性建模方法、可变性模型验证的严格程度、规则，约束条件，以及支持可变性模型在整个可变性管理中作用的其他细节。

[来源: ISO/IEC 26558:2017, 3.10]

3.4932

可变性可追溯性 variability traceability

为包含域资产和应用资产的变异模型建立跟踪链接，其中变体相互关联。

[来源: [[来源: ISO/IEC 26555:2015, 3.12]]]

3.4933

可变的 variable

值可以改变的数量或数据项。

[来源: [[IEEE 1320.2-1998 (R2004), 3.1.205]]]

3.4934

可变技术状态项 variable configuration item

技术状态项中在不同产品间可变的实体。

[来源: ISO/IEC 26563:2022, 3.9]

3.4935

可变成本 variable cost

成本，例如材料成本，直接取决于生产率。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5180]]]

3.4936

变强度 variable strength

对不同的参数子集给出不同组合强度的要求。

示例：一个系统具有 a, b, c, d, e 五个参数，默认组合强度为 2，其中 a、b、c 三个参数由于特殊要求需要设置 3 强度组合，则该系统的测试用例需同时满足 a, b, c, d, e 五个参数的 2 强度组合，以及 a、b、c 三个参数的 3 强度组合。

注：组合测试中组合强度的一种特殊要求。通常是在默认组合强度的基础上，待测软件的某几个参数子集采用不同于默认值的组合强度。

[来源: GB/T 38639-2020, 2.5]

3.4937

变量踪迹 variable trace

数据流踪迹 data-flow trace

数据踪迹 data trace

值踪迹 value trace

记录在计算机程序执行过程中访问或更改的变量的名称和值。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5182]

3.4938

偏差分析 variance analysis

差异分析

一种确定基准与实际绩效之间差异原因和程度的技术。[来源: ISO/IEC/IEEE FDIS 24765, 3.5184]。

3.4939

完工偏差 variance at completion; VAC

预算赤字或盈余金额的预测, 表示为完工预算与完工估算之间的差额。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4940

完工差异 variance at completion (VAC)

预算赤字或盈余金额的预测, 表示为完成时的预算与完成时的估计之间的差额。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5185]]

3.4941

变体 variant

在容错中, 应用软件多样性产生的程序版本。

[来源: ISO/IEC 26550:2015, 3.27]]

3.4942

变体选择 variant selection

在变异点中选择变体的决策。

[来源: ISO/IEC 26557:2016, 3.18]]

3.4943

可变性 variation

实际情况与基准计划中包含的预期情况不同[《项目管理知识体系指南》(PMBOK 指南) 第五版]。

[来源: ISO/IEC/IEEE 24765:2017, 3.4530]]

3.4944

空间变化 variation in space

不同形状的可变人工制品的不同版本的存在, 并由不同的产品使用。

参见: 空间的可变性。

[来源: ISO/IEC 26563:2022, 3.10]

3.4945

时间变化 variation in time

在不同时间存在不同版本的公共或可变人工制品。

参见: 时间变化 (variability in time)。

注: 时间维度的变化与软件演化是同步的。

[来源: ISO/IEC 26563:2022, 3.8]

3.4946

变化点 variation point

产品线上下文中产品、领域资产和应用程序资产的特定变量特征对应的表示。

[来源: ISO/IEC 26550:2015, 3.29]]

3.4947

可变性建模计划 variability modeling plan

包含时间表、定义的角色和职责以及定义的质量保证措施的文档, 这些措施将应用于可变性建模。

[来源: ISO/IEC 26558:2017, 3.9]]

3.4948

验证交叉引用矩阵 VCRM

验证交叉引用矩阵。

[来源: [IEEE 15288.2:2014 IEEE 国防计划技术审查和审计标准, 3.2]]

3.4949

版本说明文件 VDD

版本说明文件。

3.4950

速度 velocity

衡量在预定时间间隔内生产、验证和接受可交付成果的团队生产率。速度是一种容量规划方法, 经常用于预测未来的项目工作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5194]]

3.4951

速度图 velocity chart

用于追踪在预定义时间间隔内可交付成果的生产、验证和验收的速率的图表。

[来源: 项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.4952

供应商分支 vendor branch

跟踪导入软件版本的分支。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5196]]

3.4953

供应商服务上下文代码集标识符 vendor service context codeset ID; VSCID

供应商服务上下文代码集标识符。

[来源: ISO/IEC/IEEE 24765:2017, 3.4574]

3.4954

动词词组 verb phrase

关系标签的一部分, 以通过组合第一类名称、动词短语、基数表达式和第二类名称或角色名称形成句子的方式命名关系。

[来源: [IEEE 1320.2-1998 (R2004), 3.1.206]]]

3.4955

可验证的 verifiable

可以由人或工具检查正确性。

[来源: [[来源: ISO/IEC/IEEE 15289:2015, 5.27]]]

3.4956

验证 verification

通过提供客观证据来证实规定的需求已予满足。

注：在生存周期周境中，验证是将生存周期的产物与该产物所需求的特性进行对比的一系列活动。这包括但不限于规定的需求、设计描述和系统本身。

[来源：GB/T 25000.1-2021, 3.54]

3.4957

验证动作 verification action

描述要验证的内容（作为参考的元素）、对哪个项目执行操作、执行操作的预期结果、要应用的验证技术以及在哪个级别分解的操作感兴趣的系统。

[来源：ISO/IEC TS 24748-6:2016, 3.1.17]]

3.4958

验证和确认 verification and validation; V&V

确定系统或组件的需求是否完整和正确的过程，每个开发阶段的产品是否满足前一阶段的要求或条件，最终系统或组件是否符合规定的要求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5200]

3.4959

验证与确认工作 verification and validation effort; V&V effort

与执行验证与确认过程、活动和任务相关的工作。

[来源：IEEE 1012-2024, 3.1.37]

3.4960

验证和确认计划 verification and validation plan; VVP

验证和确认计划。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5247]

3.4961

验证方法 verification method

为提供客观证据证明系统的规定需求已得到满足所采用的方法。

[来源：ISO/IEC 23643:2020, 3.34]

3.4962

验证报告 Verification report

为相关方准备的帐户，用于传达验证活动的状态、结果和结果。包括验证结果和确认系统是否满足其要求、体系结构特征和设计属性的客观证据。还应传达对发现或结果的置信度的评估。

3.4963

验证发起人 verification sponsor

要求执行验证并提供资金或其他资源以实施验证的个人或组织。

[来源：ISO/IEC TR 14143-3:2003, 3.11]

3.4964

验证系统 verification system

一种软件工具，它接受计算机程序及其规范表示作为输入，并（可能需要人工辅助）生成程序正确性的证明或反证。

[来源：[GB/T 11457-2006, 2.1828]]

3.4965

验证测试 verification test

在系统开发的特定阶段，为证明系统满足全部规定需求而开展的测试。

3.4966 [[来源：ISO/IEC 2382:2015, X.X]]

验证工具 verification tool

验证期间可用于收集有关验证目标的信息、解释信息或使部分验证自动化的仪器。

[来源：ISO/IEC 23643:2020, 3.35]]

3.4967

验证 verification; V&V effort

确认工作 validation effort

与执行验证与确认（V&V）过程、活动及任务相关的工作。

[来源：IEEE 1012-2024, 3.1.37]

3.4968

验证可交付成果 verified deliverable

已通过控制质量过程检查并确认正确性的已完成项目可交付成果。

[来源：[IEEE_CS_SE_Vocabulary-2021]]

3.4969

验证范围 verify scope

对已完成的项目可交付成果进行形式化验收的过程。

[来源：[IEEE_CS_SE_Vocabulary-2025]]

3.4970

Verilog 硬件描述语言 Verilog hardware description language; VHDL

用于设计和验证数字电路的硬件描述语言。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5206]

3.4971

版本 version

与计算机软件配置项的完全编纂或重编纂相关的计算机软件配置项的初始发行或再发行。

作为与对以前的发行发出变更页导致的修订不同的文件的初始发行或完全再发行。

参见：配置控制、版本描述文件

[来源：GB/T 11457-2006, 2.1829]

3.4972

版本控制 version control

基线的建立与维护，以及对基线变更的识别与控制，以实现返回至先前基线。

[来源：ISO/IEC/IEEE 26511:2018, 3.1.40]]

3.4973

版本描述文件 version description document; VDD

随给定版本的系统或部件一起提供的标识文档。

注：典型内容包括系统或部件的详细目录、记录版本变更及提供安装与运行信息。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5209]

3.4974

版本标识符 version identifier

用于区分配置项版本与其他版本的补充信息。

注：版本号用于标识软件产品的版本，以便与其他版本进行比较。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 5210]

3. 4975

版本化 versioning

为软件配置项的各个阶段分配唯一版本名称或唯一版本号，通常用于特定目的，例如向外部组发布软件产品或识别特定基线。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 5211]

3. 4976

顶点 vertex

在状态机中，某一转移的一种源或目标。顶点可是一个状态或伪状态。

参见：状态、伪状态

[来源：GB/T 11457-2006, 2. 1831]

3. 4977

纵向微指令 vertical microinstruction

规定实现机器语言指令所需的操作序列之一的微指令。

参见：对角微指令、横向微指令。

注：纵向微指令相对较短，为 12~24 位，被称为“垂直的”是因为通常在页面上垂直列出的一系列此类指令，才能完成单条机器语言指令的执行。

[来源：ISO/IEC FDIS 24765, 3. 5212]

3. 4978

小微企业 very small and micro enterprise; VSME

具有少量雇员的组织，雇员人数很多时候小于 50 或为 1。

[来源：INCOME 系统工程手册 v4, 8. 6]

3. 4979

极小实体 very small entity; VSE

企业、组织、部门或项目，人数最多 25 人。

[来源：ISO/IEC 29110-1-2:2024, 3. 124]

3. 4980

视频显示终端 video display terminal; VDT

通常配备键盘等输入单元的显示屏用户终端。

同义词：视频显示单元（VDU）。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 5215]

3. 4981

视频随机存取存储器 video random access memory; VRAM

专为显卡帧缓冲区设计的随机存取存储器。

[来源：ISO/IEC/IEEE FDIS 24765, 3. 5216]

3. 4982

视图 view

从单个视点的角度对系统的表示。

[来源：GB/T 45803-2025, 3. 1. 27]

3. 4983

视图组件 view component

由适用模型种类或图例主导的一个或多个架构视图的可分离部分。

示例：描述访问控制机制的架构视图组件能用在架构描述（3.3）的多个视图中，以解释实体的功能流、行为和安全特征。

注：在架构描述的周境中，图例是约定的非正式文档化形式。

[来源：GB/T 45717-2025, X. X]

3.4984

视图图表 view diagram

视图底层语义的图形表示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5219]

3.4985

视图元素 view element

对模型元素的某一汇集的一种文本的与图形的投影。

[来源：GB/T 11457-2006, 2.1834]

3.4986

视图投影 view projection

模型元素在视图元素上的一种投影。视图投影为视图元素提供位置和式样。

[来源：GB/T 11457-2006, 2.1835]

3.4987

视点 viewpoint

从不同建模角度和利益相关方关注角度出发，为创建、解释和使用模型视图而建立的一组约定，以框定一个或多个关注点。

注：一个视点包含多个视图模型类型。

[来源：GB/T 45615-2025, X. X]

3.4988

视点（在系统上） viewpoint (on a system)

使用一组选定的体系结构概念和结构化规则实现的抽象形式，以关注系统中的特定问题。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5221]

3.4989

视点对应关系 viewpoint correspondence

声明从一个视点的规范中的某些术语或其他语言结构与第二个视点的规范中的术语或结构相关联（例如描述相同的实体）。

注：可以表达的关联形式将取决于所使用的规范技术。对应关系所关联的术语不必使用单一规范技术来表达。这种对应关系可以将一种规范技术中的术语与某种不同的规范技术中的术语关联起来。除了链接每一对单独的术语外，也可以在规范技术本身之间表达通用的对应关系。例如，可以关联在不同规范技术中定义的组合算子，这意味着无论在何处使用这些算子来链接各自视点中的术语，都存在对应关系。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5222]

3.4990

视点陈述 viewpoint statement

模型的 a-0 上下文图中呈现的 IDEF0 模型透视图的简要陈述。

[来源：ISO/IEC/IEEE 24765:2017, 3.4560]

3.4991

违规 violation

偏离系统期望属性或相关声明的行为、动作或事件。

注：在安全领域，“违规”一词用于指人为故意违反程序或规则的行为。一条规则或策略可以规定在违反该规则或其他规则或策略时应采取的行为。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5223]

3.4992

虚拟的 virtual

用于描述看似真实的功能单元，但其功能是通过其他方式实现的。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5224]

3.4993

虚拟地址 virtual address

在虚拟存储系统中，分配给辅助存储单元的地址，使得该单元可以被访问，就如同它是主存储器的一部分一样。

参见：实地址

[来源：ISO/IEC/IEEE FDIS 24765, 3.5225]

3.4994

虚拟机 virtual machine; VM

对计算机及其有关设备进行功能模拟的设备。

[来源：GB/T 11457-2006, 2.1837]

3.4995

虚拟样机 virtual prototype

与物理产品相对应的计算机数字模型集合。

注：它从构造、功能和性能等方面定义、模拟、试验验证和评估物理产品，并且作为物理产品在研制、生产和使用等生存周期内各项活动主要依据。

[来源：GB/T 36457-2018, X.X]

3.4996

虚拟样机系统 virtual prototype system

以虚拟样机为主体，一步集成现实系统应用环境、目标对象等多元科学模型结构的数字化仿真系统。

[来源：GB/T 36457-2018, X.X]

3.4997

虚拟引用 virtual reference

对元模型中除特定元实体以外的概念所做的引用。

注：由带有对角斜线的方框表示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5227]

3.4998

虚拟空间 virtual space

计算机制图中的坐标空间，此空间中的每一单元由用户定义的坐标系表示。

用户程序运行的一个逻辑地址空间。

[来源：GB/T 11457-2006, 2.1839]

3.4999

虚拟存储器 virtual storage; virtual memory

多级存储器 multilevel storage

一种存储分配技术，使辅助存储器能够像主存的一部分一样被寻址。

参见：实存储器、虚拟地址、调页。

注：用户程序和数据的部分内容被置于辅助存储器中，操作系统根据需要自动将其与主存进行交换调入调出。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5228]

3.5000

虚拟系统之系统 virtual system of systems; virtual SoS

系统之系统(SoS)缺乏中央管理权限和中央商定的 SoS 目的。

注：虚拟 SoS 通常是自组织的。大规模行为出现并且可能是可取的，但这种类型的 SoS 依赖于相对不可见的机制来维护它。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5229]

3.5001

虚拟团队 virtual team

在不同地点工作但具有共同目标的人群，他们主要通过电话和其他电子通信方式彼此互动。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5230]

3.5002

虚拟测试环境 virtual test environment

一个或多个组成部分采用数字化仿真的测试环境。

[来源：ISO/IEC TR 29119-11:2020, 3.1.88]

3.5003

虚拟化 virtualization

创建物理信息与通信技术设备或资源的虚拟版本，以提供更高效的信息与通信技术硬件使用。

[来源：ISO/IEC TR 30133:2023(en), 3.1.12]

3.5004

可见性 visibility

针对某一属性，明确“谁可以查看该属性”的规范。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5233]

3.5005

愿景声明 vision statement

对产品预期的概括性、高层次描述，如目标市场、用户、主要优势，以及该产品与市场上其他产品的区别。

[来源：项目管理知识体系指南（PMBOK® 指南）—第七版]

3.5006

可视化数据与信息 visual data and information

以可视化格式组织和呈现数据和信息的工具，如图表、图形、矩阵和图示。

[来源：项目管理知识体系指南（PMBOK® 指南）—第七版]

3.5007

视频显示单元 visual display unit; VDU

通常配备键盘等输入单元的显示屏用户终端。

同义词：视频显示终端（VDT）。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5215]

3.5008

关键业务服务 vital business service

对业务成功至关重要的服务。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5236]

3.5009

V 模型 V-model; Vee model (建议删除)

系统开发生存周期的图形化表示。

注：它用于构建严谨的开发周期模型和项目管理模型。

[来源：ISO/TS 15926-11:2023, 3.1.39]

3.5010

词汇 vocabulary

与特定领域术语子集相关的信息集。

注：词汇通常保持一致性、避免重复、支持同义词。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5239]

3.5011

易失性存储器 volatile memory

仅在通电时才存储数据的单元。

参见：非易失性存储器。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5241]

3.5012

波动性 volatility

快速且不可预测变化的可能性。

[来源：项目管理知识体系指南 (PMBOK® 指南) — 第七版]

3.5013

容积测试 volume testing

性能效率测试的一种类型。用于评价测试项在吞吐量、存储容量或两者兼考虑的情况下处理指定数据量（通常达到最大指定容量或接近最大值）的能力。

[来源：GB/T 38634.1-2020, 3.95]

3.5014

漏洞 vulnerability

软件设计或实现中的潜在缺陷或弱点可能会被执行（意外触发或有意利用）并导致对系统造成损害。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5246]

3.5015

放弃书 waiver

接受配置项目或其他指定项目的书面授权，这些项目在生产过程中或在提交检查后被发现偏离规定要求，但经批准方法原样使用或返工后仍被认为适宜使用的配置项或其他指定项。

参见：配置控制、偏离、工程变更。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5249]

3.5016

走查 walk-through

一种静态分析技术或评审过程，在此过程中，设计者或程序员引导开发组的成员通读已书写的设计或编码，其地成员负责提出问题并对有关技术、风格、可能的错误、是否违背开发标准等方面进行评论。

[来源：GB/T 11457-2006, 2.1843]

3.5017

作战室 war room

用于项目会议和计划的房间，通常显示成本图表、进度状态和其他关键项目数据。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5252]

3.5018

警告 warning

提供给用户的信息中的警示性信息，说明执行某一操作可能导致潜在危险情况，若不加以避免，会造成死亡或严重伤害。

参见：注意、提示。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5253]

3.5019

警告信息 warning message

与安全相关的信息，用于对危险发出警告，并指导如何规避这些危险。

注：警告信息通常在与危险任务相关的分步说明中给出。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5254]

3.5020

浪费 waste

在客户眼中没有为产品或服务增加价值的工作。

[来源：INCONE 系统工程手册，第五版]

3.5021

看门狗定时器 watchdog timer; WDT

如果主程序由于某些故障条件未能定期向其发出信号，则触发嵌入式系统重置或其他纠正措施的电子单元。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5256]

3.5022

瀑布模型 waterfall model

一种软件开发过程的模型，其组成活动典型地按概念阶段、需求阶段、设计阶段、实现阶段、测试阶段和安装及验收阶段按这样的顺序执行：可能有重叠但只有小的或没有重复。

参见：增量开发、快速原型、螺旋模型

[来源：GB/T 11457-2006 2.1844]

3.5023

波形音频文件格式 Waveform audio file format; WAV

波形音频文件格式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5258]

3.5024

磨损失效周期 wearout-failure period

系统或部件生存周期中的周期，在此期间由于恶化，硬件的失效以递增的速度发生。

参见：定失效率周期、早期失效周期、浴盆曲线

[来源：GB/T 11457-2006 2.1845]

3.5025

Web 内容可访问性指南 Web Content Accessibility Guidelines; WCAG

一套规定如何使内容具备可访问性的指南，主要面向残障人士，同时也适用于所有用户代理。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5263]

3.5026

网站负责人 web lead

对网站所有者负责,持续维护网站展示和可用性的人员。

[来源:ISO/IEC/IEEE 23026:2023, 3.1.25]

3.5027

加权里程碑法 weighted milestone method

挣值法,将工作包分成可衡量的部分,每个部分以一个可观察的里程碑结束,然后为每个里程碑的实现分配一个加权值。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5269]

3.5028

福祉 well-being

在繁荣的生态环境中,确保个人、社区和群体的身心健康及社会福利能够持续且可持续地得到发展,同时满足其经济需求。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5270]

3.5029

福祉影响评估 well-being impact assessment; WIA

福祉影响评估。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5275]

3.5030

福祉指数 well-being metrics

衡量内部现象和外部因素的主观和客观指标,涵盖能力和主观幸福感内容,包括但不限于 a) 情感、b) 社区、c) 文化、d) 教育, e) 经济, f) 环境, g) 人居环境, h) 健康, i) 政务, j) 心理健康/精神健康, k) 生活满意度, l) 工作。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5271]

3.5031

妥善定义的过程 well-defined process

包含就绪准则、输入、进行工作的标准和规程、验证机制(例如同行评审)、输出和完成准则的过程。

参见:有效过程

[来源:GB/T 11457-2006 2.1846]

3.5032

假设情景分析 what-if scenario analysis

对多种情景进行评估,以预测其对项目目标产生的影响的过程。

[来源:ISO/IEC/IEEE FDIS 24765, 3.5272]

3.5033

“当”语句 WHILE

一种单入口、单出口的循环控制语句,其中循环控制在循环体之前执行。

参见:先导判定

[来源:GB/T 11457-2006 2.1847]

3.5034

白盒 white box; glass box; transparent box

内部内容或实现方式已知的系统或组件。

参见：黑盒

[来源：ISO/IEC/IEEE FDIS 24765, 3.2000]

3.5035

空格 `whitespace`

嵌入自由文本块中的非显示格式字符。

示例：空格和制表符。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5274]

3.5036

软件开发全过程 `whole process of software development`

软件开发中的需求、设计、编码和测试过程。

注：与 GB/T 11457-2006 中定义的开发过程相比，为便于可信指标的分类，本标准中的软件开发全过程专指需求、设计、编码和测试四个过程。

[来源：GB/T 37970-2019]

3.5037

宽带德尔菲法 `wideband Delphi`

一种估算方法，主题专家独立进行多轮估算，每轮后进行团队讨论，直至达成共识。

[来源：项目管理知识体系指南（PMBOK® 指南）— 第七版]

3.5038

维基 `wiki; wiki`

允许一组用户协同添加和编辑内容的网站。

[来源：ISO/IEC/IEEE 23026:2023, 3.1.30]

3.5039

窗口 `window`

具有可见边界、用于展示软件对象视图或供用户与计算机系统进行对话的区域。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5278]

3.5040

无线应用协议 `Wireless Application Protocol; WAP`

无线应用协议。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5251]

3.5041

不损害权利 `without prejudice`

不损害或放弃数据主体的权利、特权或雇佣条件。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5279]

3.5042

向导 `wizard`

通过与用户对话来指导用户完成任务的每个步骤的过程性帮助形式。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5280]

3.5043

字 `word`

在给定计算机中作为一个单元进行存储、编址、传输和操作的位序列或字符序列。

能够容纳位序列或字符序列的计算机存储单元。

在某种语言中具有含义并被视为一个实体的位序列或字符序列。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5282]

3.5044

工作授权 work authorization

通常为书面的许可和指示,用于开始特定进度活动、工作包或控制账户的工作。

注:工作授权是批准项目工作的方法,有助于确保工作由指定组织在正确时间以正确顺序完成。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5283, 有修改]

3.5045

工作授权系统 work authorization system

整个项目管理系统的子系统。它是一系列正式书面程序的集合,规定如何授权(委托)项目工作,以保证该工作由特定的组织、在正确的时间、以合理的顺序执行。工作授权系统包括发布工作授权所需的步骤、文件、跟踪系统及审批级别。

[来源: ISO/IEC/IEEE 24765:2017, 3.4602]

3.5046

工作分解结构词典 work breakdown structure; WBS

提供工作分解结构中每个组成部分的详细可交付成果、活动和进度安排信息的文档。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5284, 有修改]

3.5047

工作分解结构组件 work breakdown structure component

工作分解结构任意层级的条目。

[来源: ISO/IEC/IEEE 24765:2017, 3.4604]

3.5048

工作量 work effort

生产指定产出所需的人力资源。

注:人力资源通常以工时表示。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5286]

3.5049

工作包 work package

在工作分解结构的最低级别定义的工作,对其成本和工期进行估计和管理。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5287]

3.5050

工作绩效数据 work performance data

在执行项目工作的活动期间确定的原始观察结果和测量数据。

示例:包括完成的工作、关键绩效指标(KPI)、技术绩效度量、计划活动的开始和完成日期、变更请求数量、缺陷数量、实际成本、实际持续时间等。

[来源: ISO/IEC/IEEE 24765:2017, 3.4608]

3.5051

工作绩效信息 work performance information

从各种控制流程中收集的绩效数据,在相关背景下进行分析,并基于不同领域的相互关系进行整合。

[来源: ISO/IEC/IEEE 24765:2017, 3.4609]

3.5052

工作产品 work product

作为定义、维护或实施过程而产生的任何制品。

[来源：GB/T 45989-2025, 1.899]

3.5053

工作单位 work unit

项目任务，例如构建或测试功能点、用户故事、特性或需求。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5289]

3.5054

解决方法 workaround

在已知存在错误的情况下继续进行操作的方法。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5290]

3.5055

工作流程板 workflow board

在软件开发中，这是一种供开发人员使用的可视化工作展示方式，他们可以从任务待办事项列表中选取任务；这种展示方式常用于按需或资源受限的调度安排。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5291]

3.5056

工作流看板 workflow board; kanban board

在软件开发中，这是一种供开发人员使用的可视化工作展示方式，他们可以从任务待办事项列表中选取任务；这种展示方式常用于按需或资源受限的调度安排。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5291]

3.5057

工作协议 working agreement

敏捷环境中两个或多个参与方之间预期工作关系的声明。

[来源：ISO/IEC 33202:2024, 3.28]

3.5058

工作区域 working area

分配给计算机程序用于临时存储数据的主存储器的一部分。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5295]

3.5059

工作元模型 working metamodel

在 CDIF 传输的模型部分中可实例化的特定元对象的定义。

注：工作元模型包括传输模型部分中所引用的主题领域所使用的 CDIF 语义元模型中的元对象，以及在模型部分中定义为扩展的元对象。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5293]

3.5060

工作集 working set

在存储分配的分页方法中，对于程序执行过程中的任何特定时刻而言，最有可能驻留在主存储器中的页面集合。

[来源：ISO/IEC/IEEE FDIS 24765, 3.5294]

3.5061

工作空间 working space

工作区 working area

工作存储器 **working storage**

分配给计算机程序用于临时存储数据的主存储器的一部分。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5295]

3.5062

工作存储器 **working storage**

分配给计算机程序用于临时存储数据的主存储器的一部分。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5295]

3.5063

负载 **workload**

通常在给定计算机系统上运行的混合任务。

注: 主要特征包括输入/输出需求、计算量和种类以及所需计算机资源。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5296]

3.5064

负载模型 **workload model**

在计算机性能评估中所使用的模型, 用于描绘计算机系统中预期或实际工作负载的资源利用情况及性能指标。

参见: 系统模型

[来源: ISO/IEC/IEEE FDIS 24765, 3.5297]

3.5065

写 **write**

将位于功能流程内部的数据组移动到持久存储中的操作。

参见: 读取

[来源: ISO/IEC/IEEE FDIS 24765, 3.5299]

3.5066

写类型 **write type; write-type**

一种数据移动类型, 它将位于功能流程内部的数据组移动至持久存储区域。

注: 一次“写入”操作会被视为包含了实现该“写入”所需的一些相关数据操作。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5300]

3.5067

写作参考 **writing reference**

数据存储实体或其他记录, 或数据写入 BFC 的另一个软件或系统的接口记录。。

注: 在适用的所有 BFC 类型中, 写入引用的数量都大于 0。

[来源: ISO/IEC 29881:2010, 3.10]

3.5068

XML 模式定义 **XML schema definition; XSD**

基于 XML 的语言, 它为创建 XML 文档指定了一组规则和结构。描述 XML 信息结构的语言。

注: XML 文档遵循 XSD 定义中定义的所有规则, 以便被认为是有效文档。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5310]

3.5069

XML 模式文档 **XML schema document; XSD**

描述 XML 信息结构的文件。

[来源: ISO/IEC/IEEE FDIS 24765, 3.5312]

3. 5070

近期工作绩效报告 yesterday's weather

最近一个报告期的工作绩效报告。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5315]

3. 5071

零地址指令 zero-address instruction

不含地址字段的计算机指令。

参见: 单地址指令、双地址指令、三地址指令、四地址指令。

[来源: ISO/IEC/IEEE FDIS 24765, 3. 5316]

附 录 A

(资料性)

缩略语

APR:	年度百分比(annual percentage rate)
BRS:	业务需求规范(business requirements specification)
CAI:	关键应用项目(Critical application item)
CARD:	成本分析需求说明(Cost analysis requirements description)
CBT:	基于计算机的培训(Computer-based training)
CDD:	能力开发文件(Capability development document)
CFP:	COSMIC功能点(COSMIC function point)
CMMI:	能力成熟度模型集成(Capability Maturity Model Integration)
COOP:	运营连续性计划(continuity of operations plan)
CORBA:	通用对象请求代理体系架构(common object request broker architecture)
COS:	通用服务对象(common object services)
COTS:	商用货架产品(Commercial off-the-shelf (COTS))
CPC:	计算机程序部件(computer program component)
CPCI:	计算机程序配置项(computer program configuration item)
CPD:	持续专业发展(continuing professional development)
CPIF:	成本加激励费用(cost plus incentive fee)
CPLD:	复杂可编程逻辑器件(complex programmable logic device)
CPM:	关键路径法(critical path method)
CPPC:	成本加成本百分比(cost plus percentage of cost)
CPU:	中央处理器(central processing unit)
CRS:	共性与复用策略(commonality and reuse strategy)
CRUD:	创建、读取、更新、删除(create, read, update, delete)
CS:	组成系统(constituent system)
CSA:	配置状态核算(configuration status accounting)
CSCI;SWCI:	计算机软件配置项(computer software configuration item;software configuration item)
CSF:	关键成功因素(critical success factor)
CSN:	云服务合作伙伴(cloud service partner)
CSS:	层叠样式表(Cascading Style Sheets)
CSS2:	层叠样式表第二级(cascading stylesheets level 2)
CT:	通讯技术(communication technology)
CUS:	客户(customer)
CVE:	常见漏洞和暴露(common vulnerabilities and exposures)
CVSS:	常见漏洞评分系统(Common Vulnerability Scoring System)
DEMIL:	非军事化(demilitarization)
DIB:	目录信息库(directory information base)

DSM: 设计结构矩阵(Design Structure Matrix)
 ECO: 工程变更单(engineering change order)
 ECR: 工程变更请求(Engineering change request)
 ETA: 事件树分析(Event tree analysis)
 ETL: 抽取、转换、加载(extract, transform, load)
 EV: 挣值(earned value)
 FBSE: 基于功能的系统工程(Functions-based systems engineering)
 FCA: 功能配置审计(Functional configuration audit)
 FDT: 形式化描述技术(formal description techniques)
 FEMA: 失效模式与影响分析(failure mode and effect analysis)
 FEMCA: 失效模式、影响及危害性分析(failure mode, effects, and criticality analysis)
 FEP: 前端处理器(front-end processor)
 FFBD: 功能流框图(Functional flow block diagram)
 FHA: 功能危险性分析(Functional hazard analysis)
 FMECA: 故障模式、影响和危害性分析(Failure modes, effects, and criticality analysis)
 FOC: 全面运行能力(full operational capability)
 FoS: 系统族(Family of systems)
 FOSS: 免费开源软件(free and open source software)
 FP: 功能点(Function Point)
 FPA: 功能点分析(Function point analysis)
 FPGA: 现场可编程门阵列(field programmable gate array)
 FQR: 正式资格审查(Formal qualification review)
 FS: 完成至开始(finish-to-start)
 FSM: 功能规模测量(Functional size measurement)
 FTA: 故障树分析(Fault tree analysis)
 HLPNG: 高级Petri网(High-level Petri Net Graph)
 HMI: 人机界面(Human-machine interface)
 HWCI: 硬件配置(Hardware configuration item)
 IDEF: 功能建模的集成定义(integrated definition for functional modeling)
 Iff: 当且仅当(if and only if)
 IFPUG: 国际功能点用户组(International Function Point Users Group)
 ILS: 综合后勤保障(integrated logistics support)
 IOR: 互操作对象引用(Interoperable Object Reference)
 IoT: 物联网(internet of things)
 IPDT: 集成产品开发团队(integrated product development team)
 IXIT: 测试用实现补充信息(Implementation Extra Information for Testing)
 JCIDS: 联合能力集成与开发系统(Joint Capabilities Integration and Development System)
 KSA: 知识、技能和能力(Knowledge, skills, and abilities)
 LAI: 精益进步倡议(Lean Advancement Initiative)
 LefMEP: 工程项目集管理精益推动者(Lean Enablers for Managing Engineering Programs)
 LefSE: 系统工程精益推动者(Lean Enablers for Systems Engineering)
 LF: 最晚完成日期(late finish date)

LFT&E: 实弹测试与评估(live fire test and evaluation)
 LOE: 工作量(level of effort)
 LRIP: 小批量试生产(low-rate initial production)
 LS: 最晚开始日期(late start date)
 LTD: 等级测试设计(level test design)
 LTP: 等级测试计划(level test plan.)
 LTR: 等级测试报告(level test report)
 M&S: 建模与仿真(modeling and simulation)
 MAC: 介质访问控制(Media Access Control)
 MARR: 最低希望回报率(minimum attractive rate of return)
 MCDC: 修正条件/判定覆盖率(modified condition/decision coverage)
 MF: 测量框架(measurement framework)
 MFLOPS: 百万次浮点运算每秒(millions of floating-point operations per second)
 MGT: 管理(management)
 MIM: 管理信息模块(Management Information Model)
 MIS: 管理信息系统(management information system)
 ML: 机器学习(machine learning)
 MMC(S): 多媒体会议(系统)(Multimedia Conferencing (System))
 MMI: 人机界面(Man-machine interface)
 MODL: 元对象定义语言(Meta-Object Definition Language)
 MOPS: 每秒百万次运算(million operations per second)
 MOU: 谅解备忘录(memorandum of understanding)
 MPLa: 管理基准程序级别活动(manage benchmarking program level activity)
 MSP: 管理服务提供商(managed service provider)
 MSS: 管理体系标准(management system standard)
 MTBF: 平均无故障时间(mean time between failures)
 MTP: 总体测试计划(master test plan)
 MTR: 总体测试报告(master test report)
 MTTR: 平均修复时间(mean time to repair)
 NDI: 非发展项目(None developing item)
 NFSSM: 软件非功能规模度量(non-functional software size measurement)
 NVRAM: 非易失性随机存取存储器(non-volatile random access memory)
 OCD: 操作概念文件(operational concept document)
 ODP-RM: 开放分布式处理-参考模型(Open Distributed Processing: Reference Mode)
 OLA: 操作级协议(operations level agreement)
 OM: 组织管理(organizational management)
 OMT: 对象建模技术(Object modeling technique)
 OOD: 面向对象设计(Object-oriented design)
 OTRR: 运行测试就绪评审(operational test readiness review)
 p-use: 谓词使用(predicate data use)
 PDL: 程序设计语言(Program design language)
 PDPC: 过程决策程序图(Process decision program chart)

PIN: 个人识别号(Personal identification number)
PIXIT: 协议实施测试附加信息(Protocol Implementation Extra Information for Testing)
PN: Petri网(Petri Net)
QA: 质量保证(quality assurance)
QC: 质量控制(quality control)
QM-RM: 质量测量参考模型(quality measurement reference model)
RUP-SE: 面向系统工程的统一软件开发过程(rational unified process for systems engineering)
RUSP: 可直接使用软件产品(ready to use software product)
RUSP: 就绪可用软件产品(ready to use software product)
RVTM: 需求验证和可追溯矩阵(Requirements Verification and Traceability Matrix)
SP: 启动配置(Start provisioning)
SQAP: 软件质量保证计划(software quality assurance plan)
SQO: 服务质量目标(service qualitative objectives)
SQuaRE: 软件产品质量要求和评价(Software product Quality Requirements and Evaluation)
STEP: 产品模型数据交换标准(Standard for the Exchange of Product Model Data)
SWEBOK: 软件工程知识体系(Software Engineering Body of Knowledge)
SysML: 系统建模语言(systems Modeling Language)
UC: 使用控制(Use control)
UOD: 论域; 话语论域; 讨论域(Universe of discourse)

参 考 文 献

- [1] GB/T 22032-2021 系统与软件工程 系统生存周期过程
- [2] GB/T 25000.1-2021 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第1部分: SQuaRE 指南
- [3] GB/T 25000.20-2021 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第20部分: 质量测量框架
- [4] GB/T 25000.21-2019 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第21部分: 质量测度元素
- [5] GB/T 25000.2-2018 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第2部分: 计划与管理
- [6] GB/T 25000.22-2019 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第22部分: 使用质量测量
- [7] GB/T 25000.23-2019 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第23部分: 系统与软件产品质量测量
- [8] GB/T 25000.30-2021 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第30部分: 质量需求框架
- [9] GB/T 25000.40-2018 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第40部分: 评价过程
- [10] GB/T 25000.41-2018 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第41部分: 开发方、需方和独立评价方评价指南
- [11] GB/T 25000.45-2018 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第45部分: 易恢复性的评价模块
- [12] GB/T 36328-2018 信息技术 软件资产管理 标识规范
- [13] GB/T 36329-2018 信息技术 软件资产管理 授权管理
- [14] GB/T 36457-2018 复杂产品虚拟样机建模方法
- [15] GB/T 36964-2018 软件工程 软件开发成本度量规范
- [16] GB/T 37691-2019 可编程逻辑器件软件安全性设计指南
- [17] GB/T 37970-2019 软件过程及制品可信度评估
- [18] GB/T 38634.1-2020 系统与软件工程 软件测试 第1部分: 概念和定义
- [19] GB/T 38634.2-2020 系统与软件工程 软件测试 第2部分: 测试过程
- [20] GB/T 38634.3-2020 系统与软件工程 软件测试 第3部分: 测试文档
- [21] GB/T 38634.4-2020 系统与软件工程 软件测试 第4部分: 测试技术
- [22] GB/T 38639-2020 系统与软件工程 软件组合测试方法
- [23] GB/T 39788-2021 系统与软件工程 性能测试方法
- [24] GB/T 41865-2022 软件与系统工程 产品线工程与管理参考模型
- [25] GB/T 41866.1-2022 系统与软件工程 信息技术项目绩效基准度量框架 第1部分: 概念和定义
- [26] GB/T 41866.2-2022 系统与软件工程 信息技术项目绩效基准度量框架 第2部分: 基准度量要求
- [27] GB/T 41866.3-2022 系统与软件工程 信息技术项目绩效基准度量框架 第3部分: 报告编制

- [28] GB/T 41905-2022 软件与系统工程 软件测试工具能力
- [29] GB/T 42448-2023 系统与软件工程 功能规模测量 FiSMA1.1方法
- [30] GB/T 42452-2023 系统与软件工程 功能规模测量 COSMIC方法
- [31] ISO/IEC/IEEE FDIS 24765 Systems and software engineering — Vocabulary
- [32] 项目管理知识体系指南 (PMBOK® 指南) — 第七版
- [33] INCOME系统工程手册, 第五版

索引

汉语拼音索引

- A
- A/B 测试 3. 1
- Alpha 测试 3. 160
- 安排 3. 280
- 安全 3. 3790
- 安全标志 3. 3799
- 安全标准 3. 3791
- 安全处理需求 3. 3867
- 安全等级 3. 3878
- 安全断言 3. 671
- 安全分支 3. 3872
- 安全关键功能 3. 3792
- 安全关键软件 3. 3801
- 安全关键系统 3. 3802
- 安全哈希算法 (SHA) 3. 3866
- 安全集成 3. 3789
- 安全交互策略 3. 3876
- 安全控制 3. 3873
- 安全内核 3. 3877
- 安全权限 3. 3871
- 安全认证 3. 3869
- 安全生存周期 3. 3796
- 安全失效 3. 3875
- 安全示范 3. 3793
- 安全说明 3. 3797
- 安全完整性要求 3. 3795
- 安全验证 3. 3800
- 安全要求 3. 3798
- 安全与知识产权方案 3. 3870
- 安全域 3. 3874
- 安全责任 3. 3880
- 安全政策 3. 3879
- 安全注入信号 3. 3794
- 安装检验阶段 3. 2232
- 安装手册 3. 2233
- 安装与检验阶段 3. 2231
- 按成本设计 (面向成本的设计) 3. 1341
- 按单元调用 3. 582
- 按名调用 3. 580
- 按需调度 3. 2805
- 按需调页 3. 1280
- 按值调用 3. 583
- 按址调用 3. 582
- B
- (变迁的) 输入弧 3. 2211
- (变迁的) 输入库所 3. 2215
- Beta 测试 3. 472
- BFC 类 3. 473
- BFC 类型 3. 474
- 巴科斯范式 3. 414
- 跋断点 3. 1652, 3. 1653
- 白盒 3. 1980, 3. 5110
- 白盒测试 3. 1981, 3. 4329
- 百分位数 3. 2960
- 搬迁透明度 3. 3586
- 版本 3. 5047
- 版本标识符 3. 5050
- 版本化 3. 5051
- 版本控制 3. 5048
- 版本描述文件 3. 5049
- 版本说明文件 3. 5025
- 办公自动化 3. 2799
- 半双工 3. 2002
- 半形式验证 3. 3909
- 帮助系统 3. 2023
- 绑定 3. 488
- 绑定端点标识符 3. 490
- 绑定对象 3. 491
- 绑定前提条件 3. 492
- 绑定时间 3. 493
- 绑定时间决策 3. 494
- 绑定行为 3. 489
- 包 3. 418, 3. 2712, 3. 2913
- 包括 3. 2111
- 包容层次 3. 959
- 包容性 3. 2112
- 包图 3. 2914
- 保护 3. 3382
- 保护异常 3. 3383
- 保留款 3. 3684
- 保留字 3. 3648
- 保密协议 3. 2733
- 保密性 3. 900
- 保障 3. 4418
- 保证案例报告 3. 351
- 保证论据 3. 350
- 保证目标 3. 354
- 保证索赔 3. 352
- 保证信息 3. 353
- 报告 3. 3603
- 报告标准 3. 3605
- 报告绩效 3. 3604

报价请求	3. 3615	编译指令源语句	3. 786
悲观持续时间	3. 2997	编译指示	3. 3391
贝叶斯规则	3. 453	变更	3. 636
备份	3. 411	变更包	3. 643
备份及恢复测试	3. 412	变更管理	3. 642
备选方案分析	3. 164	变更集	3. 647
备选方案生成	3. 166	变更记录	3. 645
备选解决方案类	3. 165	变更控制	3. 637
备选流	3. 162	变更控制程序	3. 640
备用键	3. 163	变更控制计划	3. 639
备用冗余	3. 4248	变更控制委员会	3. 638
备用预案	3. 1825	变更请求	3. 646, 3. 3613, 3. 3717
被动对象	3. 2949	变更项目功能点计数	3. 644
被动互连	3. 2948	变更转储	3. 641
被审核方	3. 376	变化点	3. 5022
本地化	3. 2448	变换分析	3. 4806
本地化定制	3. 2443	变基	3. 3592
本地化软件资产管理负责人	3. 2445	变量定义	3. 1152
崩溃	3. 1080	变量追踪	3. 1206
比对测试	3. 410	变量踪迹	3. 5013
比较器	3. 772	变强度	3. 5012
比例尺度	3. 3474	变体	3. 5017
比特操纵	3. 497	变体关系	3. 2596
必需输入	3. 3618	变体选择	3. 5018
闭盒测试	3. 4204	辨别阈值	3. 1436
闭式子例程	3. 690	标度	3. 3817
闭项	3. 691	标号	3. 2380
闭项/基项	3. 692	标记	3. 2527
闭循环	3. 689	标记语言	3. 2530
边界	3. 518	标量类	3. 3814
边界箭头	3. 519	标量属性	3. 3816
边界值	3. 520	标量值属性	3. 3815, 3. 3816
边界值分析	3. 521	标签	3. 4524
编程	3. 3306	标签 ID	3. 4526
编程（动）	3. 3274	标签遍历	3. 4525
编程参考点	3. 3304	标识	3. 1878, 3. 2076
编程系统	3. 3309	标识符依赖实体	3. 1289
编程语言	3. 3307	标识关系	3. 2075
编辑	3. 1554	标题，报头，前导文件（头文件）	3. 2021
编辑程序	3. 1555	标注主题列表	3. 184
编码	3. 717, 3. 1592	标准	3. 1087
编码 1	3. 1593	标准操作程序/小外型尺寸封装	3. 4241
编译	3. 780	标准分部	3. 1458
编译并执行	3. 782	标准过程	3. 4242
编译程序	3. 783	标准化配置文件	3. 4244
编译程序代码	3. 784	标准技术英语	3. 4243
编译程序的编译程序	3. 785	标准实施器	3. 4245
编译程序的生成程序	3. 787	标准通用标记语言	3. 3973
编译后时间	3. 3057	表[格]	3. 4521
编译时间	3. 781	表达式	3. 1770

表驱动法	3. 4523	布尔签名	3. 511
表示	3. 3607	布局	3. 2388
表示标准	3. 3609	步进操作	3. 4293
表示属性	3. 3608	步骤	3. 4291
表头	3. 4522	部分簇	3. 2944
别名	3. 154	部分的	3. 2943
并发的	3. 879	部件, 构件	3. 801
并发进程	3. 884	部件测试	3. 807
并发任务架构	3. 886	部署	3. 1292
并发使能	3. 882	部署包	3. 1294
并发使能 (针对变迁模式)	3. 883	部署图	3. 1293
并发通信图	3. 880	C	
并发性	3. 878	(测量) 单位	3. 4874
并发子状态	3. 885	(测量结果的) 再现性	3. 3611
并行	3. 2929	(测量结果的) 重复性	3. 3594
并行阐述	3. 881	[程序] 对编	3. 1513
并行构造	3. 2931	CDIF 标识符	3. 623
并行类	3. 2930	CDIF 传输语法和编码	3. 631
并行运行操作	3. 2932	CDIF 导出工具	3. 620
波动性	3. 5088	CDIF 导入工具	3. 624
波形线	3. 4215	CDIF 明文编码	3. 619
波形音频文件格式	3. 5099	CDIF 图形符号	3. 622
补偿型决策技术	3. 776	CDIF 系列标准	3. 621
补偿型模型	3. 777	CDIF 传输	3. 628
补充权利项	3. 4393	CDIF 传输格式	3. 630
补充授权模式	3. 4392	CDIF 传输文件	3. 629
补充授权实体	3. 4392	CDIF 语义元模型	3. 627
补充信息结构	3. 4394	CDIF 元标识符	3. 625
补充运行	3. 4395	CDIF 元模型	3. 626
补丁	3. 2950	Chapin 图	3. 525
不变模式	3. 2320	裁剪	3. 4528
不变式	3. 2319	采购/采办报告	3. 54
不定交付不定量合同	3. 2126	采购/采办付款	3. 51
不断迭代	3. 989	采购/采办回复	3. 53
不符合	3. 2750	采购/采办记录	3. 52
不合格	3. 2750	采购/采办需要	3. 50
不可编程终端	3. 1515, 3. 2767	采购工作说明书	3. 3218
不可变类	3. 2082	采购管理计划	3. 3216
不可否认性	3. 2768	采购绩效审查	3. 3217
不可能区间	3. 2099	采购审核	3. 3214
不利后果	3. 125	采购文件	3. 3215
不确定	3. 4854	采用过程	3. 120
不确定性绩效域	3. 4855	菜单	3. 2577
不确定性锥	3. 899	菜单结构	3. 2578
不损害权利	3. 5117	参考 FSM 方法	3. 3522
不完全排错	3. 2087	参考架构	3. 3519
不完整的过程	3. 2114	参考结点	3. 3525
不一致率	3. 2115	参考框架	3. 3521
布尔表达式	3. 510	参考模式	3. 3524
布尔量	3. 509	参考位置	3. 2452, 3. 3526

参考信息	3. 3523	测量仪器	3. 2571
参考用户需求	3. 3530, 3. 3788	测量用户	3. 2570
参考用户需求集合	3. 3529	测试	3. 4691
参考有限状态机方法	3. 3522	测试报告	3. 4668
参考知识体系	3. 3520	测试参与者	3. 4656
参考转换点	3. 3528	测试策划过程	3. 4659
参数	3. 2933	测试策略	3. 4678
参数估计	3. 2938	测试策略与计划过程	3. 4679
参数化高级网络图	3. 2935	测试场景	3. 4671
参数化集合类	3. 2934	测试待查事件报告	3. 4635
参数图	3. 2937	测试单元	3. 4688
参数值对	3. 2936	测试独立性	3. 4637
参与式设计	3. 2946	测试对象	3. 4642
参与者	3. 89	测试对象版本	3. 4683
参与者属性	3. 2945	测试发生器	3. 4633
参照点	3. 3527	测试方法	3. 4594
参照完整性	3. 3531	测试方针	3. 4660, 3. 4661
残留控制	3. 3652	测试负责人	3. 4643
残余风险	3. 3653	测试覆盖（范围）	3. 4606
仓库膨胀	3. 3601	测试覆盖项	3. 4607, 3. 4608
仓库清理	3. 3602	测试工具	3. 4695
操作	3. 2823	测试管理	3. 4647
操作记录	3. 2832	测试管理流程	3. 4648
操作接口	3. 2831	测试规程	3. 4663
操作码	3. 2826	测试规格说明	3. 4675
操作实训	3. 4841	测试规格说明技术	3. 4676
操作手册	3. 2845	测试规则	3. 4665
操作数	3. 2819	测试过程	3. 4664
操作系统	3. 2822	测试后迭代	3. 3063, 3. 4888
操作异常	3. 2829	测试环境	3. 4622
操作员/维护人员培训材料	3. 2846	测试环境构建过程	3. 4627
操作员手册	3. 2844	测试环境项	3. 4624
操作约束	3. 2836	测试环境需求	3. 4626
操作字段	3. 2830	测试环境与数据管理过程	3. 4623
测度（名）	3. 2552	测试环境准备报告	3. 4625
测量	3. 2555	测试活动	3. 4621
测量标准	3. 2569	测试级别	3. 4644, 3. 4645
测量储藏库	3. 2566	测试集	3. 4673
测量单位	3. 4875	测试集体系结构	3. 4674
测量发起人	3. 2568	测试计划	3. 4658
测量方法	3. 2560	测试技术	3. 4684
测量分析师	3. 2556	测试监控过程	3. 4651
测量管理员	3. 2559	测试件	3. 4697
测量规程	3. 2563	测试脚本	3. 4672
测量过程	3. 2564	测试阶段	3. 4645, 3. 4657
测量过程负责人	3. 2565	测试接口	3. 4638
测量函数	3. 2558	测试接口层	3. 4639
测量经验库	3. 2557	测试结果	3. 4670
测量精度	3. 41	测试金字塔	3. 4693
测量模型	3. 2561	测试就绪评审	3. 4666

测试可重复性	3. 4667	测试子过程	3. 4680
测试框架	3. 4632	测试总结报告	3. 4604, 3. 4682
测试类	3. 4601	测试组织	3. 4655
测试类型	3. 4687	策略设置行为	3. 3044
测试描述	3. 4613	层	3. 2387
测试模型	3. 4649	层次分析法	3. 178
测试模型规格说明	3. 4650	层次建模	3. 2026
测试目标	3. 4652	层次结构图	3. 2029, 3. 4331
测试驱动程序	3. 4620	插图	3. 2080
测试驱动的开发	3. 4690	插装	3. 2242
测试驱动开发	3. 4560	插装[工具]	3. 2243
测试任务迭代	3. 4694	查平图	3. 651
测试日志	3. 4646	查询语言	3. 3463
测试设计	3. 4614, 3. 4618	差错	3. 1669
测试设计规格说明	3. 4616	差错播种	3. 1678
测试设计和实现过程	3. 4615	差额现金流量	3. 1396
测试设计技术	3. 4617	差量	3. 1279
测试生成器	3. 4599	差异分析	3. 5014
测试实践	3. 4662	拆分运行测试	3. 1
测试事件	3. 4634	产品安全标签	3. 3260
测试事件报告过程	3. 4636	产品标识	3. 3231
测试数据	3. 4610	产品标准	3. 3265
测试数据生成器	3. 4599, 3. 4611	产品待办列表	3. 3223
测试数据准备报告	3. 4612	产品范围	3. 3261
测试说明	3. 4692	产品范围界定	3. 3263
测试台	3. 4596	产品范围说明	3. 3262
测试条件	3. 4605	产品分解结构	3. 3226
测试完成报告	3. 4603, 3. 4604	产品分析	3. 3220
测试完成过程	3. 4602	产品风险	3. 3258
测试文档集	3. 4619	产品改进	3. 3232
测试项	3. 4640, 3. 4642	产品工程	3. 3230
测试项传输报告	3. 4641	产品管理	3. 3253
测试需求	3. 4669	产品规格	3. 3264
测试依据	3. 4595	产品基线	3. 3224, 3. 3225
测试用例	3. 4597	产品库	3. 3234
测试用例的可变性	3. 4996	产品路线图	3. 3259
测试用例规格说明	3. 4600	产品描述	3. 3229
测试用例生成器	3. 4598, 3. 4599	产品配置标识	3. 3228
测试用例套	3. 4681	产品配置基线	3. 3224
测试有效性	3. 4696	产品权威	3. 3222
测试与评估	3. 4520	产品认证	3. 3227
测试预言	3. 4653	产品生存周期	3. 3235
测试预言问题	3. 4654	产品实例	3. 3233
测试执行	3. 4628	产品线	3. 3236
测试执行过程	3. 4631	产品线采用场景	3. 3238
测试执行日志	3. 4630	产品线采用计划	3. 3237
测试执行引擎	3. 4629	产品线参考模型	3. 3249
测试状态报告	3. 4677	产品线测试策略	3. 3251
测试追踪矩阵	3. 4685, 3. 4686, 3. 4780	产品线产品系列	3. 3248
测试准则	3. 4609	产品线范围界定	3. 3250

产品线工程工厂	3. 3243	成熟度等级	3. 2536
产品线工程工厂开发环境	3. 3245	成熟度模型	3. 2537
产品线工程工厂配置器	3. 3244	成熟度提问单	3. 2538
产品线架构	3. 3239	成熟性	3. 2535
产品线配置	3. 3240	成文信息	3. 1472
产品线配置管理	3. 3242	成效导图	3. 2084
产品线配置项	3. 3252	成员产品	3. 2573
产品线配置增量	3. 3241	承包商	3. 4399
产品线平台	3. 3247	承诺	3. 749
产品线制度化	3. 3246	承受力	3. 3655
产品要求	3. 3257	程序	3. 3274
产品拥有者	3. 3255	程序保护	3. 3291
产品支持	3. 3266	程序插装	3. 3284
产品指标	3. 3254	程序段, 存储段, 分段	3. 3887
产品质量	3. 3256	程序规范	3. 3292
产品资产实例	3. 3221	程序计数器	3. 3278
产品资料库	3. 3269	程序架构	3. 3275
常量	3. 945	程序结构	3. 3277
常驻控制程序	3. 3651	程序结构图	3. 4331
场景	3. 3820	程序库	3. 3285, 3. 4084
场景测试	3. 3822	程序块	3. 3276
场景脚本	3. 3821	程序扩展	3. 3282
超级计算机	3. 4385	程序列表	3. 3286
超类	3. 4384	程序敏感故障	3. 3310
超类型	3. 4386	程序设计语言	3. 3279
超链接	3. 2058	程序设计支持环境	3. 3308
超时	3. 4730	程序突变	3. 3289
超文本标记语言	3. 2059	程序网络图	3. 3290
超文本传输协议	3. 2060	程序维护手册	3. 3287
撤销	3. 3714	程序系列	3. 1828
沉没成本	3. 4383	程序员手册	3. 3305
陈旧	3. 1509	程序支持库	3. 4064
成本补偿合同	3. 1072	程序指令	3. 3283
成本度量	3. 1062	程序状态字	3. 3293
成本管理计划	3. 1061	程序综合	3. 3294
成本规避	3. 1056	持久存储器	3. 2987
成本汇总	3. 1055	持久化统一资源标识符	3. 2988
成本基础	3. 1058	持久性	3. 2984
成本基准	3. 1057	持久性测试	3. 1603
成本绩效基准	3. 1064	持久性对象	3. 2986
成本绩效指数	3. 1065	持久性模式	3. 2985
成本加费用	3. 1071	持续部署	3. 986
成本加固定费用合同	3. 1067	持续迭代	3. 989
成本加激励费用合同	3. 1068	持续风险管理	3. 991
成本加奖励费用合同	3. 1066	持续改进	3. 987
成本偏差	3. 1069	持续过程改进	3. 983
成本效益分析	3. 1070	持续集成	3. 988
成本约束	3. 1059	持续交付	3. 985
成功标准	3. 4378	持续时间	3. 1517
成品项	3. 1594	冲刺	3. 4211

冲刺待办列表	3. 4212	处置约束	3. 1440
冲突	3. 929	触发器	3. 1880, 3. 1881
冲突管理	3. 930	触发事件	3. 4821, 3. 4822
抽象	3. 19	触发事件类型	3. 4822
抽象层级	3. 2402	触发条件	3. 4820
抽象机	3. 18	穿通箭头	3. 4835
抽象类	3. 15	传出的	3. 1561
抽象设计	3. 17	传入的	3. 130
抽象数据类型	3. 16	传输轮廓	3. 4773
抽象行为	3. 14	传输文件	3. 4803
出错	3. 1669	传送	3. 4802
出错分析	3. 1670	串行的	3. 3920
出错类别	3. 1671	串行构造	3. 3923
出错模型	3. 1675	串行结构	3. 3924
出错数据	3. 1672	串行扩展接口	3. 3925
出错消息	3. 1674	串行通信接口	3. 3922
出错预测	3. 1676	串行外设接口 (SPI)	3. 3926
出错预测模型	3. 1675	串行总线接口	3. 3921
出口	3. 1748	窗口	3. 5115
出口点	3. 1751	创建	3. 1084
出口例程	3. 1752	创建 WBS (工作分解结构)	3. 1083
初步 MOE 数据	3. 3083	创建工作分解结构	3. 1082
初步 TPM 数据	3. 3085	创意导图	3. 2071
初步的 TPM 需求	3. 3086	创作环境	3. 381
初步的生存周期概念	3. 3082	垂直距离测量	3. 3041
初步接口定义	3. 3081	词典式决策规则	3. 2409
初步设计	3. 3079	词汇	3. 5086
初步验证标准	3. 3087	次级子特征	3. 4375
初期功能点分析	3. 2196	从当前分支合并	3. 2580
初始段	3. 2202	从属实体	3. 1288
初始风险	3. 2201	从属状态	3. 3141
初始功能点计数	3. 2197	从头到头	3. 4252
初始功能文件	3. 2063	从头到尾	3. 4251
初始化	3. 2203	存储	3. 4304, 3. 4305
初始授权项	3. 2195	存储断点	3. 1141, 3. 4301
初始投资	3. 2198	存储分配	3. 4300
储备	3. 3645, 3. 3647	存储器	3. 4299
储备分析	3. 3646	存储容量	3. 4302
储藏库	3. 3606	存储设备	3. 4304
处理 (动词)	3. 110	存储效率	3. 4303
处理逻辑	3. 3211	存货	3. 407
处理器	3. 3212	存在依赖	3. 1746
处理证据	3. 3174	存在约束	3. 1745
处置	3. 1439	措施建议	3. 60
处置报告	3. 1444	措施项	3. 58
处置策略	3. 1445	错误	3. 2634
处置程序	3. 1442	错误猜测	3. 1673
处置记录	3. 1443	错误处理	3. 1677
处置启用系统需求	3. 1441		
处置系统	3. 1446		
		D	
		“当” 语句	3. 5109

DDR2 SDRAM	3. 1223	当前范围	3. 1780
DDR3	3. 1224	当前外延	3. 1781
DDR4	3. 1225	档案	3. 276
达尔文信息分类体系结构	3. 1132	档案页面	3. 275
大致接受	3. 1972	导出参与方	3. 1302
大中取大准则	3. 2539	导出测度	3. 1299
代价函数	3. 1060	导出过程	3. 1769
代理	3. 134	导出类型	3. 1304
代码	3. 706	导出属性	3. 1302
代码冻结	3. 710	导出特征	3. 1302
代码断点	3. 708	导出元素	3. 1298
代码评审	3. 713	导航	3. 2731
代码审查	3. 712	导航辅助	3. 2732
代码审核	3. 707	导入代理	3. 2098
代码生成程序	3. 711	导入流程	3. 2097
代码数据	3. 709	德尔菲法	3. 1278
代码行	3. 2426	地点标记	3. 2529
代码优化	3. 715	地址(名词)	3. 110
代码踪迹	3. 714, 3. 1740	地址部分	3. 111
代码走查	3. 716	地址格式	3. 112
代数语言	3. 149	地址空间	3. 114
带应答的同步消息通信	3. 4434	地址修改	3. 113
带有回复的同步消息通信	3. 4435	地址字段	3. 111
带有经济价格调整条款的固定总价合同	3. 1876	登记册	3. 3537
待办改进事项	3. 2101	登录	3. 2458
待办清单细化	3. 408	等概率期望值	3. 1756
待机模式	3. 4246, 3. 4247	等级	3. 1989
单步操作	3. 4004	等级, 层级	3. 2401
单步执行	3. 4003	等级测试日志	3. 2482
单操作数指令	3. 4001	等价划分	3. 1659
单代号网络图	3. 3068	等价类	3. 1658
单地址指令	3. 3998	等价类覆盖率	3. 1660
单点估算	3. 4002	等价类划分	3. 1661
单调速率算法	3. 3470	等时性	3. 2326
单个界面	3. 2145	等效 IDL	3. 1663
单规则决策表	3. 3999	等效故障	3. 1662
单级编码	3. 4000	等效接口	3. 1664
单启动	3. 3995	低层级	3. 2477
单体执行器	3. 2686	低层级关键字	3. 2480
单一继承	3. 3996	低层级设计	3. 2479
单一实例方法	3. 3997	低层类	3. 2478
单元	3. 4873	低级语言	3. 305
单元测试	3. 4878, 3. 4880	敌对回退	3. 2046
单元测试框架	3. 4879	递归	3. 3510
单元开发文件夹	3. 4061	递归的	3. 3511
单元需求文档集	3. 4877	递归例程	3. 3512
单值的	3. 4005	第二代语言	3. 305
单值属性	3. 4006	第二范式	3. 3862
弹出窗口	3. 3046	第三范式	3. 4708
弹性	3. 1564	第四代语言	3. 1918

第五代语言	3. 1858	定义使用对	3. 1264
第五代语言 (5GL)	3. 1859	定义使用路径	3. 1265
第一范式	3. 1871	定义性模型	3. 1263
第一个输入例程	3. 1870	定义语言	3. 1459
点标记法	3. 1501	定制	3. 1120
电可擦除可编程只读存储器	3. 1565	定制化	3. 1119
电信信息网络体系结构	3. 4747	定制化过程	3. 4527
电子表格程序	3. 4210	定制软件	3. 1114
电子设计自动化	3. 1567	动词词组	3. 5030
电子数据交换	3. 1566	动态绑定	3. 1526
调度程序	3. 3839	动态测试	3. 1544
调试	3. 744, 3. 1230	动态差错	3. 1532
调页程序	3. 2925	动态产品	3. 1535
调用	3. 579	动态程序分析	3. 1536
调用表	3. 586	动态存储分配	3. 1543
调用参考	3. 587	动态的	3. 1521
调用传递	3. 2323	动态调用	3. 1533
调用跟踪	3. 4368	动态调用接口	3. 1406
调用框	3. 590	动态断点	3. 1527
调用树	3. 585	动态分类	3. 1530
调用顺序	3. 591	动态分配	3. 1522
调用提交	3. 2324	动态分析	3. 1523
调用图	3. 584, 3. 585	动态分析器	3. 1524
调用图表	3. 589	动态骨架	3. 1542
调用踪迹	3. 588	动态缓冲	3. 1528
调整规模	3. 118	动态模式	3. 1541
调整提前量与滞后量	3. 119	动态模型	3. 1534
迭代	3. 2342	动态属性	3. 1525
迭代待办事项	3. 2343	动态随机存取存储器	3. 1537
迭代方法	3. 2347	动态重定位	3. 1538
迭代规划	3. 2345	动态重构	3. 1540
迭代过程	3. 2351	动态转储	3. 1531
迭代计划	3. 2344	动态资源分配	3. 1539
迭代结构	3. 2348	动态总线定宽	3. 1529
迭代开发	3. 2349	动作, 措施	3. 56
迭代评审	3. 2346	动作链 / 任务链	3. 634
迭代生存周期	3. 2350	动作签名	3. 62
顶点	3. 5052	动作清单	3. 64
顶级	3. 4760	动作条目	3. 57
订阅式许可证	3. 4369, 3. 4371	动作序列	3. 61
定界符	3. 1272	动作状态	3. 63
定量风险分析	3. 3461	冻结分支	3. 1929
定量控制	3. 3460	冻结准备阶段	3. 4012
定时脉冲单元	3. 4741	独立的	3. 2128
定时器事件	3. 4740	独立的, 单立的	3. 4240
定位点	3. 183	独立估计	3. 2129
定位机制	3. 2449	独立评估方	3. 2130
定性风险分析	3. 3417	独立实体	3. 1635
定义范围	3. 1261	独立文档	3. 3913
定义活动	3. 1260	独立型活动	3. 1432

独立性	3. 2127	多值属性	3. 2713
独立验证和确认	3. 2132	多重集	3. 2712
独立验证和确认 (IV&V)	3. 2133	多重集基数	3. 2711
独立验证和确认 (IV&V)	3. 2134	多准则决策	3. 2708
独立状态类	3. 2131	多准则决策分析	3. 2700
独特的功能	3. 4871	多组件系统	3. 2697
读	3. 3483	E	
读弧	3. 3482	ELSE 规则	3. 1571
读取类型	3. 3483	二地址指令	3. 1504, 3. 4843
读引用	3. 3486	二级数据	3. 3863
度量绩效域	3. 2562	二加一地址指令	3. 4847
度量来源	3. 2567	二进制数字[位]	3. 485
端口	3. 3047	二元关联	3. 484
短消息服务	3. 3950	F	
断点	3. 538	F1 分数	3. 1805
断言	3. 307, 3. 3072	FSM 方法的拥有者	3. 2912
队列	3. 3464	F 型规范	3. 1919
对称网	3. 4427	发布	3. 3560
对等软件	3. 2959	发布包	3. 3563
对话	3. 1395	发布待办事项清单	3. 3561
对角微指令	3. 1386	发布地图	3. 3562
对抗性测试	3. 124	发布规划	3. 3565
对抗性攻击	3. 122	发布计划	3. 3564
对抗性实例	3. 123	发布流水线	3. 3401
对象标识符	3. 2788	发布信息	3. 1447
对象过程法	3. 2847	发射器	3. 1579
对象集	3. 2794	翻译程序	3. 4814
对象类型	3. 2795	反编译	3. 1246
对象类型集	3. 2796	反编译程序	3. 1247
对象流状态	3. 2787	反常	3. 191
对象生存线	3. 2791	反汇编	3. 1425
对象实现	3. 2789	反汇编程序	3. 1426
对象图	3. 2786	反汇编器	3. 1228
对象引用	3. 2793	反向传播	3. 416
对应关系	3. 1054	反应性	3. 3481
多[重]继承	3. 2705	反映型构念	3. 3533
多点估算	3. 2710	返工	3. 3716
多读写者	3. 2707	返回	3. 3691
多对多关系	3. 2522	返回码	3. 3692
多分类代数	3. 2521	返回值	3. 3695
多核	3. 2698	泛化	3. 1969, 3. 4885
多核处理器	3. 2699	泛化分类法	3. 1971
多级安全	3. 2703	泛化结构	3. 1970
多级存储器	3. 5075	范围管理计划	3. 3848
多命中判定表	3. 2709	范围基线	3. 3846
多模态生物特征识别系统	3. 2704	范围蔓延	3. 3847
多实例方法	3. 2706	范围确认	3. 4955
多属性决策	3. 2696	方程法	3. 1656
多维结构	3. 2701	方法	3. 2601
多学科仿真	3. 2702	方法标准	3. 2603

方法工程师	3. 2602	非主要实体	3. 2764
方法论	3. 2604	分布偏移	3. 1454
方法论元素	3. 2605	分布式处理	3. 1450
方框	3. 522	分布式计算	3. 1448
方框 ICOM 代码	3. 526	分布式拒绝服务攻击	3. 1449
方框编号	3. 528	分布式透明性	3. 1452
方框名	3. 527	分步说明	3. 4292
方框图	3. 525	分层[结构]	3. 2028
方框细节参考	3. 523	分层分解	3. 2025
方面	3. 298	分层连续	3. 2027
防御性编程	3. 1259	分层语言	3. 4318
仿真	3. 3991	分段执行器	3. 3888
仿真用户	3. 1583	分发单元	3. 1453
仿真组件	3. 3992	分发副本	3. 1451
访谈	3. 2315	分割操作	3. 1457, 3. 4185
访问	3. 29	分级表示	3. 4224
访问(存取)控制机制	3. 34	分解	3. 1249, 3. 1810
访问方法	3. 31	分解器	3. 1248
访问例程	3. 32	分解图	3. 1250
访问设施	3. 30	分块因子	3. 505
访问透明度	3. 33	分类	3. 611, 3. 675, 3. 4169, 3. 4171
放弃书	3. 5091	分类空间	3. 613
非标识关系	3. 2761	分类器	3. 679
非补偿性决策技术	3. 2748	分类树	3. 677
非补偿性模型	3. 2749	分类树法	3. 678
非发展项目	3. 2734	分类学	3. 4558
非分层语言	3. 4886	分类轴	3. 676
非功能规模	3. 2759	分离键	3. 4207
非功能规模度量	3. 2744	面搜索	3. 1807
非功能性需求	3. 2757	分配	3. 158
非过程编程语言	3. 2766	分配的基线	3. 155
非过程语言	3. 2765	分配的努力	3. 239
非技术需求	3. 2769	分配的标识	3. 156
非键属性	3. 2763	分配给软件的系统需求	3. 4496
非交付项	3. 2751	分配需求	3. 157
非交付源语句	3. 2752	分时	3. 4732
非内在关系	3. 2762	分析	3. 173
非确定性系统	3. 2753	分析技术	3. 179
非人力资源成本	3. 2760	分析阶段	3. 175
非时间关键型计算密集任务	3. 2771	分析模型	3. 174
非研制的	3. 2755	分析情境	3. 176
非易失性存储器	3. 2772	分析师	3. 177
非易失性随机存取存储器	3. 1879	分页, 调页	3. 2926
非预期用户	3. 4869	分支	3. 530, 3. 534
非正式审查	3. 2150	分支操作	3. 1896
非正式小组审查	3. 2149	分支测试	3. 533
非终结符	3. 2770	分支点	3. 535
非终止性故障	3. 2780	分支合并	3. 727
非周期任务	3. 195	分支条件测试	3. 532
非周期性任务	3. 196	分支条件组合测试	3. 531

风险报告	3. 3750	服务导入	3. 3941
风险暴露	3. 3733	服务等级	3. 2407
风险标准	3. 3729	服务等级目标(SLO)	3. 3945
风险承受能力	3. 3755	服务等级协议	3. 3942
风险处理	3. 3758	服务等级协议满意度	3. 3944
风险处置措施	3. 3759	服务等级协议信息完整性	3. 3943
风险触发	3. 3760	服务对象	3. 3928
风险触发器	3. 4823	服务管理	3. 3947
风险等级	3. 2406	服务管理体系	3. 3948
风险调整后的待办清单	3. 3762	服务管理系统	3. 3950
风险分担	3. 3752	服务管理系统(SMS)	3. 3949
风险分解结构	3. 3727	服务兼容性	3. 3934
风险分析	3. 3723	服务交付策略	3. 3936
风险概况	3. 3745	服务交付配置文件	3. 3937
风险杠杆因子	3. 3737	服务接口信息	3. 3953
风险跟踪表	3. 3749	服务节点	3. 3929
风险管理	3. 3738	服务经理	3. 3951
风险管理计划	3. 3739	服务可测量性	3. 3952
风险管理流程	3. 3740	服务可供性	3. 3957
风险规避	3. 3726	服务可靠性	3. 3960
风险缓解	3. 3742	服务可维护性	3. 3946
风险缓解措施	3. 3741	服务可用性	3. 3965
风险价值	3. 4964	服务目录	3. 3932
风险监控	3. 3743	服务请求	3. 3961
风险降低措施	3. 3748	服务设计	3. 3938
风险接受	3. 3721	服务式许可证	3. 4371
风险紧迫性评估	3. 3761	服务台	3. 3939
风险类别	3. 3728	服务提供系统	3. 3956
风险利用	3. 3732	服务替代方案	3. 3931
风险偏好	3. 3724	服务协议	3. 3930
风险审查	3. 3751	服务型领导	3. 3927
风险审核	3. 3725	服务性能效率	3. 3954
风险升级	3. 3731	服务需求	3. 3962
风险识别	3. 3736	服务原语	3. 3955
风险行动请求	3. 3722	服务支持性	3. 3964
风险因素	3. 3734	服务质量	3. 3451
风险应对措施	3. 3735	服务质量定性目标	3. 3958
风险阈值	3. 3754	服务质量特性	3. 3959
风险源	3. 3753	服务组件	3. 3935
风险再评估	3. 3746	浮动时间	3. 1882
风险责任人	3. 3744	符号变更覆盖率	3. 3981
风险增强	3. 3730	符号表	3. 4422
风险转嫁	3. 3756	符号地址	3. 4423
风险转移[策略]	3. 3757	符号语言	3. 4425
封装	3. 1591	符号执行	3. 4424
峰值测试	3. 4205	符号踪迹	3. 4426
服务	3. 3966	符合性	3. 931
服务安全性	3. 3963	符合性评价	3. 935
服务变更请求	3. 3933	符合性评价报告	3. 936
服务导出	3. 3940	福祉	3. 5104

福祉影响评估	3. 5105	概念数据模型	3. 871
福祉指数	3. 5106	概念系统设计	3. 874
辅窗口	3. 3865	概念信息	3. 872
辅用户	3. 3864	概念验证	3. 3374
辅助技术	3. 337	概要设计评审	3. 3080
辅助类	3. 399	赶工	3. 1081
负载	3. 5139	感知参考点	3. 2961
负载模型	3. 5140	干扰	3. 1455
附加进程	3. 364	高层功能点分析	3. 2032
附加性支持	3. 108	高层管理	3. 2033
复测	3. 3685, 3. 3686	高层级关键字	3. 2035
复合(组合)类型	3. 817	高层设计	3. 2034
复合对象	3. 814	高层网	3. 2036
复合关键词	3. 812	高级 Petri 网图	3. 2037
复合任务	3. 816	高级建模方法	3. 2031
复合组件模型	3. 808	高级模型	3. 2030
复位	3. 3649	高级配置文件	3. 121
复位弧	3. 3650	高级语言	3. 2038
复用	3. 3700	高阶语言(HOL)	3. 2039
复用来源声明	3. 3701	隔离	3. 2327
复杂产品	3. 795	个人防护装备	3. 2994
复杂度矩阵	3. 799	个人计算机	3. 2991
复杂可编程逻辑器件	3. 796	个人计算机(PC)	3. 2992
复杂可编程逻辑器件(CPLD)	3. 797	个人可识别信息主体	3. 1204
复杂性	3. 798	个人身份信息	3. 2995
复制	3. 1042, 3. 3598	个人数据	3. 2993
复制模式	3. 3599	个人信息	3. 3232
复制透明度	3. 3600	个人信息保护和电子文件法	3. 3007
副作用	3. 3980	根本原因分析	3. 3776
赋值	3. 335	根编译程序	3. 3777
赋值语句	3. 336	根源	3. 3775
覆盖	3. 2906	跟踪链接	3. 4777
覆盖监督程序	3. 2907	跟踪链接语义	3. 4778
覆盖率等级	3. 3987	工厂	3. 1811
覆盖项	3. 1079, 3. 4608	工程	3. 1605
G		工程变更	3. 1606
(过渡的)输出位置	3. 2896	工程变更建议	3. 1607
(过渡段的)输出弧	3. 2894	工程变更请求	3. 1608
<个人>能力	3. 3	工程化系统	3. 1604
改进建议	3. 3380	工程接口参考	3. 1611
改正性维护	3. 1051	工程接口参考管理策略	3. 1614
改正性支持	3. 1052	工程接口参考管理领域	3. 1612
概率	3. 3135	工程接口引用管理策略	3. 1615
概率估算	3. 3133	工程接口引用管理域	3. 1613
概率和影响矩阵	3. 3136	工程经济学	3. 1609
概率系统	3. 3134	工程视点	3. 1616
概念分析	3. 866	工程业绩	3. 3340
概念阶段	3. 870	工程组	3. 1610
概念模型	3. 873	工具	3. 4756
概念漂移	3. 867	工具桥接器	3. 4757

工具箱	3. 4758	功能基线	3. 1944
工具专用信息	3. 4759	功能建模的集成定义	3. 2251
工料合同	3. 4725	功能类型	3. 1940
工艺开发	3. 3172	功能目录	3. 1846
工作包	3. 5125	功能配置标识	3. 1947
工作产品	3. 5128	功能配置审核	3. 1946
工作存储器	3. 5137, 3. 5138	功能清单	3. 482
工作单位	3. 5129	功能清单组合	3. 483
工作分解结构词典	3. 5122	功能设计	3. 1948
工作分解结构组件	3. 5123	功能适合性	3. 1943
工作集	3. 5136	功能适应性	3. 1941
工作绩效数据	3. 5126	功能危险性分析	3. 1950
工作绩效信息	3. 5127	功能系统设计	3. 1957
工作空间	3. 5137	功能性技术状态审计	3. 1945
工作量	3. 5124	功能性需求	3. 1952
工作流程板	3. 5131	功能性用户需求 (FUR)	3. 1959
工作流看板	3. 5132	功能用户	3. 1958
工作区	3. 5137	功能域的适用性	3. 197
工作区域	3. 5134	功能域分类	3. 1949
工作授权	3. 5120	功能字段	3. 1933
工作授权系统	3. 5121	攻击	3. 365
工作说明书	3. 4264	供方	3. 4397
工作协议	3. 5133	供应报告	3. 4403
工作元模型	3. 5135	供应策略	3. 4405
公共存储	3. 758	供应付款	3. 4401
公共环境耦合	3. 761	供应记录	3. 4402
公共耦合	3. 755	供应链风险管理	3. 3853
公共设施软件	3. 4950	供应商	3. 4399
公共数据	3. 756	供应商分支	3. 5028
公开的	3. 3397	供应商服务上下文代码集标识符	3. 5029
公用事业	3. 4949	供应系统	3. 4396
公有链	3. 3398	供应响应	3. 4404
公有云	3. 3399	供应协议	3. 4400
功利主义	3. 4948	共存性	3. 719
功能	3. 1932	共情	3. 1581
功能测试	3. 4204	共情自主智能系统	3. 1580
功能尺寸	3. 1953	共同特征	3. 757
功能点	3. 1939	共同原因 随机原因	3. 753
功能点 (FP)	3. 1934	共同祖先约束	3. 751
功能点表	3. 1938	共享资产	3. 3975
功能点分析	3. 1936	共享资产超集	3. 3976
功能点分析 (FPA)	3. 1935	沟通管理	3. 764
功能点耗时率	3. 2989	沟通规划	3. 767
功能点数	3. 1937	沟通限制	3. 762
功能分析	3. 1942	构成系统	3. 948
功能规格	3. 1956	构件标准	3. 806
功能规模测量	3. 1955	构件图	3. 803
功能规模测量方法	3. 1954	构建	3. 551, 3. 953
功能规模度量范围	3. 3849	构建过程	3. 552
功能过程	3. 1951	构建流水线	3. 1368

构造	3. 951, 3. 4704, 3. 4705	关键项	3. 1094
构造的能力	3. 952	关键信息	3. 1093
购得系统	3. 45	关键性	3. 1104
购置成本	3. 1058	关键性能参数	3. 2366
估算	d	关键业务服务	3. 5084
估算的功能点计数	3. 1687	关键资产	3. 1088
估算的基础	3. 448	关键字	3. 2370
估算方法	3. 1688	关键字测试用例	3. 2373
孤立页面	3. 2888	关键字库	3. 2371
固定成本	3. 1874	关键字驱动测试	3. 2374
固定工期	3. 1875	关键字驱动测试框架	3. 2375
固定总价合同	3. 1868	关键字执行代码	3. 2372
固定总价加激励费用合同	3. 1877	关键字字典	3. 2371
固件	3. 1869	关联	3. 338
故事地图	3. 4306	关联端	3. 340
故事点	3. 4307	关联管理设施	3. 341
故事映射	3. 4308	关联类	3. 339
故障	3. 1832, 3. 1840	关联实体	3. 343, 3. 1634
故障安全	3. 1836	关联实体类型	3. 344
故障播种	3. 1678	关联字面量	3. 345
故障插入	3. 1837	关系	3. 3548
故障隔离	3. 1834	关系经理	3. 3552
故障恢复	3. 1822	关系类型	3. 3554
故障类别	3. 1671	关系名称	3. 3553
故障模式和影响分析	3. 1820	关系实例	3. 3551
故障平均恢复时间	3. 2693	关系数据库管理系统	3. 3549
故障屏蔽	3. 1835	关注点	3. 875
故障容忍 (名词)	3. 1838	观点	3. 3041
故障弱化	3. 1813	管理储备	3. 2509
故障树分析	3. 1839	管理独立性	3. 2513
故障透明性	3. 1824	管理过程	3. 2508
故障注入	3. 1837	管理技能	3. 2511
故障字典	3. 1833	管理角色	3. 2514
顾客	3. 1115	管理评审	3. 2510
雇员	3. 1582	管理系统	3. 2512
挂起准则	3. 4417	管态模式	3. 1020
关闭后的活动	3. 3056	光盘	3. 2851
关键部分优先	3. 1097	广义逻辑操作	3. 1782
关键的, 临界的	3. 1105	广义数学操作	3. 1783
关键过程域	3. 2368	归档	3. 277
关键计算机资源	3. 1090	归纳断言法	3. 2146
关键链法	3. 1089	规程测试	3. 3147
关键路径	3. 1095	规范	3. 2773
关键路径法	3. 1096	规范化	3. 2774
关键软件	3. 1101	规格变更通知	3. 4199
关键设计评审	3. 1091	规格说明, 规约	3. 4198
关键设计评审 (CDR)	3. 1092	规格说明变更通知	3. 3845
关键实践	3. 2367	规格说明树	3. 4201
关键属性	3. 1098	规格说明验证	3. 4202
关键系统	3. 1102	规格说明语言	3. 4200

规划	3. 3026	过程属性	3. 3158
规划范围	3. 3027	过程属性结果	3. 3159
规划干系人参与	3. 3020	过程属性评级	3. 3160
规划干系人管理	3. 3021	过程所有者	3. 3191
规划过程组	3. 3030	过程维度	3. 3173
规划资源管理	3. 3015	过程相关风险	3. 3213
规模估计	3. 4008	过程行动计划	3. 3149
规模综合单价	3. 4876	过程行动组	3. 3150
规则	3. 3779	过程性能	3. 3192
滚动式规划	3. 3774	过程性能指标	3. 3194
过程	3. 3148	过程语言	3. 3144
过程/规程	3. 3146	过程证据要素	3. 3210
过程编程语言	3. 3145	过程指标	3. 3189
过程标准	3. 3206	过程质量	3. 3197
过程参考模型	3. 3204	过程质量等级	3. 3203
过程测量	3. 3187	过程质量确定	3. 3200
过程测量框架	3. 3188	过程质量属性	3. 3198
过程成熟度	3. 3186	过程质量特性	3. 3199
过程定义	3. 3170	过程质量维度	3. 3201
过程分析	3. 3151	过程质量指标	3. 3202
过程风险确定	3. 3205	过程资产库	3. 3157
过程改进	3. 3176	过程组	3. 3175
过程改进计划	3. 3179	过程组件	3. 3167
过程改进目标	3. 3178	过渡变量	3. 4813
过程改进支持要素	3. 3182	过渡场景	3. 4812
过程管理	3. 3185	过渡系统	3. 649
过程环境	3. 3168	过拟合	3. 2902
过程基础设施	3. 3183		
过程绩效基线	3. 3193	H	
过程架构	3. 3153	哈佛架构	3. 2015
过程架构师	3. 3152	合并	3. 2579
过程剪裁	3. 3207	合格评定	3. 934
过程结果	3. 3190	合格性测试	3. 3413
过程决策程序图	3. 3169	合理成本估算	3. 3978
过程可信度	3. 3208	合理可预见的误用	3. 3500
过程轮廓	3. 3195	合同背景	3. 998
过程描述	3. 3171	合同变更控制系统	3. 993
过程目标	3. 4539	合同工作分解结构	3. 996
过程目的	3. 3196	合同工作分解结构 (CWBS)	3. 997
过程内聚	3. 3143	合同管理	3. 992
过程能力	3. 3161	合同管理计划	3. 994
过程能力等级	3. 3164	合同条款和条件	3. 995
过程能力基线	3. 3162	合同需求	3. 999
过程能力评级	3. 3165	合同要求的审核	3. 1000
过程能力确定	3. 3163	核心	3. 1043
过程评估	3. 3155	核心报告	3. 1044
过程评估模型	3. 3156	核心价值	3. 1045
过程区域	3. 3154	核心外层 (外壳) 程序	3. 3977
过程实例	3. 3184	核心资产	3. 1478
过程视图	3. 3209	盒图	3. 524
		黑盒	3. 499

黑盒测试.....	3. 500,	3. 4204	回退.....		3. 409
恒定失效期.....		3. 946	汇编.....		3. 299
横向压缩.....		3. 2386	汇编并运行.....		3. 300
宏.....		3. 2487	汇编程序.....		3. 302
宏编程.....		3. 2495	汇编代码.....		3. 304
宏处理程序.....		3. 2494	汇编的起点.....		3. 301
宏定义.....		3. 2488	汇编语言.....	3. 305,	3. 306
宏汇编程序.....		3. 2491	会话编译器.....		3. 2121
宏库.....		3. 2490	会话式[的].....		3. 1036
宏生成程序.....		3. 2489	会话式编译程序.....		3. 1037
宏生成器.....		3. 2492	会话组件.....		3. 3967
宏指令.....		3. 2493	绘制流程图.....		3. 1887
后备程序员.....		3. 413	混合方法.....		3. 2056
后处理器.....		3. 3062	混合方式.....		3. 2636
后台.....		3. 405	混合耦合.....		3. 2057
后台处理.....		3. 406	混合条目表.....		3. 2635
后同步码断点.....	3. 1653,	3. 3055	混淆矩阵.....		3. 937
后向恢复.....		3. 417	豁免条款.....		3. 2081
后向执行.....		3. 415	活动.....		3. 78
后续活动.....		3. 4379	活动编码.....		3. 79
后置辅文.....		3. 404	活动标识.....		3. 84
后置条件.....		3. 3058	活动表.....		3. 85
候选功能规模测量方法.....		3. 592	活动成本估算.....		3. 80
候选键.....		3. 593	活动持续时间估算.....		3. 82
弧.....		3. 249	活动导向型 WBS		3. 88
弧形标注.....		3. 250	活动对象.....		3. 74
互操作能力、互操作性.....		3. 2298	活动类.....		3. 70
互操作系统.....		3. 2300	活动类型.....		3. 86
互操作性测试.....		3. 2299	活动企业对象.....		3. 71
互斥.....		3. 2715	活动冗余.....		3. 75
互斥聚类.....		3. 2717	活动图.....		3. 81
互斥子状态.....		3. 1438	活动网络图.....	3. 3355	
互动小组.....		3. 2263	活动文本.....		3. 76
互换参考点.....		3. 2269	活动文件.....		3. 72
互通参考点.....		3. 2316	活动组.....		3. 83
环境.....		3. 1648	活跃空白空间.....		3. 77
环境契约.....		3. 1649	活跃区域.....		3. 69
环境特定 ORB 间协议.....		3. 1650	获取.....		3. 47
环境与社会风险免除.....		3. 1925	获取策略.....		3. 55
缓冲区.....		3. 548	获取后勤.....		3. 49
缓存.....		3. 576	获取协定.....		3. 48
恢复点.....		3. 3682	获取资源.....		3. 44
回放调试.....		3. 3708	耦合.....		3. 1077
回复紧密耦合的消息通信.....		3. 4435		I	
回顾性.....		3. 3689	I/O 任务结构化标准		3. 2062
回归.....		3. 3541	ICOM 标签		3. 2065
回归测试.....		3. 3543	ICOM 代码, ...		3. 2064
回归分析.....		3. 3542	ICT 产品, ...		3. 2068
回送.....		3. 1553	ICT 需求		3. 2069
回溯日志.....		3. 3690	IP0 图		3. 2222

IT 服务	3. 2338	基础设施即代码	3. 2191
IT 服务体系	3. 2341	基础设施生态系统	3. 2192
IT 项目数据集	3. 2337	基础许可证	3. 4860
IT 资产管理系统	3. 2331	基地址	3. 420
IT 资产组合	3. 2334	基类	3. 422
IT 的使用	3. 4914	基线	3. 431
IT 服务功能	3. 2340	基线功能点计数	3. 436
IT 服务适应性	3. 2339	基线管理	3. 437
IT 基础设施	3. 2335	基线配置管理	3. 432
IT 基础设施管理	3. 2336	基线设计	3. 434
IT 资产管理	3. 2333	基线数据	3. 433
IT 资产管理计划	3. 2330	基线文件	3. 435
IT 资产管理体系	3. 2332	基于场景的评审	3. 3823
J		基于订阅的许可	3. 4370
<精益>现场	3. 1961	基于风险的测试	3. 3763
JSON web 令牌	3. 2356	基于服务的许可	3. 4370
机构知识	3. 2240	基于规格说明的测试	3. 4203, 3. 4204
机会	3. 2848	基于规则的语言	3. 3780
机会成本	3. 2849	基于计算机软件系统	3. 861
机会和威胁	3. 4321	基于价值的系统需求	3. 4986
机器翻译	3. 2485	基于检查表的评审	3. 659
机器可读	3. 2484	基于角色的评审	3. 3770
机器可读的	3. 2486	基于结构的测试	3. 4329, 3. 4334
机器可读护照	3. 2689	基于经验的测试	3. 1758
机器人	3. 3766	基于领域的需求	3. 1497
机器学习	3. 2483	基于模型的系统工程	3. 2660
机制回环	3. 2572	基于模型的系统与软件工程	3. 2659
积极结果	3. 1343	基于期限的许可	3. 4370
基本标准	3. 429	基于人工智能的系统	3. 295
基本测度	3. 427	基于软件的系统	3. 4164
基本成熟度等级	3. 444	基于视角的阅读	3. 2996
基本符号	3. 447	基于项目的组织	3. 3368
基本概况	3. 446	基于需求的测试	3. 3644
基本工程对象	3. 440	基于用户的评估	3. 4942
基本功能部件 (BFC) 功能服务	3. 424	基于自主性的改进	3. 398
基本功能部件类型 (BFC 类型)	3. 426	基准	3. 461, 3. 3518
基本构建块	3. 553	基准测试套件	3. 462
基本过程	3. 1570	基准度量	3. 463
基本过程集	3. 445	基准度量报告	3. 467
基本互通设施	3. 443	基准度量方法	3. 466
基本假设	3. 439	基准度量分析方	3. 464
基本流程	3. 441, 3. 442	基准度量经验库	3. 465
基本路径	3. 442	基准度量实例	3. 2235
基本实践	3. 428	基准度量数据仓库	3. 468
基本选择	3. 421	基准对比用户	3. 469
基本原则	3. 1996	基准结果应用活动	3. 4953
基本值	3. 430	基准真值	3. 1997
基础功能组件	3. 423	基组	3. 449
基础功能组件类型	3. 425	激活	3. 65
基础设施	3. 2190	激活函数	3. 67

激活约束	3. 66	计划值	3. 3024
激活值	3. 68	计划质量	3. 3014
激励	3. 4296	计划质量管理	3. 3014
及时吞吐量	3. 4739	计量、度量[体制]	3. 2606
及时性	3. 4736	计时	3. 4743
及时性评价值	3. 4738	计时分析程序	3. 4744
级别测试用例	3. 2481	计数范围	3. 1076
极简主义	3. 2625	计数规则	3. 1075
极限编程	3. 1804	计数目的	3. 3403
极小实体	3. 5055	计数器	3. 1074
即时控制	3. 498	计算对象模板	3. 824
集成	3. 2250	计算工厂	3. 822
集成测试	3. 2253	计算机	3. 826
集成储藏库	3. 2249	计算机安全	3. 849
集成电路	3. 2245, 3. 2246	计算机程序	3. 836
集成电路间总线	3. 2282	计算机程序部件	3. 840
集成计划	3. 2252	计算机程序开发计划	3. 843
集成开发环境	3. 2247	计算机程序配置标识	3. 841
集成开发环境 (IDE)	3. 2248	计算机程序配置项	3. 842
集合基数	3. 728	计算机程序确认	3. 844
集合类	3. 729	计算机程序认证	3. 839
集合属性	3. 733	计算机程序验证	3. 845
集合值	3. 730	计算机程序摘要	3. 837
集合值类	3. 731	计算机程序注解	3. 838
集合值属性	3. 732	计算机代次	3. 830
集群	3. 699	计算机犯罪	3. 828
集群管理器	3. 701	计算机分代	3. 830
集群检查点	3. 700	计算机辅助的	3. 856
集群模板	3. 702	计算机辅助工具	3. 860
计划	3. 3010	计算机辅助软件工程	3. 858
计划标准	3. 3022	计算机辅助软件工程需求	3. 859
计划采购	3. 3013	计算机辅助设计	3. 857
计划采购管理	3. 3013	计算机化	3. 862
计划成本管理	3. 3012	计算机科学	3. 848
计划范围管理	3. 3019	计算机软件部件	3. 851
计划风险管理	3. 3016	计算机软件配置项	3. 852, 3. 853
计划风险应对	3. 3017	计算机软件组件	3. 850
计划工作的预算成本	3. 3025	计算机数据	3. 829
计划沟通	3. 3011	计算机图形学	3. 831
计划沟通管理	3. 3011	计算机网络	3. 834
计划过程	3. 3023	计算机系统	3. 854
计划绩效域	3. 3028	计算机性能评估	3. 835
计划价值	3. 3025	计算机语言	3. 833
计划进度管理	3. 3018	计算机指令	3. 832
计划开始日期 (SS)	3. 3838	计算机中心	3. 827
计划评估审查技术	3. 3281	计算机资源	3. 846
计划评审技术	3. 3280	计算机资源分配	3. 847
计划扑克	3. 3029	计算机字	3. 855
计划启动	3. 3326	计算接口模板	3. 823
计划完成日期 (SF)	3. 3837	计算容器	3. 821

计算设备	3. 864	价值等级	3. 4970
计算视点	3. 825	价值工程	3. 4976
计算数据使用	3. 865	价值基准	3. 4965
计算系统	3. 854	价值集群	3. 4971
计算性数据使用	3. 819	价值交付办公室	3. 4972
计算中心	3. 863	价值交付系统	3. 4973
计算组件	3. 820	价值链分析	3. 4968
记录管理系统	3. 3508	价值领导者	3. 4978
记录类型	3. 3507	价值流图	3. 4983
记录元素类型 (RET)	3. 3506	价值流图法	3. 4984
记录元素类型 (RET)	3. 3505, 3. 3506	价值伤害	3. 4977
技能	3. 4010	价值收益	3. 4967
技术标准	3. 4573	价值展示者	3. 4974
技术产品质量需求	3. 4570	价值主张	3. 4980
技术成熟度提升与风险降低	3. 4749	架构	3. 258
技术独立	3. 4567	架构定义	3. 260
技术复杂性调整	3. 4564	架构定义策略	3. 262
技术复杂性调整因素	3. 4565	架构定义记录	3. 261
技术管理	3. 4568	架构风格	3. 257
技术路线图	3. 4574	架构化	3. 252
技术审查	3. 4572	架构集合	3. 259
技术视图	3. 4575	架构结构	3. 256
技术性能测量	3. 4569	架构可追溯性	3. 271
技术需求	3. 4571	架构框架	3. 270
技术债务	3. 4566	架构描述	3. 266
继承	3. 2090	架构描述框架	3. 264
继承属性	3. 2193	架构描述语言	3. 265
寄存器	3. 3537	架构描述元素	3. 263
寄存器组	3. 3538	架构评估	3. 268
绩效报告	3. 2974	架构评估框架	3. 269
绩效评估	3. 2976	架构驱动现代化	3. 274
加法加权法	3. 109	架构设计	3. 253
加密哈希	3. 1111	架构设计阶段	3. 254
加权里程碑法	3. 5103	架构设计评审	3. 255
加载	3. 2434	架构师	3. 251
加载程序	3. 2439	架构实体	3. 267
加载即执行	3. 2437	架构视点	3. 273
加载模块	3. 2436	架构视图	3. 272
加载起点	3. 2438	架构性构造	3. 4705
加载映射表	3. 2435	间隔标度	3. 2313
假设	3. 346	间接开销	3. 2141
假设分析	3. 349	间接量测	3. 2143
假设和约束分析	3. 347	间接人工	3. 2142
假设情景分析	3. 5108	间接用户	3. 2144
假设日志	3. 348	间歇故障	3. 2287
假阳性	3. 1827	监督	3. 4415
假阴性	3. 1826	监督程序	3. 1020, 3. 1742, 3. 4388, 3. 4391
价值变化覆盖率	3. 4969	监督风险	3. 2681
价值承载者	3. 4966	监督沟通	3. 2680
价值处置	3. 4975	监督器	3. 1742

监督相关方参与	3. 2682	降级权	3. 1506
监督学习	3. 4387	交叉编译程序	3. 1107
监督状态	3. 1741, 3. 4389, 3. 4390	交叉汇编程序	3. 1106
监管背景	3. 3546	交叉引用	3. 1108
监控	3. 2684	交叉引用工具	3. 1110
监控程序, 监控器	3. 2677	交叉引用清单	3. 1109
监控风险	3. 1024, 3. 2679	交叉引用生成器	3. 1108
监控过程组	3. 2685	交错	3. 2284
监控项目工作	3. 2678	交付	3. 1276
监听	3. 4017	交付过程	3. 597
兼容的评估模型	3. 775	交付绩效域	3. 1277
兼容性	3. 773	交付物	3. 1273
兼容性测试	3. 774	交付源码声明	3. 1275
检测误差权衡	3. 1355	交互能力	3. 2262
检查, 检出	3. 660	交互式	3. 2266
检查表	3. 662	交互式编译器	3. 2121
检查表分析	3. 658	交互系统	3. 2268
检查点	3. 661	交互序列	3. 2265
检查和审计	3. 2228	交互语言	3. 2267
检查评估	3. 2227	交换	3. 4419
减损	3. 2085	交钥匙的	3. 4839
剪裁指南	3. 4529	交钥匙系统	3. 4840
简并	3. 1281	胶囊	3. 602
简单缓冲	3. 3988	焦点类	3. 1888
简单性	3. 3990	角色	3. 2990, 3. 3768
简单知识组织系统	3. 3989	角色扮演会议	3. 3771
简洁性	3. 876	角色名称	3. 3769
建立工作关系网	3. 2740	脚本测试	3. 3852
建立规则的运作	3. 3092	脚手架	3. 3810
建立和维护	3. 1681	教育部的初步需求	3. 3084
建立行为	3. 1683	阶段	3. 2999, 3. 4225
建模	3. 2661	接口储藏库	3. 2274
建模工具	3. 2662	接口规格说明	3. 2279
建议	3. 3390	接口控制	3. 2271
健壮性, 稳健性	3. 3767	接口任务	3. 2280
渐进明细	3. 3311	接口设计文档	3. 2070, 3. 2273
鉴定(系统)	3. 3407	接口特征标记	3. 2278
鉴定限制	3. 3409	接口需求	3. 2275
鉴定需求	3. 3411	接口需求规格说明	3. 2276
鉴定裕度	3. 3410	接续的	3. 940
键迁移	3. 2365	节点	3. 3009
键式视图	3. 2369	节点号	3. 2746
箭头	3. 282	节点字母	3. 2745
箭头标签	3. 283	节拍	3. 4723
箭头参考	3. 2064	节奏	3. 577
箭头段	3. 286	结对编程	3. 2927
箭头含义	3. 284	结对评审	3. 2928
箭头角色	3. 285	结构	3. 1920
奖励黑客行为	3. 3715	结构测试	3. 4327, 3. 4328, 3. 4329
降低风险	3. 3747	结构管理	3. 913

结构化编写	3. 4335	进度压缩	3. 3826
结构化查询语言	3. 4214	进度预测	3. 3829
结构化程序	3. 4337	进度制定	3. 3828
结构化程序设计	3. 4338	进入动作	3. 1642
结构化程序设计语言	3. 4339	进入准则	3. 1643
结构化设计	3. 4336	进行定量分析	3. 2968
结构化写作	3. 4341	进行定量风险分析	3. 2969
结构化走查	3. 4340	近期工作绩效报告	3. 5146
结构模型	3. 4333	禁闭	3. 927
结构视图	3. 612	禁令	3. 3312
结构图	3. 4330, 3. 4331	经过确认的交付件	3. 4956
结构相搅	3. 4332	经验教训	3. 2397
结构性模型方面	3. 4326	经验教训登记册	3. 2398, 3. 2399
结构性特征	3. 4325	经验教训知识库	3. 2399
结果	3. 2891, 3. 3683	精化	3. 3532
结算	3. 3968	警告	3. 618, 3. 5094
截止日期	3. 1121	警告信息	3. 5095
解绑	3. 4853	静寂	3. 3465
解绑行为	3. 4851	静态绑定	3. 4271
解除绑定行为	3. 4852	静态测试	3. 4282
解除紧缩	3. 4883	静态差错	3. 4275
解决方法	3. 5130	静态产品	3. 4277
解块	3. 1229	静态程序分析	3. 4278
解耦	3. 1251	静态的	3. 4267
解释	3. 2301	静态断点	3. 4272
解释程序（解释器）	3. 2303	静态分类	3. 4273
解释代码	3. 2304	静态分析	3. 4268
解释性报告	3. 1762	静态分析程序	3. 4269
解析	3. 2941	静态模式	3. 4281
解析器	3. 2942	静态模型	3. 4276
解域	3. 4168	静态属性	3. 4270
界面，接口	3. 2270	静态随机存取存储器	3. 4279, 3. 4280
界面角色	3. 2277	静态转储	3. 4274
界面控制文件	3. 2272	镜像站点	3. 2632
仅供说明页	3. 1891	纠正	3. 1049
仅作说明页	3. 1890	纠正措施	3. 1050
紧急情况	3. 1576	局部变量	3. 2446
紧急维修	3. 1577	局部符合逻辑	3. 2441
紧耦合消息通信	3. 4438	局部数据	3. 2444
紧前关系绘图法	3. 3068	局部微代码压缩	3. 2442
紧缩	3. 771	局域网	3. 2440
进度管理计划	3. 3830	矩阵图	3. 2534
进度基线	3. 3825	句法测试	3. 4444
进度绩效度量	3. 3836	句法差错	3. 4442
进度绩效指标	3. 3834	具名常量	3. 2722
进度绩效指数	3. 3833	具名常数	3. 2722
进度模型	3. 3831	具名约束	3. 2723
进度偏差	3. 3835	具体类	3. 877
进度数据	3. 3827	具有不同抽象级别的可追溯性	3. 4781
进度网络分析	3. 3832	聚合	3. 139

聚合资源利用率	3. 138	开放系统	3. 2816
聚集	3. 135	开放系统互连	3. 2818
聚集[类]型	3. 136	开放系统环境	3. 2817
聚集方法	3. 140	开机时间	3. 4891
聚类	3. 703	开始到完成	3. 3972
决策	3. 1498	开始-结束块	3. 454
决策策略	3. 1235	开箱即用	3. 2801
决算	3. 1863	开销操作	3. 2904
绝对代码 特定代码	3. 9	开销时间	3. 2905
绝对地址 显式地址 特定地址	3. 7	开源软件	3. 2815
绝对汇编程序	3. 8	看板	3. 2358
绝对机器代码	3. 12	看门狗定时器	3. 5097
绝对时间	3. 13	抗风险	3. 1928
绝对指令	3. 10	抗攻击性	3. 3656
绝对装入程序	3. 11	考核	3. 1726
均值	3. 2541	可编程断点	3. 3295
K		可编程计数器阵列	3. 3296
[控制的]线程	3. 4711	可编程逻辑器件	3. 3297
[捆]绑	3. 486	可编程逻辑器件软件	3. 3298
[捆]绑者	3. 487	可编程脉冲发生器	3. 3299
KDM 关系	3. 2362	可编程芯片系统	3. 4490
KDM 模型	3. 2361	可编程只读存储器	3. 3300
KDM 实例	3. 2360	可编程终端	3. 3302
KDM 实体	3. 2359	可编程终端智能终端	3. 3303
开发	3. 1362	可编程重载定时器	3. 3301
开发测试	3. 1376	可编微程序的计算机	3. 2613
开发方法	3. 1363	可变成本	3. 5011
开发方法学	3. 1367	可变的	3. 5009
开发方法与生存周期绩效域	3. 1364	可变机制操作化 (VMO)	3. 5002
开发分支	3. 1365	可变技术状态项	3. 5010
开发工具	3. 1377	可变类	3. 2714
开发过程	3. 1370	可变性	3. 4988, 3. 5019
开发后评审	3. 4474	可变性的绑定时间	3. 495
开发后审查	3. 3059, 3. 4473	可变性管理	3. 4999
开发环境	3. 1366	可变性机制	3. 5000
开发基线	3. 1378	可变性机制可操作化	3. 5001
开发计划	3. 1369	可变性建模策略	3. 5005, 3. 5007
开发流水线	3. 1368	可变性建模计划	3. 5004, 3. 5006, 3. 5023
开发配置	3. 1379	可变性结合	3. 4989
开发配置管理	3. 1380	可变性可追溯性	3. 5008
开发团队	3. 1375	可变性模型	3. 5003
开发项目	3. 1371	可变性实施	3. 4993
开发项目功能点计数	3. 1373	可变性文件	3. 4992
开发项目功能点计数 (DFP)	3. 1372	可变性依赖	3. 4991
开发项目功能规模	3. 1374	可变性约束	3. 4990
开发源语句	3. 1361	可变性指数	3. 4998
开发运维	3. 1383	可擦除可编程只读存储器	3. 1666
开放分布式处理	3. 2813	可擦可编程只读存储器	3. 1667
开放认证	3. 2784	可测试性	3. 4689
开放软件基金会	3. 2814	可撤销性	3. 3706

可达标识	3. 3478	可信赖性要求	3. 1285
可达集合	3. 3477	可信性测量	3. 1284
可达性	3. 35	可信原则	3. 4830
可达性图	3. 3476	可行性研究	3. 1842
可定制性	3. 1118	可修改的	3. 2664
可读性	3. 3484	可修改性	3. 2663
可分析性	3. 172	可选非标标识关系	3. 2857
可复用软件产品	3. 3699	可选任务	3. 2859
可改进性	3. 2100	可选属性	3. 2856
可干预性	3. 2314	可选要求	3. 2858
可更新参数	3. 4892	可选择的	3. 2855
可恢复性	3. 3509	可验收标准	3. 21
可计数故障	3. 1073	可验收的	3. 22
可监控性	3. 2683	可验证的	3. 5031
可检索查看	3. 3093	可验证的需求语言	3. 1905
可见性	3. 5080	可验证需求语言	3. 1904
可交付产品	3. 1274	可移植性	3. 3048
可接受风险	3. 4754	可用交互	3. 402
可接受性	3. 20	可用性	3. 400
可解释性	3. 1761, 3. 2302	可用性测试	3. 4903
可纠正性	3. 1048	可用性发现	3. 4898
可靠性	3. 3567	可用性检查	3. 4899
可靠性测试	3. 3574	可用性模型	3. 401
可靠性等级	3. 3569	可用性目标	3. 4901
可靠性模型	3. 3572	可用性缺陷	3. 4897
可靠性评估	3. 3568	可用性实验室	3. 4900
可靠性数据	3. 3570	可用性问题	3. 4902
可靠性增长	3. 3571	可用性演练	3. 4906
可靠性值	3. 3573	可运输性	3. 3048
可扩充性	3. 1753, 3. 1772	可支持性	3. 4413
可伸缩性	3. 3818	可执行标准	3. 2088
可扩展标记语言	3. 1777	可执行需求规范	3. 1729
可扩展性	3. 1772, 3. 3811	可执行语句	3. 1731
可扩展样式表语言转换	3. 1778	可执行源语句	3. 1730
可扩展样式语言转换	3. 1779	可重定位代码	3. 3577
可理解性	3. 4861	可重定位地址	3. 3576
可利用的	3. 1766	可重定位的	3. 3575
可能性	3. 2423	可重定位机器[代]码	3. 3578
可逆执行	3. 3707	可重复使用的产品	3. 3698
可配置性	3. 901	可重复性	3. 3593
可迁移性	3. 2617	可重入的	3. 3515
可审核性	3. 375	可重用的, 可复用的	3. 3697
可使用性测试	3. 4904	可重用性, 可复用性	3. 3696
可视化数据与信息	3. 5082	可转换性	3. 1040
可替代性	3. 4374	可追溯	3. 4782
可替换性	3. 3596	可追踪性	3. 4779
可维护性	3. 2499	克隆	3. 688
可问责性	3. 39	客户	3. 1117
可信度	3. 4829	客户端	3. 685
可信过程域	3. 4832	客户端对象	3. 686

客户满意度	3. 1116	快速原型法	3. 3469
课程标准	3. 1113	快照转储	3. 4016
空操作	3. 1500	宽带德尔菲法	3. 5113
空格	3. 5111	框架	3. 1922
空间变化	3. 5020	框图	3. 503
空间的可变性	3. 4995	扩展	3. 1771
空间极小实体	3. 4183	扩展过程集	3. 1776
空间位置	3. 2450	扩展设置	3. 1780
控制	3. 1002	扩展条目表	3. 1775
控制变量	3. 1032	扩展性测试	3. 3812
控制成本	3. 1008	扩展元素	3. 1774
控制程序	3. 1020, 3. 1742, 3. 4391	扩展组件	3. 1773
控制存储器	3. 1029		
控制断点	3. 1004	L	
控制范围	3. 1026	[列]表, 清单	3. 2430
控制风险	3. 1023, 3. 1024	莱文斯坦距离	3. 2408
控制沟通	3. 1007	浪费	3. 5096
控制管理器	3. 1019	老化周期	3. 560, 3. 1547
控制环回	3. 1018	乐观工期	3. 2852
控制结构	3. 1030	类[别]	3. 672
控制进度	3. 1025	类比估算	3. 169
控制聚焦	3. 1889	类别簇	3. 4348
控制聚类	3. 1006	类别判别器	3. 1437
控制流	3. 1012, 3. 1884	类别实体	3. 614
控制流程图	3. 1013	类层次结构	3. 674
控制流子路径	3. 1014	类对象二元体	3. 670
控制流踪迹	3. 1015, 3. 1740	类级方法	3. 681
控制耦合	3. 1009	类级属性	3. 680
控制器	3. 1033	类级职责	3. 682
控制器区域网络	3. 1034	类数据	3. 2675
控制任务	3. 1031	类图	3. 673
控制数据	3. 1010	类推法	3. 171
控制图	3. 1005	类型	3. 4848
控制相关方参与	3. 1027	类型表达式	3. 4849
控制信息	3. 1016	累计流程图	3. 1112
控制与观察点	3. 3040	离散数据	3. 1431
控制语句	3. 1028	离散型	3. 1433
控制语言	3. 1017	礼貌	3. 1078
控制账户	3. 1003	里程碑回顾	3. 2622
控制资源	3. 1022	里程碑进度计划	3. 2623
控制字段	3. 1011	里程碑清单	3. 2621
库	3. 2411	理想工时	3. 2073
库存量单位	3. 4297	历史文件	3. 2041
库所初始标记	3. 2200	历史信息	3. 2040
跨度	3. 4184	利害关系属性	3. 3377
块【名】	3. 501	利益相关方	3. 4226
块分配	3. 502, 3. 2926	利益相关方绩效域	3. 4234
块结构语言	3. 506	利益相关方满意度	3. 4239
快乐的一天场景	3. 442	利益相关方权益	3. 4229
快速跟进	3. 1830	利益相关方视角	3. 4235
		利益相关方需求规格说明	3. 4237

利益相关方需求可追溯性	3. 4238	领域层	3. 1483
利益相关方需要	3. 4230	领域超集	3. 1492
利益相关者（方）需求和要求定义策略	3. 4233	领域范围界定	3. 1491
利益相关者（方）需求和要求定义记录	3. 4232	领域分析	3. 1474
利益相关者（方）要求	3. 4236	领域工程	3. 1480
利益相关者参与计划	3. 4228	领域工程过程	3. 1481
利益相关者参与评估矩阵	3. 4227	领域架构	3. 1475
利益相关者的需求	3. 4231	领域架构、产品线架构、参考架构	3. 1476
利益行动	3. 59	领域接口	3. 1482
利用界限定理	3. 4952	领域可变性模型	3. 1496
利用率	3. 4951	领域模型	3. 1484
利用提前量与滞后量	3. 238	领域实现	3. 1485
例程	3. 3778	领域需求分析	3. 1486
例会	3. 4288	领域需求管理	3. 1488
粒度	3. 1990	领域需求规格说明	3. 1489
连接	3. 938	领域需求获取	3. 1487
连接，链接	3. 2428	领域需求验证与确认	3. 1490
连接加载程序	3. 2429	领域专家	3. 4358
连通性	3. 939	领域资产	3. 1478
连续表示	3. 990	领域组件	3. 1479
连续分配	3. 977	令牌	3. 4753
连续性	3. 984	浏览器	3. 540
联机帮助	3. 2809	流	3. 1883
联络	3. 2410	流程改进措施	3. 3177
联系，关系	3. 3550	流程改进计划	3. 3180
联想类	3. 342	流程改进项目	3. 3181
链路	3. 562	流程类别	3. 3166
链式	3. 635	流程图	3. 1885
两地址指令	3. 4842	流程图生成工具	3. 1886
两级编码	3. 4845	流接口	3. 4319
两级地址	3. 4844	流接口签名	3. 4320
两阶段获得	3. 4846	流水线	3. 3008
谅解函	3. 2400	龙卷风图	3. 4766
量化属性	3. 3376	漏洞	3. 5090
量子计算	3. 3462	路径	3. 2951
列表	3. 2432	路径表达式	3. 2954
列表功能	3. 2431	路径分析	3. 2952
劣势	3. 4321	路径条件	3. 2953
邻接分配	3. 977	路线图	3. 3765
临界段	3. 1100	伦理风险	3. 1693
临界范围	3. 1099	伦理价值	3. 1694
临界区	3. 1100	伦理价值需求	3. 1695
临界值	3. 1103	伦理需求	3. 1692
临时审查	3. 95	伦理原则	3. 1691
零地址指令	3. 5147	伦理政策声明	3. 1690
零投资备选方案	3. 1460	逻辑	3. 2459
领域、取值范围	3. 1473	逻辑编程语言	3. 2460
领域测试	3. 1495	逻辑布局	3. 2464
领域测试需求	3. 1494	逻辑关系	3. 2466
领域测试资产	3. 1493	逻辑轨迹	3. 2468

逻辑记录	3. 2465	命名图	3. 2727
逻辑接口	3. 2463	命名域	3. 2726
逻辑内聚	3. 2461	命题/主张	3. 3381
逻辑事务	3. 2469	模板	3. 4577
逻辑文件	3. 2462	模板 <X> 模板	3. 4578
逻辑型	3. 2470	模板类	3. 4579
逻辑源语句	3. 2467	模板类型	3. 4581
螺旋模型	3. 4206	模板评级模块	3. 4580
绿地系统工程	3. 1995	模糊性	3. 167
M			
买方	3. 1117	模块	3. 2673
卖方	3. 3898, 3. 4399	模块分解	3. 2669
卖家建议	3. 3899	模块化	3. 2672
脉冲宽度调制	3. 3402	模块化编程	3. 2670
满意决策	3. 3809	模块强度	3. 2676
枚举型	3. 1647	模块数据	3. 2674, 3. 2675
每日站会	3. 1128	模块性	3. 2671
每日站立	3. 1127	模拟	3. 168
蒙特卡洛模拟	3. 2687	模拟对象	3. 2639
面	3. 1806	模拟计算机	3. 170
面包屑路径	3. 536	模拟器	3. 1584
面向成本的设计	3. 1336	模式	3. 2641
面向对象的设计	3. 2798	模式规则语言	3. 3841
面向对象的系统工程方法	3. 2812	模式敏感故障	3. 1220, 3. 2956
面向过程的语言	3. 3144	模型	3. 2642
面向活动的 WBS	3. 88	模型层	3. 2651
面向数据结构的设计	3. 1221	模型层级	3. 2649
面向问题的语言	3. 3142	模型存储库	3. 2658
面向应用的语言	3. 235	模型基线	3. 2643
描述	3. 1305	模型名称缩写	3. 2653
描述标准	3. 1306	模型模式	3. 2657
描述模型	3. 1307	模型配置项	3. 2644
敏感性分析	3. 3911	模型驱动体系结构	3. 2645
敏捷	3. 141	模型术语表	3. 2648
敏捷成熟度	3. 145	模型页面	3. 2656
敏捷概念	3. 142	模型元素	3. 2646
敏捷环境	3. 144	模型元素库	3. 2647
敏捷开发	3. 143	模型种类	3. 2650
敏捷开发框架	3. 3854	模型注释	3. 2654
敏捷团队	3. 146	模型注释编号	3. 2655
敏捷团队领导	3. 147	磨损失效周期	3. 5100
名称解析	3. 2720	末期功能点分析	3. 1864
名义尺度	3. 2747	目标	3. 1987
明文文件编码	3. 684	目标, 对象	3. 2785
命令	3. 738	目标陈述	3. 4263
命令驱动	3. 740	目标程序	3. 2792
命令语言	3. 739	目标单元	3. 4357
命名操作	3. 2724	目标风险	3. 4542
命名空间	3. 2721	目标工具	3. 4356
命名上下文	3. 2725	目标过程概要	3. 4541
		目标函数	3. 2797

目标机	3. 4537	内在差错	3. 2140
目标计算机	3. 4533	内在的、固有的	3. 2317
目标节点	3. 4538	内在关系	3. 2318
目标能力	3. 4532	内置随机存取存储器	3. 554
目标软件	3. 4355	内置随机存取存储器 (RAM)	3. 555
目标实体	3. 4534	内置只读存储器	3. 556
目标受众	3. 4531	能动性	3. 133
目标系统	3. 4513	能力成熟度模型	3. 595
目标优化分析	3. 537	能力维	3. 594
目标语言	3. 2790, 3. 4535	能力需求	3. 596
目标语言 目标语言	3. 4536	逆变	3. 1001
目的地址	3. 1346	逆向调试	3. 3708
目录系统代理	3. 1423	逆向工程	3. 3704
目录信息树	3. 1456	逆向执行	3. 3705
目录用户代理	3. 1424	年当量	3. 188
N		年等值	3. 187
N 2 图	3. 2719	年金	3. 190
Nassi-Shneiderman 图	3. 525	年利率	3. 189
NVRAM	3. 1879	O	
N 平方图	3. 2778	偶发事件	3. 2106
n 元关系	3. 2729	偶然性因素	3. 979
内部动作	3. 2288	P	
内部度量 (软件质量)	3. 2293	PESTEL 分析	3. 3045
内部服务提供方	3. 2294	Petri 网	3. 2998
内部基数	3. 2209	PNML 对称网文档	3. 3038
内部集成电路总线 (I2C)	3. 2283	PNML 高层网文档	3. 3037
内部箭头	3. 2289	PNML 高级网络文档	3. 3037
内部可变性	3. 2296	排斥的	3. 2716
内部逻辑文件	3. 2079, 3. 2292	排列图	3. 585, 3. 4724
内部任务结构标准	3. 2295	排序生成器	3. 4170
内部属性	3. 2290	派生参与者项	3. 1301
内部性质的质量测度	3. 3448	派生类	3. 1296
内部依赖关系	3. 2291	派生属性	3. 1300, 3. 1301
内存	3. 2574	派生数据	3. 1297
内存容量	3. 2575	派生性质	3. 1301
内存映像	3. 2576	派生需求	3. 1303
内核	3. 2363	判别器	3. 1437
内核常驻控制程序	3. 2364	判定	3. 1231
内聚度 (内聚性)	3. 721	判定变量	3. 1242
内聚模块强度	3. 722	判定表	3. 1237
内容	3. 960	判定表测试	3. 1238
内容单元	3. 969	判定测试	3. 1239
内容对象	3. 967	判定规则	3. 1236
内容管理	3. 964	判定技术	3. 1243
内容管理平台	3. 965	判定结果	3. 1234
内容管理系统	3. 966	判定门	3. 1233
内容块	3. 961	判定树	3. 1240
内容类型	3. 968	判定树分析	3. 1241
内容耦合	3. 963	判定准则	3. 1232
内容一致性	3. 962	培训计划	3. 4791

配置	3. 902	平均修复时间	3. 2549
配置标识	3. 908	平均修复时间 (MTTR)	3. 2550
配置参数	3. 98, 3. 924	平均支持间隔时间	3. 2692
配置单元	3. 926	平均执行时间	3. 2542
配置管理 (CM)	3. 914	平均执行时间参考值	3. 2544
配置管理报告	3. 921	平均执行时间评级值	3. 2543
配置管理策略	3. 922	平台	3. 3034
配置管理服务	3. 704	平台即服务	3. 3035
配置管理工具	3. 705	平台即服务 (PaaS)	3. 3036
配置管理计划 (CMP)	3. 919	平台族	3. 1829
配置管理记录	3. 920	评标基准价	3. 438
配置管理库系统	3. 918	评定	3. 309, 3. 3471
配置管理权限	3. 915	评定的能力	3. 308
配置管理数据库	3. 916	评定工具	3. 314
配置管理数据库 (CMDB)	3. 917	评定级别	3. 3473
配置管理系统	3. 923	评估 (评价) 报告	3. 1706
配置基线	3. 904	评估 (评价) 程序	3. 1704
配置控制	3. 905	评估 (评价) 发起方	3. 1708
配置控制委员会	3. 906	评估 (评价) 检查表	3. 1698
配置控制委员会 (CCB)	3. 907	评估参与者	3. 241, 3. 316
配置审计	3. 903	评估发起者	3. 320
配置数据	3. 97	评估范围	3. 319
配置索引	3. 909	评估机构	3. 310
配置文件组	3. 3273	评估记录	3. 318
配置项	3. 910	评估结果	3. 240
配置项 (CI)	3. 912	评估目标	3. 4752
配置项开发记录	3. 911	评估目的	3. 317
配置状态记录 (CSA)	3. 925	评估输出	3. 315
喷泉模型	3. 1915	评估输入	3. 313
批处理的	3. 450	评估小组	3. 321
批准	3. 244, 3. 384	评估小组组长	3. 242
批准的变更请求	3. 246	评估员	3. 322
批准的变更请求审查	3. 247	评估约束	3. 311
批准的修改	3. 248	评估指标	3. 312
批准机构	3. 245	评级区间	3. 3472
偏差	3. 475	评价	3. 1696
偏差分析	3. 5014	评价方	3. 1712
偏离	3. 1381	评价方法	3. 1702
偏移量	3. 2800	评价覆盖率	3. 1699
片上系统	3. 4489	评价工具	3. 1711
片上振荡器	3. 2804	评价活动	3. 1697
漂移	3. 1509	评价级别	3. 1701
品质因数	3. 3431	评价记录	3. 1705
平均故障间隔时间	3. 2547	评价技术	3. 1710
平均检测时间	3. 2548	评价模块	3. 1703
平均软件故障间隔时间	3. 2691	评价请求方	3. 1707
平均失效时间间隔	3. 2545	评价严格度	3. 1709
平均失效时间间隔 (MTBF)	3. 2546	评价组	3. 1700
平均时间类指标 (统称)	3. 2695	评审	3. 3709
平均无故障时间	3. 2694	评审/审核输出	3. 3711

评审集	3. 3710	强制性非标识关系	3. 2517
评审输出/审计输出	3. 3713	强制性依赖	3. 2516
屏幕截图	3. 3851	强制转换	3. 718
屏幕图像	3. 3850	切换	3. 1122
屏幕文档	3. 2810	切换场景	3. 4812
屏幕转储	3. 3851	切片	3. 4011
破坏性读	3. 1347	亲和图	3. 131
剖面	3. 3272	轻量级进程	3. 2422
瀑布模型	3. 5098	清除	3. 683
Q		情感	3. 3912
（缺陷的）共同原因	3. 752	情感关系	3. 128
期末惯例	3. 1595	情感计算	3. 127
期望结果	3. 1342	情感权利	3. 129
期望值	3. 1755	情境因素	3. 975
期限式许可证	3. 4371	情况语句	3. 608
其他质量规划工具	3. 106	情绪图	3. 2688
企业	3. 1625	请求	3. 3612
企业观点	3. 1627	请求的更改	3. 3617
企业环境因素	3. 1626	穷尽测试	3. 1744
启动触发器	3. 2207	球栅阵列	3. 419
启动对象	3. 2204	驱动程序	3. 1510
启动过程	3. 2206	趋势分析	3. 4819
启动过程组	3. 2205	取消特化	3. 4885
启发式评估	3. 2024	取值范围约束	3. 4981
启用行为	3. 1585	去标识化	3. 1269
起始地址	3. 4250	权衡	3. 4785
气泡图	3. 541	权衡分析	3. 4786
千次运算每秒	3. 2377	权衡研究	3. 4784
迁移	3. 2619	权利	3. 3719
迁移场景	3. 4812	权利分配	3. 159
迁移键	3. 1894	权利转移	3. 4804
迁移属性	3. 2618	权限	3. 2982
迁移透明度	3. 2620	全簇	3. 4770
前导判定	3. 2392	全局标签	3. 1985
前台	3. 1892	全局属性	3. 1982
前台处理	3. 1893	全局数据	3. 1984
前同步码断点	3. 3371	全局唯一标识符	3. 1986
前向（正向）恢复	3. 1914	全局压缩	3. 1983
前置时间	3. 2390	全面质量管理	3. 4772, 3. 4774
前置时间图	3. 2391	全双工	3. 1931
前置条件	3. 3071	缺失变体	3. 6
潜在变量	3. 2385	缺陷	3. 549, 3. 1254
欠拟合	3. 4858	缺陷播种	3. 1837
嵌入式帮助系统	3. 1573	缺陷的特殊原因	3. 4188
嵌入式计算机系统	3. 1572	缺陷根源	3. 1258
嵌入式用户信息	3. 1574	缺陷密度	3. 1255
嵌套	3. 2737	缺陷去除	3. 3591
强化学习	3. 3547	缺陷修复	3. 1257
强类型法	3. 4324	缺陷预防	3. 1256
强制的	3. 2515	缺陷植入	3. 550

确保	3. 1623	容错	3. 1679
确定性系统	3. 1357	容积测试	3. 5089
确认	3. 4958	容量	3. 598
确认测试	3. 3686, 3. 4963	容量测试	3. 600
确认的系统之系统	3. 43	容量受限位置	3. 599
确认动作	3. 4959	容器	3. 603, 3. 957, 3. 3503
确认工作	3. 5043	容器接口	3. 958
确认偏误	3. 928	冗余	3. 3513
确认数据	3. 4960	入口	3. 1646
R		入口[点]	3. 1646
（软件）运行环境	3. 2820	入门资料	3. 1645
（软件产品或系统的）质量特性	3. 3426	软差错	3. 4021
“若-则-否则”语句	3. 2078	软件	3. 4167
燃尽率	3. 559	软件安全计划	3. 4145
燃尽图	3. 557, 3. 558	软件安全性	3. 4144
燃起图	3. 561	软件版本标识	3. 4160, 3. 4161
人工量	3. 1562	软件包	3. 4097
人工语言	3. 296, 3. 1901	软件被许可人	3. 4086
人工智能	3. 291	软件部件	3. 4039
人工智能系统	3. 294	软件测试	3. 4155
人工智能效应	3. 292	软件测试待查事件	3. 4154
人工智能质量元模型	3. 293	软件测试环境	3. 4153
人机接口（人机界面）	3. 2042	软件测试计划	3. 4309
人力资源规划	3. 2049	软件产品	3. 4109
人因	3. 2048	软件产品评估	3. 4110
人因系统工程	3. 2050	软件产品质量度量参考模型	3. 4209
人因系统集成	3. 2051	软件持续保障	3. 4150
认证（人员）	3. 3414	软件储藏库	3. 4134
认知逻辑	3. 1654	软件创建者	3. 4047
认知走查	3. 720	软件打包器	3. 4098
任期评价	3. 4585	软件代理	3. 4025
任务	3. 2633, 3. 4543	软件单元	3. 4157
任务板	3. 4555	软件盗版	3. 4099
任务倒置	3. 4548	软件的非功能性需求	3. 2758
任务构建	3. 4552	软件多样性	3. 4068
任务构建准则	3. 4557	软件发布管理	3. 4130
任务接口	3. 4547	软件发布管理计划信息	3. 4166
任务聚类准则	3. 4556	软件非功能评估过程（SNAP）点	3. 4091
任务看板	3. 4545	软件非功能评估过程（SNAP）子类别	3. 4092
任务类型	3. 4554	软件非功能评估过程点	3. 4095
任务模式	3. 4550	软件非功能评估过程计数单元	3. 4094
任务启动会	3. 4549	软件非功能评估过程类别	3. 4093
任务提交	3. 4553	软件非功能评估过程子类	3. 4096
任务完成	3. 4546	软件非功能性评估过程（SNAP）类别	3. 4090
任务行为规格说明	3. 4544	软件风险	3. 4078
任务优先级准则	3. 4551	软件服务	3. 4147
韧性	3. 3654	软件更正	3. 4046
日历单位	3. 578	软件工程	3. 3971
日志	3. 2456	软件工程过程组	3. 4072
容差	3. 4755	软件工程环境	3. 3886

软件工程环境服务	3. 3884	软件配置项	3. 852, 3. 4042
软件工程经济学	3. 4070	软件设计	3. 4049
软件工程开发方法元模型	3. 3908	软件设计表示法	3. 4054
软件工程组	3. 4071	软件设计符号	3. 4053, 3. 4054
软件工件	3. 4028	软件设计理念	3. 4051
软件工具	3. 4156	软件设计描述	3. 4052
软件工作产品	3. 4162	软件设计审计	3. 4050
软件功能	3. 4077	软件设计验证	3. 4055
软件构造版	3. 4036	软件生存周期	3. 4088
软件规格说明评审	3. 4148	软件失效	3. 4075
软件过程	3. 4101	软件识别标签	3. 4079
软件过程成熟度	3. 4107	软件使用	3. 4158
软件过程改进计划	3. 2102, 3. 4105	软件使用者	3. 4045
软件过程改进建议	3. 4106	软件使用质量	3. 4125
软件过程描述	3. 4103	软件授权	3. 4073
软件过程评估	3. 4102	软件授权核对	3. 4074
软件过程相关文档	3. 4108	软件授权模式	3. 1630
软件过程元素	3. 4104	软件数据库	3. 4048
软件获取	3. 4023	软件特性	3. 4038
软件获取过程	3. 4024	软件特征	3. 4076
软件基线	3. 4032	软件体系结构	3. 4027
软件基线库	3. 4034	软件维护	3. 4089
软件基线审核	3. 4033	软件文档	3. 4159
软件即服务	3. 4029	软件系统	3. 4151
软件集成	3. 4081	软件系统元素	3. 4152
软件计划	3. 4100	软件相关组	3. 4165
软件开发笔记本	3. 4061	软件项	3. 4082
软件开发成本	3. 4057	软件项目	3. 4111
软件开发方法	3. 4065	软件项目管理计划信息	3. 4118
软件开发过程	3. 4067	软件项目生存周期	3. 4112
软件开发环境	3. 4059	软件项目生存周期 (SPLC)	3. 4113
软件开发计划	3. 4066	软件项目生存周期过程	3. 4116
软件开发库	3. 4063, 3. 4064	软件项目生存周期过程 (SPLCP)	3. 4117
软件开发全过程	3. 5112	软件项目生存周期模型	3. 4114
软件开发人员	3. 4056	软件项目生存周期模型 (SPLCM)	3. 4115
软件开发收入	3. 4062	软件信息安全性	3. 4146
软件开发文件	3. 4060, 3. 4061	软件需求	3. 4135
软件开发文件夹	3. 4061	软件需求分析	3. 4137
软件开发周期	3. 4058	软件需求工程	3. 4138
软件可靠性	3. 4131	软件需求管理	3. 4139
软件可靠性工程	3. 4132	软件需求规格说明	3. 4142
软件可靠性管理	3. 4133	软件需求阶段	3. 4140
软件库	3. 4083, 3. 4084	软件需求评审	3. 4136
软件能力评估	3. 4037	软件需求评审 (SRR)	3. 4141
软件配置	3. 4040	软件需求验证	3. 4143
软件配置管理	3. 3842, 3. 4043	软件许可方	3. 4087
软件配置管理计划	3. 3844	软件许可证	3. 4085
软件配置管理计划信息	3. 3843	软件与系统产品线	3. 4026
软件配置管理项	3. 4044	软件元素	3. 4069
软件配置控制板	3. 4041	软件在环仿真	3. 4080

软件支持	3. 4149	设计单元	3. 1338
软件质量	3. 4119	设计叠加	3. 1325
软件质量保证	3. 4120	设计对象	3. 1334
软件质量测量	3. 4127	设计方法学	3. 1323
软件质量管理	3. 4126	设计分析	3. 1310
软件质量计量	3. 4128	设计分析器	3. 1309
软件质量控制	3. 4122	设计概念	3. 1313
软件质量控制(SQC)	3. 4123	设计关系	3. 1329
软件质量目标	3. 4124	设计关注点	3. 1314
软件质量特性	3. 4121	设计规格说明	3. 1316, 3. 1317
软件质量外部测度	3. 1796	设计级别	3. 1322
软件质量要求	3. 4129	设计阶段	3. 1327
软件资产	3. 4030	设计可追溯性	3. 1337
软件资产管理	3. 4031	设计描述	3. 1316
软件资产管理工具	3. 3804, 3. 3805	设计模式	3. 1326
软拷贝图像	3. 4020	设计评审	3. 1331
软逻辑	3. 1435	设计缺陷	3. 1320
软失效	3. 4022	设计实体	3. 1319
S		设计视点	3. 1340
[宿]主机	3. 2045	设计视图	3. 1339
SAM 管理工具/软件资产管理工具	3. 3806	设计属性	3. 1311
SAM 项目范围/软件资产管理项目范围	3. 3803	设计思维	3. 1335
SAM 工具	3. 3805	设计特性	3. 1312
Scrum 主管	3. 3855	设计文档	3. 1316, 3. 1317
scrum 报告	3. 3856	设计需求	3. 1330
scrum 团队	3. 3857	设计语言	3. 1321
SEE 服务获取方	3. 3882	设计元素	3. 1318
SEE 服务供应商	3. 3883	设计原理	3. 1328
s 曲线	3. 3858	设计约束	3. 1315
三地址指令	3. 4713	设施	3. 1809
三加一地址指令	3. 4714	设置时间	3. 3969
三重底线	3. 4824	社会	3. 4019
三重约束	3. 4825	社会与伦理风险缓解	3. 4018
散点图	3. 3819	社区	3. 768
闪存	3. 1879	社区对象	3. 769
商业案例	3. 563	申请人	3. 198
商业模式画布	3. 567	身份风格视图	3. 2077
商业现货	3. 742	深度神经网络	3. 1253
商业现货软件产品	3. 743	深度学习	3. 1252
商用现货软件产品	3. 2668	神经网络	3. 2741
上层架构	3. 4850	神经元覆盖率	3. 2742
上传	3. 4893	审查	3. 2226
上下文敏感帮助	3. 974	审查/审核输出	3. 3712
上下文相关帮助	3. 974	审查输出	3. 3712
上下文因素	3. 975	审核	3. 373
设备	3. 1657	审核输出	3. 3712
设备接口任务	3. 1382	审核员	3. 377
设计	3. 1308	审核组	3. 374
设计标准	3. 1332	渗透式沟通	3. 2890
设计策略	3. 1333	生产	3. 3267

生产计划	3. 3270	实测结果	3. 94
生产率	3. 3271	实存储器	3. 3491
生产配置	3. 3224	实地址	3. 3490
生产商	3. 4399	实际成本	3. 90
生产与部署	3. 3268	实际持续时间	3. 92
生产者对象	3. 3219	实际利率	3. 1559
生成	3. 1975	实际折旧	3. 91
生成地址	3. 1974	实践	3. 1601
生存性	3. 4416	实践, 惯例	3. 3064
生存周期	3. 2416	实践元素	3. 1602
生存周期成本	3. 2418	实例化	3. 2239, 3. 4191
生存周期管理	3. 2419	实例化跟踪链接	3. 2238
生存周期过程	3. 2421	实例级操作	3. 2237
生存周期模型	3. 2420	实例级属性	3. 2236
生存周期评估	3. 2417	实例数据	3. 2675
声明	3. 1244	实施采购	3. 898
声明语言	3. 1245	实施定性风险分析	3. 2965
胜任特征	3. 778	实施过程	3. 4618
失效	3. 1814	实施后评审	3. 4474
失效安全	3. 1812	实施后审查	3. 4473
失效比	3. 1821	实施阶段	3. 2092
失效方式和影响分析	3. 1819	实施评级模块	3. 2093
失效类别	3. 1815	实施要求	3. 2094
失效率	3. 1821	实施一致性声明	3. 2067
失效模式	3. 1818	实施整体变更控制	3. 2964
失效强度	3. 1816	实施质量保证	3. 2966
失效强度目标	3. 1817	实施质量控制	3. 2967
失效数据	3. 1672	实时操作系统 (RTOS)	3. 3497
失效严重性	3. 1823	实时操作系统 (RTOS)	3. 3496
时标进度网络图	3. 4742	实时的	3. 3494
时间 Petri 网	3. 4731	实时调度理论	3. 3498
时间变化	3. 5021	实时时钟	3. 3495
时间表达	3. 4729	实体	3. 1631
时间的可变性	3. 4997	实体-关系-属性	3. 1665
时间分片	3. 4733	实体关系图	3. 1201, 3. 1639
时间关键任务	3. 4735	实体联系图	3. 1638
时间盒	3. 4734	实体实例	3. 1636
时间聚类	3. 4582	实体属性	3. 1632
时间类别	3. 4727	实体组件	3. 1633
时间内聚	3. 4583	实现	3. 2089, 3. 2090, 3. 3493
时间位置	3. 2451	实现可用性	3. 42
时间行为	3. 4726	实现类	3. 2091
时期	3. 1655	实型	3. 3492
时效功能	3. 4737	实验设计	3. 1324
时序分析	3. 4745	史诗	3. 1651
时序图	3. 4746	使能	3. 1587
时延	3. 2384	使能系统	3. 1588
时钟脉冲发生器	3. 687	使能行为	3. 1586
识别事务	3. 2074	使用错误	3. 4913
实参	3. 93	使用方式	3. 4907

使用所致损害	3. 2014	适配器	3. 99
使用相关质量要求	3. 4944	适应型生存周期	3. 101
使用信息	3. 2157	适应性	3. 96
使用指导信息	3. 2241	适应性策略	3. 103
使用质量（测度）	3. 3435	适应性方法	3. 100
使用质量测度	3. 3436	适应性维护	3. 102
使用周境	3. 973	适应性支持	3. 104
使自动化	3. 386	适用性	3. 1873
示例实例表	3. 3808	适用性测试	3. 4904
示例实例图	3. 3807	适用性的可识别性	3. 243
势	3. 604	收入备选方案	3. 3703
势约束	3. 605	收入函数	3. 2113
事故	3. 38	收尾过程组	3. 693
事后转储	3. 3060	收益	3. 470
事件	3. 1713	手动测试	3. 2518
事件报告	3. 2108	首选逻辑	3. 1435
事件跟踪	3. 1721	受测系统	3. 4510
事件管理者（IM）	3. 2107	受到管理和控制	3. 2507
事件记录	3. 1715	受限项表	3. 2425
事件接收器	3. 1718	受众	3. 372
事件历史	3. 1714	授权标签	3. 1628
事件排序逻辑	3. 1724	授权创建者/许可模式创建者	3. 1624
事件驱动的评审或活动	3. 1723	授权个人	3. 385
事件树分析	3. 1722	授权管理	3. 1629
事件同步	3. 1720	授权合并	3. 944
事件序列分析	3. 1716	授权模式	3. 1630
事件序列图	3. 1717	舒适性	3. 737
事件源	3. 1719	输出	3. 2893
事务	3. 4792	输出产品	3. 2898
事务分析	3. 4793, 3. 4794	输出断言	3. 2895
事务功能	3. 4799	输出原语	3. 2897
事务功能类型	3. 4800	输入	3. 2210
事务矩阵	3. 4796	输入 — 处理 — 输出图	3. 2221
事务模式	3. 4797	输入（类型）	3. 1640
事务透明性	3. 4798	输入/输出	3. 2061
事务文件	3. 4795	输入变量	3. 2212
事务中心设计	3. 4801	输入程序	3. 2217
视点	3. 5063	输入-处理-输出	3. 2219
视点（在系统上）	3. 5064	输入-处理-输出（IPO）图	3. 2220
视点陈述	3. 5066	输入-处理-输出图	3. 2325
视点对应关系	3. 5065	输入断言	3. 2213
视频随机存取存储器	3. 5057	输入回送	3. 2214
视频显示单元	3. 5083	输入类型	3. 1641
视频显示终端	3. 5056	输入输出耦合	3. 1147
视图	3. 5058	输入-输出耦合	3. 2218
视图投影	3. 5062	输入原语	3. 2216
视图图表	3. 5060	输入字段	3. 1644
视图元素	3. 5061	属性	3. 366
视图组件	3. 5059	属性抽样	3. 369
适合性	3. 4380	属性关系	3. 370

属性名称	3. 368	数据流追踪	3. 1206
属性实体类型	3. 371	数据流踪迹	3. 5013
术语管理系统	3. 4593	数据敏感故障	3. 1220
树结构图	3. 4818	数据模型	3. 1183
树状图	3. 4817	数据耦合	3. 1147
数据	3. 1134	数据配置	3. 1145
数据保护	3. 1188	数据漂移	3. 1156
数据保护影响评估	3. 1189	数据驱动	3. 1215
数据暴露	3. 1161	数据日期	3. 1148
数据表示技术	3. 1197	数据设计	3. 1154
数据部分	3. 1218	数据声明源语句	3. 1149
数据擦除	3. 1159	数据使用	3. 1209
数据仓库	3. 1196	数据视图	3. 1222
数据操作	3. 1180	数据收集和分析方法	3. 1171
数据抽象	3. 1135	数据收集技术	3. 1172
数据处理	3. 1186, 3. 1187	数据输入表	3. 1176
数据传输控制器	3. 1207	数据属性	3. 1139
数据存储	3. 1199	数据损坏	3. 1146
数据定义	3. 1152	数据特性	3. 1143
数据定义计算使用对	3. 1150	数据提供者	3. 1190
数据定义使用对	3. 1153	数据通信设备	3. 1144
数据定义谓词使用对	3. 1151	数据文件	3. 1163
数据动作	3. 1136	数据系统	3. 1205
数据断点	3. 1141	数据项	3. 1158, 3. 1178, 3. 1216
数据分析	3. 1137	数据泄露	3. 1179
数据分析技术	3. 1138	数据信息	3. 1175
数据格式	3. 1168	数据序列器	3. 1198
数据故障	3. 1162	数据移动	3. 1184
数据关键字	3. 1217	数据移动类型	3. 1184
数据函数	3. 1169	数据异常	3. 1160
数据函数类型	3. 1170	数据银行	3. 1140
数据缓冲寄存器	3. 1142	数据隐私	3. 1185
数据基	3. 1211	数据元素	3. 1158
数据集偏移	3. 1454	数据元素类型	3. 1157
数据记录	3. 1195	数据元素类型的数量	3. 2781
数据角色	3. 1219	数据值	3. 1210
数据结构	3. 1200	数据质量	3. 1191
数据结构图	3. 1201	数据质量测度	3. 1193
数据结构中心设计	3. 1202	数据质量模型	3. 1194
数据介质	3. 1182	数据质量特性	3. 1192
数据库	3. 1211	数据治理	3. 1173
数据库存	3. 1177	数据主体	3. 1204
数据库管理系统	3. 1213	数据追踪	3. 1206
数据库设计规范	3. 1212	数据字典	3. 1155
数据库视图	3. 1214	数据踪迹	3. 5013
数据类型	3. 1208	数据组	3. 1174
数据流	3. 1164	数据组类型	3. 1174
数据流测试	3. 1167	数字	3. 1398
数据流程图	3. 1166, 3. 1201	数字产品	3. 1401
数据流图	3. 1165, 3. 1166	数字的	3. 2782

数字通用光盘	3. 1405	隧道符号	3. 4834
数字信号处理	3. 1402, 3. 1403	损害	3. 2013
数字信号处理器	3. 1403, 3. 1404	损耗	3. 2476
数字信息内容资产	3. 1400	缩略图	3. 4722
数字资产	3. 1399	所关注实体	3. 1637
数字字符	3. 1397, 3. 1398	所关注系统	3. 4486
数组	3. 281	所属属性	3. 2911
刷新	3. 3536	所需输出	3. 3619
双倍数据速率 x2	3. 1503	锁	3. 2454
双倍数据速率同步动态随机存取存储器	3. 1502	锁存器	3. 1880, 3. 1881
双操作数指令	3. 1504, 3. 4843	锁定	3. 2455
双列直插式封装	3. 1514	T	
双系统启动	3. 1512	TCPI	3. 4559
双向可追溯性	3. 478	T 恤尺码估算法	3. 4833
二元选择构造	3. 1520	探测后	3. 3061
睡眠模式	3. 4247	探查	3. 3137
顺序标度	3. 2860	探索性测试	3. 1767
顺序的	3. 3916	逃脱	3. 1680
顺序构造	3. 3924	特定代码	3. 4194
顺序进程	3. 3919	特定地址	3. 4193
顺序聚类	3. 3917	特定符号	3. 4196
顺序内聚度	3. 3918	特定配置组	3. 4195
顺序图	3. 3915	特定学科模型	3. 1429
瞬时对象	3. 4809	特权指令	3. 3130
瞬态错误	3. 4808	特权状态	3. 1741, 3. 3131, 3. 4390
私有	3. 3128	特殊功能寄存器	3. 4189
私有类型	3. 3129	特殊原因	3. 4186
思考时间	3. 4707	特殊原因/可归因原因	3. 4187
思维导图	3. 2072, 3. 2624	特性	3. 655
思维导图,	3. 2071	特征	3. 1843
斯马特原则	3. 4197	特征标记	3. 3985
死锁	3. 1227	特征冻结	3. 1849
四地址指令	3. 1916	特征分支	3. 1844
四加一地址指令	3. 1917	特征工程	3. 1848
松弛时间	3. 1882	特征集	3. 1852
松耦合消息通信	3. 359	特征目录	3. 1845
搜索算法	3. 3861	特征实体	3. 656
速度	3. 5026	特征引用	3. 1851
速度图	3. 5027	特征语言	3. 1850
宿主计算机	3. 2044	特征约束	3. 1847
塑料引线芯片载体	3. 3033	提案评估技术	3. 3379
塑料有引线芯片载体	3. 3032	提出疑虑 / 陈述问题	3. 1921
算法	3. 150	提供仪器活动	3. 3389
算法分析	3. 151	提交	3. 745
算法漂移	3. 152	提交窗口	3. 748
算法语言	3. 153	提交混乱	3. 747
随机测试	3. 3467	提交信息	3. 746
随机存取存储器	3. 3468	提交者	3. 750
随机失效	3. 3466	提取法	3. 1802
隧道	3. 4836	提示	3. 3373

填充	3. 2915	同行评审	3. 2958
条件	3. 887	统计测试模型	3. 4285
条件分支	3. 891	统计抽样	3. 4284
条件过程	3. 895	统计管理过程	3. 4286
条件控制结构	3. 892	统计过程控制	3. 4283
条件码	3. 888	统一建模语言	3. 4865, 3. 4866
条件任务	3. 896	统一软件开发过程	3. 3475
条件条目	3. 889	统一资源标识符	3. 4867
条件跳转	3. 894	统一资源定位符	3. 4868
条件文本	3. 897	头脑风暴	3. 529
条件信息	3. 893	投标价	3. 476
条件阈值	3. 982	投标书	3. 3378
条件桩	3. 890	投标者	3. 477
跳转	3. 2357	投产库	3. 3234
停车场图	3. 2940	投资回报率	3. 3693
停机时间	3. 1505	投资回报率 (ROI)	3. 3694
停用/去激活	3. 1226	投资组合管理报告	3. 3052
停止	3. 4298	投资组合管理记录	3. 3051
通告	3. 186	透明度	3. 4815
通过 / 失败准则	3. 2947	图边界	3. 1387
通信接口	3. 763	图标	3. 2066
通信领域	3. 766	图表标题	3. 1393
通信内聚	3. 765	图号	3. 1390
通用程序单元	3. 1979	图解技术	3. 1394
通用串行总线	3. 4882	图例	3. 2396
通用寄存器	3. 1966	图示符号	3. 1993
通用配置文件组	3. 1978	图式	3. 3840
通用配置项	3. 754	图形处理器	3. 1992
通用人工智能	3. 1962	图形符号	3. 1994
通用实践	3. 1977	图形特征	3. 1388
通用实体	3. 1976	图形特征引用	3. 1389
通用输入 / 输出端口	3. 1965	图形信息	3. 1991
通用输入/输出端口	3. 1973	图形用户界面	3. 1999
通用输入/输出端口 (GPIO)	3. 1964	图页	3. 1391
通用数据保护条例	3. 1963	图引用	3. 1392
通用系统特征	3. 1967	图注	3. 3485
通用性	3. 759, 3. 1968	团队	3. 4561
通用性指数	3. 760	团队绩效域	3. 4562
通用异步收发器	3. 4881	团队选拔计划	3. 4563
同伴检查	3. 542	推荐性条款	3. 3504
同步	3. 4429	推理	3. 2147
同步的	3. 4430	推理技术	3. 3501
同步动态随机存取存储器	3. 4431	退出标准	3. 1750
同步请求	3. 4439	退出动作	3. 1749
同步图形随机存取存储器	3. 4432	退化	3. 1509
同步消息通信	3. 4433, 3. 4438	退役	3. 3687
同步状态	3. 4428	退役阶段	3. 3688
同构冗余	3. 2043	蜕变关系	3. 2690
同时的	3. 3993	吞吐量	3. 4717
同时递归	3. 3994	吞吐量参考值	3. 4721

吞吐量评级值	3. 4720	完工偏差	3. 5015
吞吐量图表	3. 4718	完工尚需估算	3. 1686
吞吐率	3. 4719	完工尚需绩效指数	3. 4751
妥善定义的过程	3. 5107	完工预算	3. 544
V		完工预算 (BAC)	3. 545
Verilog 硬件描述语言	3. 5046	完美信息的期望值	3. 1757
V 模型	3. 5085	完全簇 / 完备簇	3. 4768
W		完全映射	3. 4767
(网的) 初始标记	3. 2199	完全正确性	3. 4771
(网络的) 标记	3. 2528	完善性维护	3. 2962
Web 内容可访问性指南	3. 5101	完善性支持	3. 2963
外包	3. 2900	完整的 ICOM 编码	3. 788
外部测度 (系统与软件质量)	3. 1795	完整过程	3. 789
外部测量	3. 1794	完整性等级要求	3. 2257
外部查询	3. 1790	完整性级别	3. 2254
外部的	3. 1784	完整性级别定义权威	3. 2256
外部接口文件	3. 1791	完整性级别方案	3. 2258
外部接口需求	3. 1792	完整性级别声明	3. 2255
外部可变性	3. 1801	网络开放端	3. 2738
外部逻辑文件	3. 1793	网络路径	3. 2739
外部软件质量	3. 1800	网络视频录像机	3. 2783
外部事件	3. 1787	网站地图	3. 4007
外部视角	3. 2899	网站负责人	3. 5102
外部输出	3. 1798	往返时间	3. 4838
外部输入	3. 1789	危害警示	3. 2018
外部输入输出设备	3. 1788	危害情形	3. 2020
外部属性	3. 1785	危害事件	3. 2019
外部性质的质量测度	3. 3447	危机	3. 1085
外部依赖	3. 1786	危机管理	3. 1086
外部质量	3. 1799	危险	3. 1129, 3. 2016
外层基数	3. 2892	危险识别	3. 2017
外键	3. 1894, 3. 1895	危险状态	3. 1130
外廓	3. 3272	威胁建模	3. 4712
外联网	3. 1803	威胁与风险评估	3. 4775
外围设备	3. 2980	微操作	3. 2611
外延集	3. 1781	微程序	3. 2612
完备簇	3. 4770	微程序的计算机	3. 2614
完备判定表	3. 790	微程序设计	3. 2615
完备性	3. 791	微控制器单元	3. 2608
完成	3. 1499	微码汇编器	3. 2607
完成到完成关系	3. 1853	微芯片	3. 2246
完成的定义	3. 1499	微型发生器	3. 2609
完成-开始	3. 1866	微指令	3. 2610
完成时间定理	3. 794	违规	3. 5067
完成-完成	3. 1865	唯一性约束	3. 4872
完成准则	3. 793	维度	3. 1407
完代码	3. 792	维护	3. 2500
完工差异	3. 5016	维护程序	3. 2504
完工估算	3. 1684	维护计划	3. 2503
完工估算 (EAC)	3. 1685	维护计划 维护基础设施	3. 2505

维护阶段	3. 2502	问题登记表	3. 2939
维护手册	3. 2501	问题定义	3. 3139
维护知识库活动	3. 2498	问题描述	3. 3139
维基	3. 5114	问题状态	3. 3141
维修级别分析	3. 2405	无风险最低期望收益率	3. 3764
伪操作	3. 3391	无故障性	3. 1841
伪静态随机存取存储器	3. 3395	无回复的紧耦合消息通信	3. 4437
伪码	3. 3392	无回复的同步消息通信	3. 4437
伪预言	3. 3393	无监督学习	3. 4887
伪指令	3. 3391	无健康风险	3. 1926
伪状态	3. 3394	无脚本测试	3. 4884
尾部	3. 4787	无经济风险	3. 1924
尾随判定	3. 4788	无量纲标度	3. 2756
委派	3. 1270	无生命安全风险	3. 1927
委托方	3. 4208	无条件分支	3. 4856
为用户打印的信息	3. 3117	无条件跳转	3. 4857
未解释的	3. 4870	无我程序设计	3. 1563
未新开发源语句	3. 2754	无线应用协议	3. 5116
未展开论证项	3. 4862	无向图	3. 4863
位	3. 496	无效测度	3. 4987
位控制	3. 498	无应答的同步消息通信	3. 4436
位置透明性	3. 2453	物理架构	3. 3001
谓词数据使用	3. 3073	物理接口	3. 3004
温度	3. 4576	物理配置审计	3. 3002
文本	3. 4698	物理配置审计 (PCA)	3. 3003
文本编辑程序	3. 4699	物理需求	3. 3005
文本处理	3. 4701	物理源语句	3. 3006
文本处理 / 文档处理	3. 4702	物联网	3. 2297
文本处理文字处理	3. 4703	物料清单	3. 480
文本页	3. 4700	物料清单 (BOM)	3. 481
文档	3. 1461	误差	3. 1669
文档等级	3. 2403		
文档管理系统	3. 1464	X	
文档汇集、文档编制, 文档管理	3. 1468	XML 模式定义	3. 5144
文档集	3. 1465	XML 模式文档	3. 5145
文档计划	3. 1469	系统	3. 4446
文档控制	3. 1463	系统/子系统规格说明	3. 4218
文档类型定义	3. 1466, 3. 1511	系统安全	3. 4502
文档类型定义 (DTD)	3. 1467	系统边界	3. 4455
文档审查	3. 1470	系统表	3. 4507
文档树	3. 1471	系统操作概念	3. 2850
文件分析	3. 1462	系统测试	3. 4509
文件夹	3. 3053	系统颠簸	3. 4709
文字式常数	3. 1860	系统分解结构	3. 4456
稳定分支	3. 4221	系统分析	3. 4514
稳定过程	3. 4222	系统分析报告	3. 4448
稳定阶段	3. 4220	系统分析策略	3. 4449
稳定性模式	3. 4219	系统分析记录	3. 4447
问题	3. 2328, 3. 3138	系统跟进	3. 4473
问题报告	3. 3140	系统跟踪	3. 4472, 3. 4474
		系统工程	3. 4516

系统工程项目	3. 4517	系统需求审查	3. 4499
系统工程管理计划	3. 3910, 3. 4518	系统需求文档	3. 4216
系统工程项目	3. 3914, 3. 4518	系统要素	3. 4467
系统项目组	3. 4470	系统要素描述	3. 4468
系统功能定义	3. 4475	系统要素文档	3. 4469
系统功能接口识别	3. 4477	系统与软件产品质量	3. 4450
系统功能识别	3. 4476	系统与软件工程	3. 4217
系统规范	3. 4504	系统元素	3. 4163
系统和软件质量的外部测度	3. 1797	系统之系统	3. 4487
系统化重用	3. 4512	系统之系统工程	3. 4488
系统集成	3. 4479	系统支持	3. 4506
系统集成测试	3. 4519	细分	3. 3532
系统集成模块	3. 4480	狭义人工智能	3. 2728
系统架构描述	3. 4453	下推式存储器	3. 3404
系统架构师	3. 4451	下溢异常	3. 4859
系统架构依据	3. 4454	先进先出队列	3. 1857
系统接口任务	3. 4481	先期调页	3. 193
系统结构	3. 4505	先期缓冲	3. 192
系统开发	3. 4462	显式地址	3. 1763
系统开发生存周期	3. 4515	显式需求	3. 1765
系统开发生存周期	3. 3859	显现需求	3. 3702
系统开发周期	3. 4463	显性知识	3. 1764
系统可靠性	3. 4492	现场可编程门阵列	3. 1855
系统库	3. 4482	现场总线	3. 1856
系统描述	3. 4458	现货产品	3. 2802
系统模型	3. 4485	现有软件	3. 1747
系统配置文件	3. 4491	限定符	3. 3416
系统软件	3. 4503	限制	3. 2424
系统设计	3. 4459	线程	3. 4710
系统设计描述	3. 4460	线性插值	3. 2427
系统设计评审	3. 3860	陷入	3. 4816
系统设计原理	3. 4461	相对地址	3. 3555
系统生存周期	3. 4483	相对估计	3. 3557
系统实体	3. 4471	相对链频率	3. 3556
系统特性	3. 4457	相对时间	3. 3559
系统体系结构	3. 4452	相对装入程序	3. 3558, 3. 3583
系统危害	3. 4478	相关利益相关者	3. 3566
系统维护	3. 4484	相互包容	3. 2718
系统文档	3. 4464	相互关系图	3. 2305
系统效能	3. 4465	相互排斥	3. 2716
系统效能分析	3. 4466	相互作用点	3. 2264
系统性能规范	3. 4213	相似性分组	3. 132
系统性失效	3. 4511	相位门	3. 3000
系统需求	3. 4493	相依耦合	3. 2955
系统需求定义策略	3. 4498	详细功能点分析	3. 1353
系统需求定义记录	3. 4497	详细功能点计数	3. 1354
系统需求规范	3. 4500	详细设计	3. 1350
系统需求规格	3. 4495	详细设计规格说明	3. 1349
系统需求可追溯性	3. 4501	详细设计阶段	3. 1351
系统需求评审	3. 4494	详细设计描述	3. 1348, 3. 1349

详细设计评审	3. 1352	项目进度管理	3. 3353
响应对象	3. 3673	项目进度网络图	3. 3354, 3. 3355
响应时间	3. 3674	项目经理	3. 3339
响应式方法	3. 3480	项目控制	3. 3318
响应式网页设计	3. 3678	项目库	3. 4064
响应性	3. 3679	项目平衡	3. 3314
想法	3. 2072	项目启动会议	3. 2376
想法导图	3. 2072	项目群管理	3. 3288
向导	3. 5118	项目日历	3. 3316
向前传递	3. 1913	项目生存周期	3. 3329
向上兼容的	3. 4894	项目时间管理	3. 3365
向上压缩	3. 4895	项目团队	3. 3363
向下兼容	3. 1507	项目团队目录	3. 3364
向下压缩	3. 1508	项目文件	3. 3320
项	3. 4584	项目愿景陈述	3. 3366
项目	3. 3313	项目整合管理	3. 3327
项目簿	3. 3321	项目质量管理	3. 3346
项目采购管理	3. 3345	项目治理	3. 3325
项目成本管理	3. 3319	项目状态报告	3. 3361
项目定义的软件过程	3. 3370	项目资金需求	3. 3324
项目范围	3. 3356	项目资源管理	3. 3348
项目范围管理	3. 3357	项目资源约束	3. 3347
项目范围说明书	3. 3358	项目总体风险	3. 2901
项目风险	3. 3349	项目组合	3. 3344
项目风险概况	3. 3351	项目组合管理	3. 3049
项目风险管理	3. 3350	项目组合管理计划	3. 3050
项目负责人	3. 3328	象征	3. 4421
项目干系人管理	3. 3360	消费品	3. 955
项目工作绩效领域	3. 3367	消费者对象	3. 954
项目功能点数	3. 3322	消费者软件包	3. 956
项目功能规模	3. 3323	消息	3. 2581
项目沟通管理	3. 3317	消息摘要 5	3. 2582
项目管理 (PM)	3. 3330	消息摘要算法 5	3. 2582
项目管理办公室	3. 3332	小外形薄型封装	3. 3979
项目管理过程组	3. 3334	小外形封装	3. 4015
项目管理计划	3. 3333	小微企业	3. 5054
项目管理人员	3. 3336	小型集成电路	3. 4014
项目管理软件	3. 3335	小中取大准则	3. 2540
项目管理团队	3. 3338	校验	3. 660
项目管理系统	3. 3337	效益成本分析	3. 471
项目管理知识领域	3. 3331	协商	3. 2736
项目规范	3. 3359	协同交互	3. 1041
项目规划	3. 3343	协同例程	3. 1046
项目后评估	3. 3054	协议	3. 148, 3. 3384
项目化组织	3. 3369	协议对象	3. 3386
项目计划	3. 3342	协议实现一致性声明	3. 3385
项目剪裁策略	3. 3362	协作	3. 724
项目简报	3. 3315	协作式系统之系统	3. 726
项目阶段	3. 3341	协作图	3. 725
项目进度	3. 3352	写	3. 5141

写类型	3. 5142	信息系统要求	3. 2181
写作参考	3. 5143	信息项	3. 2161
写作系统	3. 383	信息项类型	3. 2163
写作语言	3. 382	信息项内容	3. 2162
心理测验学	3. 3396	信息需要	3. 2167
芯片	3. 2246	信息隐蔽	3. 2160
新源语句	3. 2743	行波时钟	3. 3720
新增源代码语句	3. 105	行为	3. 455
信道容量	3. 650	行为兼容性	3. 457
信号	3. 3982	行为模型	3. 456
信号接口	3. 3983	行为模型方面	3. 459
信号接口签名	3. 3984	行为驱动开发	3. 460
信号量	3. 3907	行为特征	3. 458
信息	3. 2151	形参	3. 1902
信息安全	3. 2175	形成性结构	3. 1911
信息安全事件	3. 2176	形成性评价	3. 1912
信息安全性	3. 3868	形式	3. 1897
信息安全性测试	3. 3881	形式、适配和功能	3. 1898
信息安全政策	3. 2177	形式规格说明	3. 1907
信息部分	3. 2168	形式化	3. 1910
信息产品	3. 2171	形式化设计	3. 1899
信息处理系统	3. 2170	形式化需求语言	3. 1904, 3. 1905
信息处理要求	3. 2169	形式验证	3. 1909
信息处理中心	3. 827	形式语言	3. 1901
信息发射源	3. 2173	型号名称	3. 2652
信息分析	3. 2152	性能	3. 2970
信息供应	3. 2172	性能报表	3. 2974
信息管理计划	3. 2164	性能测度	3. 2554
信息管理系统	3. 2165	性能度量基线	3. 2973
信息技术服务	3. 2185	性能分析	3. 2971
信息技术公司治理	3. 1047	性能规格	3. 2977
信息技术项目	3. 2184	性能基线	3. 2972
信息技术资产	3. 2182	性能需求	3. 2975
信息技术资产管理	3. 2329	修订通知	3. 2775
信息技术资产管理体系	3. 2183	修复	3. 3591
信息架构	3. 2153	修改后的源语句	3. 2667
信息检索	3. 2174	修改请求	3. 2665
信息结构	3. 2178	修改条件/决策覆盖测试	3. 2666
信息结构标识符	3. 2180	虚参数	3. 1516
信息结构创建者	3. 2179	虚拟测试环境	3. 5078
信息开发	3. 2156	虚拟存储器	3. 5075
信息类型	3. 2186	虚拟地址	3. 5069
信息领域	3. 2188	虚拟的	3. 5068
信息模型	3. 2166	虚拟化	3. 5079
信息内容	3. 2154	虚拟机	3. 5070
信息请求	3. 3718	虚拟空间	3. 5074
信息设计	3. 2155	虚拟团队	3. 5077
信息视点	3. 2187	虚拟系统之系统	3. 5076
信息收集技术	3. 2159	虚拟样机	3. 5071
信息通信技术产品	3. 2068	虚拟样机系统	3. 5072

虚拟引用	3. 5073	循环控制变量	3. 2475
需方	3. 46	循环搜索	3. 1126
需求	3. 3620	循环体	3. 2473
需求标准	3. 3622	训练数据	3. 4789
需求表	3. 3616	训练组	3. 4790
需求测试矩阵	3. 4686	Y	
需求陈述	3. 2735	(一个类的)自反祖先	3. 3535
需求的导出	3. 1295	(异步任务)	3. 196
需求的可变性	3. 4994	压力测试	3. 4323
需求分配	3. 3623	哑终端	3. 1515
需求分析	3. 3624	严重错误/勘误	3. 1668
需求跟踪工具	3. 3640	严重性	3. 3970
需求跟踪矩阵	3. 3639	研发工作	3. 1598
需求工程	3. 3630	研发工作单元	3. 1599
需求管理	3. 3632	研发工作组成单元	3. 1600
需求管理计划	3. 3633	衍型	3. 4295
需求规格说明	3. 3636	掩码	3. 2531
需求规格说明语言	3. 3637	掩膜只读存储器	3. 2532
需求划分	3. 3634	演示	3. 1282
需求获取	3. 3629	演示设备	3. 3094
需求阶段	3. 3635	验收	3. 23
需求可追溯性	3. 3638	验收测试	3. 26
需求领域资产	3. 1477	验收测试驱动的开发	3. 25
需求评审	3. 3621	验收的可交付成果	3. 27
需求确认	3. 3641	验收准则	3. 24
需求属性	3. 3625	验证	3. 5032, 3. 5043
需求推导	3. 3626	验证报告	3. 5038
需求文档	3. 3627	验证测试	3. 5041
需求文件	3. 3628	验证程序	3. 4961
需求验证	3. 3642	验证动作	3. 5033
需求验证表	3. 4686	验证发起人	3. 5039
需求验证和可追溯性矩阵	3. 3643	验证范围	3. 5045
需求逐级分解	3. 3631	验证方法	3. 5037
许可模式	3. 2414	验证工具	3. 5042
许可证	3. 2412	验证和确认	3. 5034
许可证发放标准	3. 2415	验证和确认计划	3. 5036
许可证合规审计	3. 2413	验证记录	3. 4962
序断点	3. 3371	验证交叉引用矩阵	3. 4686, 3. 5024
序言断点	3. 3372	验证可交付成果	3. 5044
选择性依赖关系	3. 1435	验证目标	3. 4540
选择性转储	3. 3889	验证系统	3. 5040
选择性踪迹	3. 3890	验证与确认工作	3. 5035
寻址(动词)	3. 110	样式	3. 4343
寻址范围	3. 117	要点设计	3. 3039
寻址方式	3. 116	要求	3. 297
寻址异常	3. 115	业务关键产品	3. 564
询问	3. 2306	业务价值	3. 572
循环	3. 2471	业务流程	3. 569
循环断言	3. 2472	业务目标	3. 568
循环控制	3. 2474	业务信息管理	3. 566

业务需求	3. 570	异/反常终止	3. 4
业务需求规范	3. 571	异步的	3. 355
业务影响分析	3. 565	异步任务	3. 195
页	3. 2916	异步输入输出设备	3. 357
页 0	3. 2923	异步输入输出设备接口任务	3. 358
页表	3. 2921	异步通信接口适配器	3. 356
页交换	3. 2920	异步消息通信	3. 359, 3. 360
页亏	3. 2917	异常	3. 1727
页面参考	3. 2919	异常处理	3. 1728
页面类型信函	3. 2922	抑制弧	3. 2194
页帧	3. 2918	易安装性	3. 2229
一次性可编程	3. 2807	易安装性测试	3. 2230
一次性密码/一次性编程	3. 2806	易访问性	3. 37
一级地址	3. 1409	易访问性测试	3. 36
一维性	3. 4864	易失性存储器	3. 5087
一线软件经理	3. 1872	易用性	3. 4896
一致的	3. 942	易用性测试	3. 4905
一致内聚度	3. 723	意识要求	3. 403
一致性	3. 941	溢出异常	3. 2903
一致性点	3. 932	因果图	3. 616
一致性工作	3. 933	因果图法	3. 617
一致状态	3. 943	阴影类	3. 3974
依从性	3. 800	引出	3. 1768
依赖	3. 1291, 3. 2466	引导	3. 512
依赖实体	3. 1289	引导程序	3. 514
依赖项导出	3. 1287	引导加载程序	3. 515
依赖性	3. 1283	引导模式	3. 513
依赖性确定	3. 1286	引导式研讨会	3. 1808
依赖状态类	3. 1290	引领式创业画布	3. 2389
仪表盘	3. 1133	引入	3. 2096
移动设备	3. 2637	引用的文件类型	3. 1861
移动性规约	3. 2638	引用文件类型	3. 1930
遗留软件	3. 2395	引用文件类型 (FTR)	3. 1862
遗留硬件	3. 2394	引址调用	3. 581, 3. 582
已安装的功能点计数	3. 2234	隐错播种	3. 1678
已定义过程	3. 1262	隐含的要求	3. 2095
已公布模型	3. 3400	隐私风险	3. 3127
已接受系统或系统元素	3. 28	隐私功能	3. 3122
已确立需求	3. 1682	隐私控制	3. 3123
已删除的源语句	3. 1271	隐私框架	3. 3124
已完成工作的实际成本	3. 90	隐私要求	3. 3126
已验证测量	3. 4957	隐私原则	3. 3125
以前开发的软件	3. 3101	隐性问题	3. 1131
以人为本的设计	3. 2052, 3. 2053	印刷电路板 (PCB)	3. 3113
以人为中心的设计	3. 2054	印刷电路板组装 (PCBA)	3. 3115
以人为中心的质量	3. 2055	印刷文件	3. 3116
以事务为中心的设计	3. 4794	印制电路板	3. 3112
以数据结构为中心的设计	3. 1203	印制电路板组件	3. 3114
以用户为中心的设计	3. 2053	应急储备	3. 981, 3. 3647
以转换为为中心的设计	3. 4806, 3. 4807	应急方案	3. 980

应急事件	3. 978	映像程序	3. 2523
应急支持	3. 1578	硬故障	3. 2004
应用	3. 2573	硬件	3. 2012
应用[程序]生成器	3. 214	硬件标识方案	3. 2011
应用测试	3. 233	硬件标识符生成器	3. 2010
应用测试资产	3. 232	硬件工程	3. 2009
应用程序	3. 199	硬件配置项	3. 2006, 3. 2008
应用程序边界	3. 205	硬件配置项 (HWC1)	3. 2007
应用程序功能点计数	3. 212	硬件要求规范	3. 2047
应用程序功能规模	3. 213	硬件资产管理	3. 2005
应用程序接口 (API) 管理平台	3. 221	硬拷贝	3. 2003
应用程序接口管理平台	3. 222	永久性	3. 2981
应用程序界面	3. 223	永久许可证	3. 2983
应用程序内编程 (IAP)	3. 2104	泳道	3. 4420
应用对象	3. 218	涌现性	3. 1575
应用工程	3. 209	用处	3. 4915
应用工程过程	3. 210	用户	3. 1117, 3. 4916
应用管理	3. 216	用户帮助	3. 4917
应用管理职能	3. 200	用户参与度	3. 4920
应用管理组织	3. 217	用户错误保护	3. 4921
应用架构	3. 201	用户的屏幕信息	3. 2811
应用接口	3. 215	用户定义的可靠性类	3. 4943
应用可变性模型	3. 234	用户功能需求	3. 1960
应用框架	3. 211	用户功能需求的特性	3. 657
应用领域	3. 202	用户故事点数	3. 4937
应用领域 (规格)	3. 1854	用户交互	3. 4925
应用内编程	3. 2103	用户接口	3. 4926
应用配置	3. 207	用户界面	3. 4926
应用软件	3. 230	用户界面对象	3. 4930
应用设计	3. 208	用户界面美学	3. 4927
应用实现	3. 224	用户界面任务	3. 4931
应用特定需求	3. 237	用户界面元素	3. 4928, 3. 4930
应用特定组件	3. 236	用户界面元素 用户界面对象	3. 4929
应用问题	3. 220	用户可控性	3. 4918
应用需求分析	3. 225	用户可识别的	3. 4946
应用需求管理	3. 227	用户类型	3. 4940
应用需求规格说明	3. 228	用户视图	3. 4941
应用需求获取	3. 226	用户手册	3. 4932
应用需求验证和确认	3. 229	用户体验	3. 4922
应用需求资产	3. 204	用户通知	3. 4934
应用资产	3. 203	用户文档集	3. 4919
应用组合	3. 219	用户文档系统测试	3. 4508
应用组件	3. 206	用户-系统交互	3. 4947
影响	3. 126	用户系统交互要求	3. 4938
影响程度	3. 1268	用户信息	3. 2158
影响分析	3. 2083	用户需求	3. 4936
影响图	3. 2148	用户需要	3. 4933
映射	3. 2524	用户友好的	3. 4945
映射完整性	3. 2525	用户指南	3. 4924
映射文件	3. 2526	用户终端	3. 4939

用户状态	3. 3141	预测方法	3. 3075
用户资料	3. 4935	预测生存周期	3. 3076
用户组	3. 4923	预测试迭代	3. 3096
用例	3. 607, 3. 4908	预测指标	3. 3077
用例规范	3. 4912	预测指标值	3. 3078
用例模型	3. 4911	预处理	3. 3090
用例实例	3. 4910	预处理器	3. 3091
用例图	3. 4909	预存储	3. 3095
用时代件	3. 4728	预打印	3. 3097
用于评价的技术	3. 1710	预防措施	3. 3098
优化分析	3. 2853	预防性维护	3. 3099
优化过程	3. 2854	预防性支持	3. 3100
优势	3. 4321	预防原则	3. 3067
优势、劣势、机遇和威胁分析	3. 4322	预估功能点分析	3. 2136
优先关系	3. 3069	预期结果	3. 1343, 3. 1754
优先级反转	3. 3121	预期用途	3. 2261
优先级方案	3. 3119	预算	3. 543
优先级矩阵	3. 3118	预算编制	3. 547
优先级中断	3. 3120	预算价	3. 546
优选逻辑	3. 1435	预探测	3. 3089
有限状态机	3. 1867	元	3. 2583
有向图	3. 1421	元编译程序	3. 2586
有向系统之系统	3. 1422	元对象	3. 2597
有效地址	3. 1556	元对象设施	3. 2598
有效的	3. 4954	元对象设施 MOF	3. 2599
有效过程	3. 1560	元关系	3. 2600
有效完整许可证	3. 1557	元类	3. 2585
有效性测度	3. 2553	元模型	3. 2594
有效指令	3. 1558	元模型元素	3. 2595
有形性	3. 4530	元实体	3. 2588
有意义的	3. 2551	元属性	3. 2584
语法	3. 4443	元数	3. 279
语法错误	3. 4441, 3. 4442	元数据	3. 2587
语法一致	3. 4440	元素	3. 1568
语境	3. 970	元素类型	3. 1569
语境模型	3. 976	元语言	3. 2589
语句测试	3. 4265	元元关系	3. 2593
语句覆盖率	3. 4262	元元模型	3. 2592
语言标准	3. 2382	元元实体	3. 2591
语言处理程序	3. 2381	元元属性	3. 2590
语义	3. 3906	元祖	3. 4837
语义变异点	3. 3905	原料	3. 2533
语义标签	3. 3902	原始的	3. 3110
语义传递语言	3. 3903	原始来源声明	3. 2887
语义传输语言	3. 3904	原始类型	3. 3111
语义错误	3. 3901	原始设备制造商	3. 2886
语义一致性	3. 3900	原始属性	3. 2885
浴盆曲线	3. 451	原型	3. 2640, 3. 3387
预编译程序	3. 3070	原型法	3. 3388
预测	3. 3074	原因分析	3. 615

原子类型	3. 362	运行中	3. 4890
原子性	3. 363	运营	3. 2843
原子值	3. 3813	运营构想	3. 868
原子组件模型	3. 361	运营构想文档	3. 869
源[代]码	3. 4173	运营和维护成本	3. 2824
源程序	3. 4181	Z	
源代码生成器	3. 4174	“直至”语句	3. 4888
源代码行数	3. 4179	夭折, 异常中止	3. 5
源代码组织	3. 4175	阈值	3. 4715
源地址	3. 4172	阈值覆盖	3. 4716
源节点	3. 4180	灾难恢复	3. 1427
源文档	3. 4177	灾难恢复计划	3. 1428
源文件	3. 4176	灾难性失效	3. 610
源语句	3. 4182	再工程	3. 3514
源语言	3. 4178	再现性	3. 3610
远程批处理入口	3. 3589	再专业化	3. 3672
远程终端仿真器	3. 3590	在特定场所对不同系统进行的组装	3. 3031
远程作业入口	3. 3588, 3. 3589	在系统编程	3. 2244
愿景声明	3. 5081	在线编译器	3. 2121
约定	3. 1035	在线仿真器	3. 2109
约束	3. 949	在线仿真器 (ICE)	3. 2110
约束依赖	3. 950	在线文档	3. 2808
云服务	3. 696	在役	3. 2223
云服务客户	3. 697	在役参考模型	3. 2225
云服务提供商	3. 698	在役周期	3. 2224
云互操作性	3. 695	载体	3. 606
云计算	3. 694	早期失效期	3. 1547
运算	3. 2823	责任	3. 1518
运行	3. 2823, 3. 3781	责任分配矩阵	3. 3677
运行报告	3. 2833	责任伦理	3. 1519
运行测试与评估	3. 2842	增量	3. 2116
运行策略	3. 2834	增量编译程序	3. 1037, 3. 2120
运行产品	3. 2838	增量编译器	3. 2121
运行场景	3. 2840	增量方法	3. 2118
运行方式	3. 3782	增量分析	3. 2117
运行概念	3. 2835	增量开发	3. 2122
运行和维护阶段	3. 2825	增量生产率	3. 2125
运行环境	3. 2837, 3. 3787	增量生存周期	3. 2124
运行可靠性	3. 2839	增量式	3. 2105
运行可用性	3. 194	增量收入	3. 2119
运行流	3. 3783	增量投资	3. 2123
运行模式	3. 2821	增强	3. 1618
运行软件	3. 2841	增强的能力	3. 1617
运行失效	3. 2803	增强现实系统	3. 378
运行时	3. 3785	增强项目	3. 1619
运行时环境	3. 3786	增强项目功能点分析	3. 1620
运行时间	3. 3784, 3. 3785	增强项目功能点计数	3. 1621
运行使能系统需求	3. 2828	增强项目功能点计数 (EFP)	3. 1622
运行态	3. 3782	增项维护	3. 107
运行约束	3. 2827	增值经销商	3. 4985

展开	3. 4889	支持人员小时	3. 4410
占用	3. 573	支持软件	3. 4409
占用时段	3. 574	支持手册	3. 4408
栈指针	3. 4223	支持向量机	3. 4412
战略计划	3. 4315	支持型活动	3. 2404
战略文件	3. 4317	支付矩阵	3. 2957
战略信息技术资产管理计划	3. 4314	知情同意	3. 2189
战略性 IT 资产管理计划	3. 4313	知识库	3. 2379
战略制品	3. 4316	知识体系	3. 508
战略资产管理计划	3. 4312	知识域	3. 2378
站立会议	3. 4249	执行	3. 4391
障碍	3. 2086	执行（动）	3. 1732
招标	3. 3614	执行（名）	3. 1735
招标书	3. 2322	执行程序	3. 1020, 3. 4391
招标邀请	3. 2321	执行过程	3. 1734, 3. 2978
召回率	3. 3502	执行过程组	3. 1733
折现回收期	3. 1430	执行监控程序	3. 1737
真反例	3. 4826	执行人	3. 1743
真实世界的对象	3. 3499	执行时间	3. 1738
真实性	3. 379	执行时间理论	3. 1739
真正例	3. 4827	执行系统	3. 1020
诊断	3. 1384	执行效率	3. 1736
诊断手册	3. 1385	执行状态	3. 1741, 3. 4390
挣值	3. 1548	执行踪迹	3. 1740
挣值（EV） 已完成工作预算成本（BCWP）	3. 1549	直接测量	3. 1414
挣值分析	3. 1550	直接测量值	3. 1418
挣值管理（EVM）	3. 1551	直接成本	3. 1410
挣值技术	3. 1552	直接地址	3. 1408, 3. 1409
正常运行响应	3. 2776	直接度量	3. 1417
正常运行终止	3. 2777	直接工时	3. 1419
正常支持响应	3. 2779	直接劳动力	3. 1413
正交可变性模型	3. 2889	直接内存访问	3. 1415
正确性	3. 1053	直接内存访问控制器	3. 1416
正确性证明	3. 3375	直接人工	3. 1412
正式测试	3. 1908	直接用户	3. 1420
正式合格评审	3. 1903	直接指令	3. 1411
正式评估过程	3. 1900	直线编码	3. 4311
正式审查	3. 1906	直线代码	3. 4310
正文元数据	3. 507	值得信赖的数据	3. 4831
正则表达式	3. 3544, 3. 3545	值寄存器	3. 4982
正则表达式（Regex）	3. 3545	值列表约束	3. 4979
证据	3. 1725	值追踪	3. 1206
政策与程序	3. 3043	值踪迹	3. 5013
政府现货	3. 1988	职责	3. 3675
支撑系统需求	3. 1589	职责分配矩阵	3. 3676
支持的产品	3. 4414	只读	3. 3487
支持活动组	3. 4406	只读存储器	3. 3488
支持阶段	3. 4411	只读光盘存储器（CD-ROM）	3. 770
支持经理	3. 4407	指标	3. 2135, 3. 2138
		指标值	3. 2139

指导	3. 2000	质量因数	3. 3431
指导委员会	3. 4290	质量因子样本	3. 3432
指南	3. 2001	质量因子值	3. 3433
指示性功能点计数	3. 2137	质量责任局部化	3. 2447
指引标识	3. 3986	质量政策	3. 3452
指针	3. 3042	质量子特性	3. 3458
制定	3. 1590	质量子因素	3. 3459
制定进度计划	3. 1360	致命错误	3. 1831
制定项目管理计划	3. 1359	智能视觉系统	3. 2260
制定项目章程	3. 1358	智能体	3. 2259
制定预算	3. 1356	中断服务例程	3. 2312
制品	3. 287	中断控制器	3. 2307
制品可信度	3. 289	中断潜伏期	3. 2308
制品证据	3. 288	中断请求	3. 2311
制品证据测量元	3. 290	中断掩码	3. 2309
制造	3. 2519	中断优先级	3. 2310
制造或购买分析	3. 2506	中继线	3. 4828
制造阶段	3. 2520	中间产品	3. 2285
质量	3. 3418	中间件	3. 2616
质量保证	3. 3420	中间软件产品	3. 2285
质量保证报告	3. 3405	中间软件产品要求	3. 2286
质量保证计划	3. 3422	中间系统产品	3. 2285
质量保证记录	3. 3423	中期功能点分析	3. 2281
质量保证评估报告	3. 3421	中小企业	3. 4013
质量报告	3. 3456	中央处理器	3. 632
质量测度	3. 3445	中央处理器 (CPU)	3. 633
质量测度的属性	3. 367	终端	3. 4586
质量测度元素	3. 3446	终结符	3. 4587
质量成本	3. 1063	终止构造	3. 4589
质量分析	3. 3419	终止交付	3. 4590
质量功能部署	3. 3434	终止提交	3. 4592
质量管理	3. 3437	终止行为	3. 4588
质量管理报告	3. 3444	终止证明	3. 4591
质量管理和控制工具	3. 3438	种子	3. 3885
质量管理计划	3. 3442	重定位	3. 3579
质量管理记录	3. 3443	重定位[代]码	3. 3582
质量管理纠正措施	3. 3439	重定位地址	3. 3580
质量管理评估报告	3. 3440	重定位汇编程序	3. 3581
质量管理指南	3. 3441	重定位器	3. 3587
质量核对单	3. 3427	重定位因素	3. 3585
质量计量	3. 3449	重定位装入程序	3. 3583
质量记录报告	3. 3455	重定位字典	3. 3584
质量控制	3. 1021, 3. 3428	重复寻址	3. 3595
质量控制措施	3. 3429	重构	3. 3517
质量模型	3. 3450	重量级进程	3. 2022
质量评定模块	3. 3454	重启点	3. 3682
质量评价	3. 3430	重启动	3. 3680
质量属性	3. 3424	重启动点	3. 3681
质量特性	3. 3425, 3. 3453	重入的	3. 3515
质量需求	3. 3457	重入点	3. 3516

重写	3. 2909	转换	3. 609, 3. 1038
重写属性	3. 2910	转换程序	3. 4814
重新激活	3. 3479	转换分析	3. 4805, 3. 4806
重演	3. 3597	转换功能	3. 1039
重用	3. 3700	转换系统	3. 648
重载	3. 2908	转换中心设计	3. 4805
周境覆盖	3. 972	转入	3. 3772
周境完备性	3. 971	转移	3. 4802, 3. 4810
周期	3. 1123	转移发生转移规则	3. 4811
周期窃取	3. 1124	桩[模块]	3. 4342
周期时间图	3. 1125	状态	3. 4253
周期性任务	3. 2979	状态报告	3. 4289
逐步细化	3. 4294	状态不变条件	3. 4257
逐页审阅	3. 2924	状态机	3. 4258
主程序	3. 2496	状态类	3. 4254
主程序员	3. 663	状态码	3. 4287
主程序员组	3. 664	状态名称	3. 4259
主动的方法	3. 3132	状态数据	3. 4255
主动互联	3. 73	状态图	3. 4256
主动性	3. 2208	状态转换	3. 4266
主干线	3. 4828	状态转移测试	3. 4261
主管	3. 4391	状态转移图	3. 4260
主管评估员	3. 779	追踪程序	3. 4783
主键	3. 3108	准备好的	3. 3489
主进度计划	3. 2497	准备时间	3. 3088
主控状态	3. 4390	准确[性]	3. 40
主实体类型	3. 3105	桌面出版	3. 1345
主授权标签	3. 3103	桌面检查	3. 1344
主授权模式	3. 3104	资本支出	3. 601
主题	3. 4706, 3. 4765	资产	3. 323
主题领域	3. 4352	资产保护	3. 331
主题域	3. 4353	资产范围界定	3. 332
主题域职能	3. 4354	资产管理	3. 326
主体	3. 4351	资产管理计划	3. 327
主信息结构	3. 3106	资产管理体系	3. 328
主要数据	3. 3102	资产建议书	3. 330
主要意图	3. 3107	资产库	3. 324
主要用户	3. 3109	资产类型	3. 334
注册标识符	3. 3539	资产寿命	3. 325
注册表	3. 3540	资产系统	3. 333
注解	3. 185	资产组合	3. 329
注释	3. 741	资格人员	3. 3415
注销	3. 2457	资格认证	3. 3406
专家系统	3. 1759	资格认证方案	3. 3412
专家系统 (ES)	3. 1760	资格认证机构	3. 3408
专业工程	3. 4192	资料图	3. 1181
专业化	3. 4190	资源	3. 3657
专业人员	3. 3066	资源池	3. 3666
专用集成电路	3. 231	资源分解结构	3. 3659
转出	3. 3773	资源分配	3. 3658

资源管理	3. 3663	自动验证工具	3. 391
资源监控任务	3. 3665	自动验证系统	3. 390
资源经理	3. 3664	自反的	3. 3534
资源利用率	3. 3668	自反祖先（一个类）	3. 3535
资源利用率测量	3. 3670	自含[带]文档的	3. 3894
资源利用率度量	3. 3669	自描述性	3. 3893
资源平衡	3. 3662	自然单位	3. 2730
资源平滑	3. 3667	自上而下的测试	3. 4764
资源日历	3. 3660	自上而下的分解	3. 4763
资源有限的时间表	3. 3671	自上而下的设计	3. 4763
资源直方图	3. 3661	自下而上设计	3. 517
子	3. 665	自相对地址	3. 3897
子包	3. 4361	自学习系统	3. 3895
子标签	3. 669	自由浮动时间	3. 1923
子程序	3. 4364	自由依赖关系	3. 1434
子程序跟踪	3. 4368	自主/智能系统	3. 396
子活动	3. 4344	自主/智能系统创建者	3. 2
子活动状态	3. 4345	自主系统	3. 395
子机[器]状态	3. 4359	自主性	3. 397
子集选取	3. 4372	自主增长	3. 394
子阶段	3. 4363	自助供应	3. 3896
子类	3. 4346	自组装产品	3. 3891
子类簇	3. 4347, 3. 4348	字	3. 5119
子类型	3. 4377	字段	3. 1178
子类职能	3. 4349	字符	3. 652
子例程	3. 4366	字符集	3. 653
子例程跟踪	3. 4367	字符类型	3. 654
子路径	3. 4362	字节	3. 575
子实体	3. 668	字面值	3. 2433
子图	3. 667	字母数字混合	3. 161
子图框	3. 666	综合	3. 4445
子网络	3. 4360	棕地系统工程	3. 539
子系统	3. 4376	踪迹	3. 4776
子项目	3. 4365	总簇	3. 4769
子域	3. 4350	总结活动	3. 4381
子状态	3. 4373	总结性评价	3. 4382
自包含	3. 3892	总揽测试	3. 479
自变量, 变元, 变元的值, 自变量的值	3. 278	总体责任	3. 137
自变量进度	3. 3824	纵向微指令	3. 5053
自底向上	3. 516	走查	3. 5092
自顶向下的	3. 4761	阻塞性缺陷	3. 504
自顶向下设计	3. 4762	组	3. 1998
自动测试生成器	3. 389	组成配置项	3. 947
自动的	3. 392	组合	3. 303, 3. 818
自动构建与回归测试工具	3. 4748	组合[类]	3. 809
自动化	3. 393	组合测度	3. 813
自动化系统流程辅助流程辅助软件流程辅助系统 流程自动化流程自动化软件流程	3. 388	组合测试	3. 735
自动设计工具	3. 387	组合测试方法	3. 736
自动数据处理	3. 1187	组合键	3. 811
		组合聚合	3. 810

组合强度	3. 734	作业成本核算	3. 87
组合状态	3. 815	作业功能	3. 2354
组件集成测试	3. 805	作业控制语言	3. 2353
组件内容管理系统	3. 802	作业流	3. 2355
组件实现	3. 804	作战室	3. 5093
组块【动】	3. 501	作者检查	3. 380
组织裁剪战略	3. 2866	做法	3. 3065
组织测试规程	3. 2881		
组织成熟度	3. 2870		
组织方针、程序和资产	3. 2872		
组织分解结构	3. 2867		
组织风险	3. 2880		
组织概况	3. 2878		
组织管理	3. 2868		
组织管理概况	3. 2869		
组织过程成熟度	3. 2875		
组织过程绩效测量数据	3. 2876		
组织过程绩效度量需要	3. 2877		
组织过程资产	3. 2874		
组织基础设施	3. 2861		
组织基础设施需求	3. 2862		
组织级测试策略	3. 2884		
组织级测试方针	3. 4661		
组织级测试规格说明	3. 2883		
组织级测试过程	3. 2882		
组织级项目管理成熟度	3. 2879		
组织计划	3. 2871		
组织经验教训	3. 2863		
组织战略规划	3. 2865		
组织政策	3. 2873		
组织组合方向和约束	3. 2864		
祖先框	3. 181		
祖先类(类的)	3. 180		
祖先图表	3. 182		
最低期望收益率	3. 2627		
最低吸引力收益率	3. 2628		
最低限度任务	3. 2630		
最后责任时刻	3. 2383		
最佳可用技术	3. 452		
最小可行产品	3. 2631		
最小延迟编程	3. 2629		
最小最大后悔值准则	3. 2626		
最早开始日期	3. 1546		
最早完成日期	3. 1545		
最终母版	3. 1266		
最终软件库	3. 1267		
最终用户	3. 1596		
最终用户代表	3. 1597		
左移	3. 2393		
作业	3. 2352		

英文对应词索引

A

(asynchronous task)	3. 196
A/B testing	3. 1
A/IS	3. 396
A/IS creator	3. 2
Aa	3. 42
ABC	3. 87
ability	3. 3
abnormal end	3. 4
abort	3. 5
absent variant	3. 6
absolute address explicit address specific address	3. 7
absolute assembler	3. 8
absolute code specific code	3. 9
absolute instruction	3. 10
absolute loader	3. 11
absolute machine code	3. 12
absolute time	3. 13
abstract behaviour	3. 14
abstract class	3. 15
abstract data type	3. 16
abstract design	3. 17
abstract machine	3. 18
abstraction	3. 19
AC, ...	3. 90
acceptability	3. 20
acceptability criteria	3. 21
acceptable	3. 22
acceptance	3. 23
acceptance criteria	3. 24
acceptance test-driven development	3. 25
acceptance testing	3. 26
accepted deliverables	3. 27
accepted system or system element	3. 28
access	3. 29
access facility	3. 30
access method	3. 31
access routine	3. 32
access transparency	3. 33
access-control mechanism	3. 34

accessibility	3. 35
accessibility testing	3. 36
accessiblity	3. 37
accident	3. 38
accountability	3. 39
accuracy	3. 40
accuracy of measurement	3. 41
achieved availability	3. 42
ACIA	3. 356
acknowledged SoS	3. 43
acknowledged system of systems	3. 43
acquire resources	3. 44
Acquired system	3. 45
acquirer	3. 46
acquisition	3. 47
acquisition agreement	3. 48
acquisition cost	3. 1058
acquisition logistics	3. 49
Acquisition need	3. 50
Acquisition payment	3. 51
Acquisition record	3. 52
Acquisition reply	3. 53
Acquisition report	3. 54
acquisition strategy	3. 55
action	3. 56
action entry	3. 57
action item	3. 58
action of interest	3. 59
action proposal	3. 60
action sequence	3. 61
action signature	3. 62
action state	3. 63
action stub	3. 64
activation	3. 65
activation constraint	3. 66
activation function	3. 67
activation value	3. 68
active area	3. 69
active class	3. 70
active enterprise object	3. 71
active file	3. 72
active interconnection	3. 73

active object	3. 74
active redundancy	3. 75
active text	3. 76
active white space	3. 77
activity	3. 78
activity code	3. 79
activity cost estimates	3. 80
Activity diagrams	3. 81
activity duration estimate	3. 82
activity group	3. 83
activity identifier	3. 84
activity list	3. 85
activity network diagram	3. 3355
activity type	3. 86
activity-based costing	3. 87
activity-on-node	3. 3068
activity-oriented WBS	3. 88
actor	3. 89
actual cost of work performed	3. 90
actual cost, ...	3. 90
actual depreciation	3. 91
actual duration	3. 92
actual parameter	3. 93
actual results	3. 94
ACWP	3. 90
AD	3. 266
AD element	3. 263
ad hoc reviewing	3. 95
adaptability	3. 96
adaptation data	3. 97
adaptation parameter	3. 98
adapter	3. 99
adaptive approach	3. 100
adaptive life cycle	3. 101
adaptive maintenance	3. 102
adaptive strategy	3. 103
adaptive support	3. 104
added source statements	3. 105
additional quality planning tools	3. 106
additive maintenance	3. 107
additive support	3. 108
additive weighting	3. 109

address	3. 110
address field	3. 111
address format	3. 112
address modification	3. 113
address part	3. 111
address space	3. 114
addressing exception	3. 115
addressing mode	3. 116
addressing range	3. 117
ADF	3. 264
adjusted size	3. 118
adjusting leads and lags	3. 119
ADL	3. 265
ADM	3. 274
adoption process	3. 120
ADP	3. 1187
advanced profile	3. 121
adversarial attack	3. 122
adversarial example	3. 123
adversarial testing	3. 124
adverse consequence	3. 125
AE	3. 188, 3. 268
AE(i)	3. 187
affect	3. 126
affective computing	3. 127
affective relationship	3. 128
affective rights	3. 129
afferent	3. 130
affinity diagram	3. 131
affinity grouping	3. 132
agency	3. 133
agent	3. 134
aggregate	3. 135
aggregate [class]	3. 136
aggregate responsibility	3. 137
aggregated resource utilization	3. 138
aggregation	3. 139
aggregation method	3. 140
agile	3. 141
agile concept	3. 142
agile development	3. 143
agile environment	3. 144

agile maturity	3. 145
agile team	3. 146
agile team lead	3. 147
agreement	3. 148
AHP	3. 178
AI	3. 291
AI effect	3. 292
AI quality metamodel	3. 293
AI-based system	3. 295
AIS	3. 294
algebraic language	3. 149
algorithm	3. 150
algorithm analysis	3. 151
algorithmic drift	3. 152
algorithmic language	3. 153
alias	3. 154
allocated baseline	3. 155
allocated configuration identification	3. 156
allocated requirement	3. 157
allocation	3. 158
allocation of an entitlement	3. 159
alpha testing	3. 160
alphanumeric	3. 161
alternate flow alternate path	3. 162
alternate key	3. 163
alternative analysis	3. 164
Alternative solution classes	3. 165
alternatives generation	3. 166
AM	3. 326
ambiguity	3. 167
analog	3. 168
analogous estimating	3. 169
analogue computer	3. 170
analogy	3. 171
analysability	3. 172
analysis	3. 173
analysis model	3. 174
analysis phase	3. 175
Analysis situations	3. 176
analyst	3. 177
analytic hierarchy process, ...	3. 178
analytical techniques	3. 179

ancestor (of a class)	3. 180
ancestral box	3. 181
ancestral diagram	3. 182
anchor point	3. 183
annotated topic list	3. 184
annotation	3. 185
announcement	3. 186
annual equivalent	3. 187
annual equivalent AE(i)	3. 188
annual percentage rate	3. 189
annuity	3. 190
anomaly	3. 191
anticipatory buffering	3. 192
anticipatory paging	3. 193
Ao, ...	3. 194
AON	3. 3068
aperiodic task	3. 196
aperiodic task asynchronous task	3. 195
API	3. 223
API management platform	3. 222
applicability to a functional domain	3. 197
applicant	3. 198
application	3. 199, 3. 2573
application administration function	3. 200
application architecture	3. 201
application area	3. 202
application asset	3. 203
application assets in requirements	3. 204
application boundary	3. 205
application component	3. 206
application configuration	3. 207
application design	3. 208
application engineering	3. 209
application engineering process	3. 210
application frameworks	3. 211
application function point count	3. 212
application functional size	3. 213
application generator	3. 214
application interface	3. 215
application management	3. 216
application management organization	3. 217
application object	3. 218

application portfolio	3. 219
application problem	3. 220
application program interface (API) management platform	3. 221
application program interface management platform, ...	3. 222
Application programming interface	3. 223
application realization	3. 224
application requirements analysis	3. 225
application requirements elicitation	3. 226
application requirements management	3. 227
application requirements specification	3. 228
application requirements verification and validation	3. 229
application software	3. 230
application specific integrated circuit, ...	3. 231
application test asset	3. 232
application testing	3. 233
application variability model	3. 234
application-oriented language	3. 235
application-specific component	3. 236
application-specific requirements	3. 237
applying leads and lags	3. 238
apportioned effort	3. 239
appraisal findings	3. 240
appraisal participants	3. 241
appraisal team leader	3. 242
appropriateness recognizability	3. 243
approval	3. 244
approval authority	3. 245
approved change request	3. 246
approved change requests review	3. 247
approved modification	3. 248
APR	3. 189
arc	3. 249
arc annotation	3. 250
architect	3. 251
architecting	3. 252
architectural description	3. 266
architectural design	3. 253
architectural design phase	3. 254
architectural design review	3. 255
architectural structure	3. 256
architectural style	3. 257
architectural texture	3. 4705

architecture	3. 258
architecture collection	3. 259
Architecture definition	3. 260
Architecture definition record	3. 261
Architecture definition strategy	3. 262
architecture description	3. 266
architecture description element, ...	3. 263
architecture description framework, ...	3. 264
architecture description language, ...	3. 265
architecture entity	3. 267
architecture evaluation	3. 268
architecture evaluation framework	3. 269
architecture framework	3. 270
Architecture traceability	3. 271
architecture view	3. 272
architecture viewpoint	3. 273
architecture-driven modernization	3. 274
archival page	3. 275
archive	3. 276
archiving	3. 277
argument	3. 278
arity	3. 279
arranging	3. 280
array	3. 281
arrow	3. 282
arrow label	3. 283
arrow meaning	3. 284
arrow refrence	3. 2064
arrow role	3. 285
arrow segment	3. 286
artifact	3. 287
artifact evidence	3. 288
artifact trustworthiness	3. 289
artifact-evidence element	3. 290
artificial intelligence	3. 291
artificial intelligence effect	3. 292
artificial intelligence quality metamodel	3. 293
artificial intelligence system, ...	3. 294
artificial intelligence-based system	3. 295
artificial language	3. 296, 3. 1901
ASIC	3. 231
ask	3. 297

aspect	3. 298
assemble	3. 299
assemble-and-go	3. 300
assembled origin	3. 301
assembler	3. 302
assembler language, ...	3. 306
assembly	3. 303
assembly code assembler code	3. 304
assembly language assembler language low-level language second-generation language..	3. 305
assembly language, ...	3. 306
assertion	3. 307
assessed capability	3. 308
assessment	3. 309
assessment body	3. 310
assessment constraint	3. 311
assessment indicator	3. 312
assessment input	3. 313
assessment instrument	3. 314
assessment output	3. 315
assessment participant	3. 316
assessment purpose	3. 317
assessment record	3. 318
assessment scope	3. 319
assessment sponsor	3. 320
assessment team	3. 321
assessor	3. 322
asset	3. 323
asset base	3. 324
asset life	3. 325
asset management	3. 326
asset management plan	3. 327
asset management system	3. 328
asset portfolio	3. 329
asset proposal	3. 330
asset protection	3. 331
asset scoping	3. 332
asset system	3. 333
asset type	3. 334
assignment	3. 335
assignment statement	3. 336
assisted process assisted software process	3. 388
assisted systems process	3. 388

assistive technology	3. 337
association	3. 338
association class	3. 339
association end	3. 340
association management facility	3. 341
associative class	3. 342
associative entity	3. 343
associative entity type	3. 344
associative literal	3. 345
assumption	3. 346
assumption and constraint analysis	3. 347
assumption log	3. 348
assumptions analysis	3. 349
assurance argument	3. 350
assurance case report	3. 351
assurance claim	3. 352
assurance information	3. 353
assurance objective	3. 354
asynchronous	3. 355
asynchronous communication interface adapter	3. 356
asynchronous I/O device	3. 357
asynchronous I/O device interface task	3. 358
asynchronous message communication	3. 359
asynchronous message communication, ...	3. 360
AT	3. 374
ATDD	3. 25
ATL	3. 184
atomic model	3. 361
atomic type, ...	3. 362
atomicity	3. 363
attached process	3. 364
attack	3. 365
attribute	3. 366
attribute for quality measure	3. 367
attribute name	3. 368
attribute sampling	3. 369
attributed relationship	3. 370
attributive entity type	3. 371
audience	3. 372
audit	3. 373
audit output	3. 3712, 3. 3713
audit team	3. 374

auditability	3. 375
auditee	3. 376
auditor	3. 377
augmented reality system	3. 378
authenticity	3. 379
author check	3. 380
authoring environment	3. 381
authoring language	3. 382
authoring system	3. 383
authorization	3. 384
authorized individual	3. 385
automate	3. 386
automated design tool	3. 387
automated process automated software process	3. 388
automated systems process	3. 388
automated test generator	3. 389
automated verification system	3. 390
automated verification tools	3. 391
automatic	3. 392
automatic data processing (ADP)	3. 1187
automation	3. 393
autonomous growth	3. 394
autonomous system	3. 395
autonomous/intelligent system	3. 396
autonomy	3. 397
autonomy-based improvement	3. 398
auxiliary class	3. 399
availability	3. 400
availability model	3. 401
available interaction	3. 402
awareness requirement	3. 403

B

BAC	3. 544
back matter	3. 404
background	3. 405
background processing	3. 406
backlog	3. 407
backlog refinement	3. 408
backout	3. 409
back-to-back testing	3. 410
backup	3. 411
backup and recovery testing	3. 412

backup programmer	3. 413
Backus–Naur Form	3. 414
backward execution	3. 415, 3. 3708
backward propagation	3. 416
backward recovery	3. 417
bag	3. 418, 3. 2712
ball grid array	3. 419
base address	3. 420
base choice	3. 421
base class	3. 422
base functional component	3. 423
base functional component (BFC) functional service	3. 424
base functional component type	3. 425
base functional component type (BFC type)	3. 426
base measure	3. 427
base practice	3. 428
base standard	3. 429
base value	3. 430
baseline	3. 431
baseline configuration management	3. 432
baseline data	3. 433
baseline design	3. 434
baseline document	3. 435
baseline function point count	3. 436
baseline management	3. 437
baseline price for bid evaluation	3. 438
basic assumption	3. 439
basic engineering object	3. 440
basic flow	3. 441
basic flow basic path happy day scenario	3. 442
basic interworking facility	3. 443
basic maturity level	3. 444
basic process set	3. 445
basic profile	3. 446
basic symbol	3. 447
basis of estimates	3. 448
basis set	3. 449
BAT	3. 452
batch	3. 450
bathtub curve	3. 451
Bayes' rule	3. 453
BCWS	3. 3025

BDD	3. 460
begin-end block	3. 454
behavior	3. 455
behavior model	3. 456
behavioral compatibility	3. 457
behavioral feature	3. 458
behavioral model aspect	3. 459
behaviour-driven development	3. 460
benchmark	3. 461
benchmark suite	3. 462
benchmarking	3. 463
benchmarking analyst	3. 464
benchmarking experience base	3. 465
benchmarking method	3. 466
benchmarking report	3. 467
benchmarking repository	3. 468
benchmarking user	3. 469
beneficiary	3. 1117
benefit	3. 470
benefit cost analysis	3. 471
best available technique	3. 452
beta test	3. 472
BFC	3. 423, 3. 424
BFC class	3. 473
BFC type	3. 426, 3. 474
BGA	3. 419
bias	3. 475
bid price	3. 476
bidder	3. 477
bidirectional traceability	3. 478
big-bang testing	3. 479
bill of materials	3. 480
bill of materials (BOM)	3. 481
bill-of-features	3. 482
bill-of-features portfolio	3. 483
binary association	3. 484
binary digit	3. 485
bind	3. 486
binder	3. 487
binding	3. 488
binding behavior	3. 489
binding endpoint identifier	3. 490

binding object	3. 491
binding precondition	3. 492
binding time	3. 493
binding time decision	3. 494
binding time of variability	3. 495
BIT	3. 485, 3. 496
bit steering	3. 497
bit steering immediate control	3. 498
black box	3. 499
black-box testing	3. 500, 3. 4204
block	3. 501
block allocation	3. 502, 3. 2926
block diagram	3. 503
blocking bug	3. 504
blocking factor	3. 505
block-structured language	3. 506
body metadata	3. 507
body of knowledge	3. 508
BOK	3. 508
BOM	3. 480, 3. 481
Boolean	3. 509
Boolean expression	3. 510
Boolean signature	3. 511
boot	3. 512
boot mode	3. 513
bootstrap	3. 514
bootstrap loader	3. 515
bottom-up	3. 516
bottom-up design	3. 517
boundary	3. 518
boundary arrow	3. 519
boundary value	3. 520
boundary value analysis	3. 521
box	3. 522
box detail reference	3. 523
box diagram	3. 524
box diagram Chapin chart Nassi-Shneiderman chart	3. 525
box ICOM code	3. 526
box name	3. 527
box number	3. 528
BP	3. 428
brainstorming	3. 529

branch	3. 530
branch condition combination testing	3. 531
branch condition testing	3. 532
branch testing	3. 533
branching	3. 534
branchpoint	3. 535
breadcrumb trail	3. 536
break-even analysis	3. 537
breakpoint	3. 538
brownfield system engineering	3. 539
browser	3. 540
bubble chart	3. 541
buddy check	3. 542
budget	3. 543
budget at completion	3. 544
budget at completion (BAC)	3. 545
budget price	3. 546
budgeted cost of work scheduled	3. 3025
budgeting	3. 547
buffer	3. 548
bug	3. 549, 3. 1840
bug seeding	3. 550, 3. 1678, 3. 1837
build	3. 551
build pipeline	3. 1368
build process	3. 552
building block	3. 553
Built-in RAM	3. 554
built-in random access memory	3. 554
built-in random access memory (RAM)	3. 555
built-in read only memory built-in ROM	3. 556
burndown	3. 557
burndown chart, burn chart	3. 558
burndown rate	3. 559
burn-in period	3. 560, 3. 1547
burnup	3. 561
bus	3. 562
business case	3. 563
business critical product	3. 564
business impact analysis	3. 565
business information management	3. 566
business model canvas	3. 567
business objective	3. 568

business process	3. 569
Business requirements	3. 570
Business Requirements Specification	3. 571
business value	3. 572
busy	3. 573
busy time	3. 574
byte	3. 575

C

c use	3. 819
CA	3. 856
cache	3. 576
CAD	3. 857
cadence	3. 577
calendar unit	3. 578
call	3. 579
call by name	3. 580
call by reference	3. 581
call by reference call by address call by location	3. 582
call by value	3. 583
call graph	3. 584
call graph call tree, tier chart	3. 585
call list	3. 586
call reference	3. 587
call trace	3. 588, 3. 4368
called diagram	3. 589
calling box	3. 590
calling sequence	3. 591
CAN	3. 1034
candidate FSM method	3. 592
candidate key	3. 593
capability dimension	3. 594
capability maturity model	3. 595
capability requirement	3. 596
capable process	3. 597
capacity	3. 598
capacity place	3. 599
capacity testing	3. 600
capital expenditure	3. 601
capsule	3. 602
capsule manager	3. 603
cardinality	3. 604
cardinality constraint	3. 605

cardinality of a multiset	3. 2711
carrier	3. 606
case	3. 607, 3. 858
case multiple exclusive selective construct	3. 608
CASE needs	3. 859
CASE tool	3. 860
cast	3. 609
catastrophic failure	3. 610
categorization	3. 611
categorization scheme	3. 612
categorization space	3. 613
category cluster	3. 4348
category discriminator	3. 1437
category entity	3. 614
causal analysis	3. 615
cause and effect diagram	3. 616
cause-effect graphing	3. 617
caution	3. 618
CBSS	3. 861
CCB	3. 638, 3. 906, 3. 907
CCMS	3. 802
CCR	3. 1090
CD	3. 985
CDIF clear text encoding	3. 619
CDIF exporter	3. 620
CDIF family of standards	3. 621
CDIF graphical notation	3. 622
CDIF identifier	3. 623
CDIF importer	3. 624
CDIF metaidentifier	3. 625
CDIF meta-metamodel	3. 626
CDIF semantic metamodel	3. 627
CDIF transfer	3. 628
CDIF transfer file	3. 629
CDIF transfer format	3. 630
CDIF transfer syntax and encoding	3. 631
CDR	3. 1091, 3. 1092
CD-ROM	3. 770
central processing unit	3. 632
central processing unit (CPU)	3. 633
CFD	3. 1112
chain	3. 634

chain type	3. 635
change	3. 636
change control	3. 637
change control board (CCB)	3. 638
change control plan	3. 639
change control procedure	3. 640
change dump	3. 641
change management	3. 642
change package	3. 643
change project function point count	3. 644
change record	3. 645
change request	3. 646
change set	3. 647
changeover system	3. 648
change-over system	3. 649
channel capacity	3. 650
Chapin chart	3. 651
character	3. 652
character set	3. 653
character type	3. 654
characteristic	3. 655
characteristic entity	3. 656
characteristic of FUR	3. 657
checklist analysis	3. 658
checklist-based reviewing	3. 659
checkout	3. 660
checkpoint	3. 661
checksheets	3. 662
chief programmer	3. 663
chief programmer team	3. 664
child	3. 665
child box	3. 666
child diagram	3. 667
child entity	3. 668
child tag	3. 669
chip	3. 2246
CI	3. 910, 3. 912, 3. 988
clabject	3. 670
claim	3. 671
class	3. 672
class data	3. 2675
class diagram	3. 673

class hierarchy	3. 674
classification	3. 675
classification axis	3. 676
classification tree	3. 677
classification tree method	3. 678
classifier	3. 679
class-level attribute	3. 680
class-level operation	3. 681
class-level responsibility	3. 682
clear	3. 683
clear text file encoding	3. 684
client	3. 685
client object	3. 686
clock pulse generator	3. 687
cloning	3. 688
closed loop	3. 689
closed subroutine	3. 690
closed term	3. 691
closed term/ground term	3. 692
closed-box testing	3. 4204
closing process group	3. 693
cloud computing	3. 694
cloud interoperability	3. 695
cloud service	3. 696
cloud service customer	3. 697
cloud service provider	3. 698
cluster	3. 699
cluster checkpoint	3. 700
cluster manager	3. 701
cluster template	3. 702
clustering	3. 703
CM	3. 914
CM service	3. 704
CM service configuration management service	3. 704
CM tool	3. 705
CM tool configuration management tool	3. 705
CMDB	3. 916, 3. 917
CMP	3. 919, 3. 965
CMS	3. 966
code	3. 706
code audit	3. 707
code breakpoint control breakpoint	3. 708

code data	3. 709
code freeze	3. 710
code generator	3. 711
code inspection	3. 712
code review	3. 713
code trace	3. 714, 3. 1740
code tuning	3. 715
code walk-through	3. 716
coding	3. 717
coerce	3. 718
coexistence	3. 719
co-existence	3. 719
cognitive walkthrough	3. 720
cohesion	3. 721
cohesion module strength	3. 722
coincidental cohesion	3. 723
collaboration	3. 724
collaboration diagram	3. 725
collaborative SoS	3. 726
collaborative system of systems	3. 726
collapse	3. 727
collection cardinality	3. 728
collection class	3. 729
collection-valued	3. 730
collection-valued class	3. 731
collection-valued property	3. 732
collection-valued property collection property	3. 733
combinatorial strength	3. 734
combinatorial testing	3. 735
combinatorial testing method	3. 736
comfort	3. 737
command	3. 738
command language	3. 739
command-driven	3. 740
comment	3. 741
commercial-off-the-shelf	3. 742
commercial-off-the-shelf software product	3. 743
commissioning	3. 744
commit	3. 745
commit message	3. 746
commit war	3. 747
commit window	3. 748

commitment	3. 749
committer	3. 750
common ancestor constraint	3. 751
common cause (of a defect)	3. 752
common cause random cause	3. 753
common configuration item	3. 754
common coupling	3. 755
common data	3. 756
common features	3. 757
common storage	3. 758
commonality	3. 759
commonality index	3. 760
common-environment coupling	3. 761
communication constraints	3. 762
communication interface	3. 763
communication management	3. 764
communicational cohesion	3. 765
communications domain	3. 766
communications planning	3. 767
community	3. 768
community object	3. 769
compact disc read only memory	3. 770
compaction	3. 771
comparator	3. 772
compatibility	3. 773
compatibility testing	3. 774
compatible assessment model	3. 775
compensatory decision technique	3. 776
compensatory model	3. 777
competency	3. 778
competent assessor	3. 779
compile	3. 780
compile time	3. 781
compile-and-go	3. 782
compiler	3. 783
compiler code	3. 784
compiler compiler	3. 785
compiler directive source statement	3. 786
compiler generator	3. 787
complete cluster	3. 4770
complete ICOM code	3. 788
complete procedure	3. 789

complete table	3. 790
completeness	3. 791
completion code	3. 792
completion criteria	3. 793
completion time theorem	3. 794
complex products	3. 795
complex programmable logic device	3. 796
complex programmable logic device (CPLD)	3. 797
complexity	3. 798
complexity matrix	3. 799
compliance	3. 800
component	3. 801
component content management system	3. 802
component diagram	3. 803
component implementation	3. 804
component integration test	3. 805
component standard	3. 806
component testing	3. 807
composed model	3. 808
composite [class]	3. 809
composite aggregation	3. 810
composite key	3. 811
composite keyword	3. 812
composite measure	3. 813
composite object	3. 814
composite state	3. 815
composite task	3. 816
composite type	3. 817
composition	3. 818
computation data use	3. 819
computational component	3. 820
computational container	3. 821
computational factory	3. 822
computational interface template	3. 823
computational object template	3. 824
computational viewpoint	3. 825
computer	3. 826
computer centre	3. 827
computer crime	3. 828
computer data	3. 829
computer generation	3. 830
computer graphics	3. 831

computer instruction	3. 832
computer language	3. 833
computer network	3. 834
computer performance evaluation	3. 835
computer program	3. 836
computer program abstract	3. 837
computer program annotation	3. 838
computer program certification	3. 839
computer program component (CPC)	3. 840
computer program configuration identification	3. 841
computer program configuration item (CPCD)	3. 842
computer program development plan	3. 843
computer program validation	3. 844
computer program verification	3. 845
computer resource	3. 846
computer resource allocation	3. 847
computer science	3. 848
computer security	3. 849
computer software component	3. 850
computer software component (CSC)	3. 851
computer software configuration item	3. 853
computer software configuration item (CSCI) software configuration item (SWCI)	3. 852
computer system computing system	3. 854
computer word	3. 855
computer-aided	3. 856
computer-aided design	3. 857
computer-aided software engineering	3. 858
computer-aided software engineering needs	3. 859
computer-aided software engineering tool	3. 860
computer-based software system	3. 861
computerization	3. 862
computing centre	3. 863
computing device	3. 864
computation data use	3. 865
concept analysis	3. 866
concept drift	3. 867
Concept of operations	3. 868
concept of operations document	3. 869
concept phase	3. 870
conceptual data model	3. 871
conceptual information	3. 872
conceptual model	3. 873

conceptual system design	3. 874
concern	3. 875
conciseness	3. 876
concrete class	3. 877
concurrency	3. 878
concurrent	3. 879
concurrent communication diagram	3. 880
concurrent elaboration	3. 881
concurrent enabling	3. 882
concurrent enabling (of transition modes)	3. 883
concurrent processes	3. 884
concurrent substate	3. 885
concurrent task architecture	3. 886
condition	3. 887
condition code	3. 888
condition entry	3. 889
condition stub	3. 890
conditional branch	3. 891
conditional control structure	3. 892
conditional information	3. 893
conditional jump	3. 894
conditional process	3. 895
conditional task	3. 896
conditional text	3. 897
Conduct Procurements	3. 898
cone of uncertainty	3. 899
confidentiality	3. 900
configurability	3. 901
configuration	3. 902
configuration audit	3. 903
Configuration baseline	3. 904
configuration control	3. 905
configuration control board	3. 906
configuration control board (CCB)	3. 907
configuration identification	3. 908
configuration index	3. 909
configuration item	3. 910
configuration item development record	3. 911
configuration item(CI)	3. 912
Configuration management	3. 913
configuration management (CM)	3. 914
configuration management authority	3. 915

configuration management database	3. 916
configuration management database (CMDB)	3. 917
configuration management library system	3. 918
configuration management plan (CMP)	3. 919
Configuration management record	3. 920
Configuration management report	3. 921
Configuration management strategy	3. 922
configuration management system	3. 923
configuration parameter	3. 924
configuration status accounting (CSA)	3. 925
configuration unit	3. 926
confinement	3. 927
confirmation bias	3. 928
confirmation testing	3. 3686
conflict	3. 929
conflict management	3. 930
conformance	3. 931
conformance point	3. 932
conformance work	3. 933
conformity assessment	3. 934
conformity evaluation	3. 935
conformity evaluation report	3. 936
confusion matrix	3. 937
connection	3. 938
connectivity	3. 939
ConOps	3. 868
ConOps document	3. 869
consecutive	3. 940
consistency	3. 941
consistent	3. 942
consistent state	3. 943
consolidation of an entitlement	3. 944
constant	3. 945
constant-failure period	3. 946
constituent configuration item	3. 947
constituent system	3. 948
constraint	3. 949
constraints dependency	3. 950
construct	3. 951
constructed capability	3. 952
construction	3. 953
consumer object	3. 954

consumer product	3. 955
consumer software package	3. 956
container	3. 957
container interface	3. 958
containment hierarchy	3. 959
content	3. 960
content chunk	3. 961
content consistency	3. 962
content coupling	3. 963
content management	3. 964
content management platform	3. 965
content management system	3. 966
content object	3. 967
content type	3. 968
content unit	3. 969
context	3. 970
context completeness	3. 971
context coverage	3. 972
context of use	3. 973
context-sensitive help	3. 974
contextual factor	3. 975
contextual schema	3. 976
contiguous allocation	3. 977
contingency	3. 978
contingency allowance	3. 3647
contingency factor	3. 979
contingency plan	3. 980
contingency reserve	3. 981
contingent threshold	3. 982
continual process improvement	3. 983
continuity	3. 984
continuous delivery	3. 985
continuous deployment	3. 986
continuous improvement	3. 987
continuous integration	3. 988
continuous iteration	3. 989
continuous representation	3. 990
continuous risk management	3. 991
contract administration	3. 992
contract change control system	3. 993
contract management plan	3. 994
contract terms and conditions	3. 995

contract work breakdown structure	3. 996
contract work breakdown structure (CWBS)	3. 997
contractor	3. 4399
contractual context	3. 998
contractual requirement	3. 999
contractually required audit	3. 1000
contravariance	3. 1001
control	3. 1002
control account	3. 1003
control breakpoint	3. 1004
control chart	3. 1005
control clustering	3. 1006
control communications	3. 1007
Control Costs	3. 1008
control coupling	3. 1009
control data	3. 1010
control field	3. 1011
control flow	3. 1012, 3. 1884
control flow diagram	3. 1013
control flow sub-path	3. 1014
control flow trace	3. 1015, 3. 1740
control Information	3. 1016
control language	3. 1017
control loopback	3. 1018
control manager	3. 1019
control program	3. 1020, 3. 1742, 3. 4391
control quality	3. 1021
control resources	3. 1022
control risks	3. 1023, 3. 1024
Control Schedule	3. 1025
Control Scope	3. 1026
control stakeholder engagement	3. 1027
control statement	3. 1028
control store	3. 1029
control structure	3. 1030
control task	3. 1031
control variable	3. 1032
controller	3. 1033
controller area network	3. 1034
convention	3. 1035
conversational	3. 1036
conversational compiler	3. 1037, 3. 2121

conversion	3. 1038
conversion functionality	3. 1039
convertibility	3. 1040
coordinated interactions	3. 1041
copy	3. 1042
COQ	3. 1063
core	3. 1043
core asset	3. 1478
core report	3. 1044
core value	3. 1045
coroutine	3. 1046
corporate governance of IT	3. 1047
correctability	3. 1048
correction	3. 1049
corrective action	3. 1050
corrective maintenance	3. 1051
corrective support	3. 1052
correctness	3. 1053
correspondence	3. 1054
cost aggregation	3. 1055
cost avoidance	3. 1056
cost baseline	3. 1057
cost basis	3. 1058
cost constraint	3. 1059
cost function	3. 1060
cost management plan	3. 1061
cost measurement	3. 1062
cost of quality	3. 1063
cost performance baseline	3. 1064
cost performance index	3. 1065
cost plus award fee contract	3. 1066
cost plus fixed fee contract	3. 1067
cost plus incentive fee contract	3. 1068
cost reimbursable contract	3. 1072
cost variance	3. 1069
cost-benefit analysis	3. 1070
cost-plus-fee	3. 1071
cost-reimbursable contract	3. 1072
countable failure	3. 1073
counter	3. 1074
counting rule	3. 1075
counting scope	3. 1076

coupling	3. 1077
courtesy	3. 1078
coverage item	3. 1079, 3. 4608
CPAF	3. 1066
CPC	3. 840
CPCD	3. 842
CPF	3. 1071
CPFF	3. 1067
CPG	3. 687
CPI	3. 1065
CPIF	3. 1068
CPLD	3. 796, 3. 797
CPM	3. 1096
CPU	3. 632, 3. 633
crash	3. 1080
crashing	3. 1081
create WBS	3. 1082
create WBS (work breakdown structure)	3. 1083
creation	3. 1084
crisis	3. 1085
crisis management	3. 1086
criteria	3. 1087
critical	3. 1105
critical asset	3. 1088
critical chain method	3. 1089
critical computer resource	3. 1090
critical design review	3. 1091
critical design review (CDR)	3. 1092
critical information	3. 1093
critical item	3. 1094
critical path	3. 1095
critical path method	3. 1096
critical piece first	3. 1097
critical property	3. 1098
critical range	3. 1099
critical region	3. 1100
critical section	3. 1100
critical software	3. 1101
critical system	3. 1102
critical value	3. 1103
criticality	3. 1104
cross-assembler	3. 1106

cross-compiler	3. 1107
cross-reference generator	3. 1108
cross-reference list	3. 1109
cross-reference tool	3. 1110
cross-referencer	3. 1108
cryptographic hash	3. 1111
CS	3. 948
CSC	3. 697, 3. 850, 3. 851
CSCI	3. 853
CSP	3. 698
CTOS	3. 742
cumulative flow diagram	3. 1112
current extent	3. 1780, 3. 1781
curriculum standard	3. 1113
custom software	3. 1114
customer	3. 1115, 3. 1117
customer satisfaction	3. 1116
customizability	3. 1118
customization	3. 1119
customize	3. 1120
cut-off date	3. 1121
cutover	3. 1122
CV	3. 1069
CWBS	3. 996, 3. 997
cycle	3. 1123
cycle stealing	3. 1124
cycle time chart	3. 1125
cyclic search	3. 1126

D

daily standup	3. 1127, 3. 1128
daily stand-up	3. 1128
danger	3. 1129
dangerous condition	3. 1130
dark matter	3. 1131
Darwin Information Typing Architecture	3. 1132
dashboard	3. 1133
data	3. 1134
data abstraction	3. 1135
data action	3. 1136
data analysis	3. 1137
data analysis technique	3. 1138
data attribute	3. 1139

data bank	3. 1140
data breakpoint	3. 1141
data buffer register	3. 1142
data characteristic	3. 1143
Data Communications Equipment	3. 1144
data configuration	3. 1145
data corruption	3. 1146
data coupling	3. 1147
data date	3. 1148
data declaration source statement	3. 1149
data definition	3. 1152
data definition c-use pair	3. 1150
data definition p-use pair	3. 1151
data definition-use pair	3. 1153
data design	3. 1154
data dictionary	3. 1155
data drift	3. 1156
data element	3. 1158
data element type	3. 1157
data erasure	3. 1159
data exception	3. 1160
data exposure	3. 1161
data failure	3. 1162
data file	3. 1163
data flow	3. 1164
data flow diagram	3. 1165, 3. 1166, 3. 1201
data flow graph	3. 1165, 3. 1166
data flow testing	3. 1167
data flowchart	3. 1165, 3. 1166
data format	3. 1168
data function	3. 1169
data function type	3. 1170
data gathering and analysis methods	3. 1171
data gathering technique	3. 1172
data governance	3. 1173
data group	3. 1174
data group type	3. 1174
data information	3. 1175
data input sheet	3. 1176
data inventory	3. 1177
data item	3. 1158, 3. 1178
data leakage	3. 1179

data manipulation	3. 1180
data map	3. 1181
data medium	3. 1182
data model	3. 1183
data movement	3. 1184
data movement type	3. 1184
data privacy	3. 1185
data processing	3. 1186
data processing (DP)	3. 1187
data processing centre	3. 827
data protection	3. 1188
data protection impact assessment	3. 1189
data provider	3. 1190
data quality	3. 1191
data quality characteristic	3. 1192
data quality measure	3. 1193
data quality model	3. 1194
data record	3. 1195
data repository	3. 1196
data representation techniques	3. 1197
data sequencer	3. 1198
data store	3. 1199
data structure	3. 1200
data structure diagram	3. 1201
data structure-centered design	3. 1202
data structure-centred design	3. 1203
data subject	3. 1204
data system	3. 1205
data trace	3. 1206, 3. 5013
data transfer controller	3. 1207
data type	3. 1208
data use	3. 1209
data value	3. 1210
database	3. 1211
database design specification	3. 1212
database management system	3. 1213
database view	3. 1214
data-driven	3. 1215
data-flow trace	3. 1206, 3. 5013
DataItem	3. 1216
DataKey	3. 1217
DataPart	3. 1218

DataRole	3. 1219
data-sensitive fault	3. 1220
dataset shift	3. 1454
data-structure-oriented design	3. 1221
DataView	3. 1222
DBMS	3. 1213
DBR	3. 1142
DCE	3. 1144
DDOS attack	3. 1449
DDR SDRAM	3. 1502
DDR2	3. 1503
DDR2 SDRAM	3. 1223
DDR3	3. 1224
DDR4	3. 1225
deactivation	3. 1226
deadlock	3. 1227
deassembler	3. 1228
deblock	3. 1229
debugging	3. 1230
decision	3. 1231
decision criteria	3. 1232
decision gate	3. 1233
decision outcome	3. 1234
decision policy	3. 1235
decision rule	3. 1236
decision table	3. 1237
decision table testing	3. 1238
decision testing	3. 1239
decision tree	3. 1240
decision tree analysis	3. 1241
decision variable	3. 1242
decision-making technique	3. 1243
declaration	3. 1244
declarative language	3. 1245
decompile	3. 1246
decompiler	3. 1247
decomposer	3. 1248
decomposition	3. 1249
decomposition diagram	3. 1250
decoupling	3. 1251
deep learning	3. 1252
deep neural net	3. 1253

defect	3. 1254
defect density	3. 1255
defect prevention	3. 1256
defect removal	3. 3591
defect repair	3. 1257
defect root cause	3. 1258
defensive programming	3. 1259
Define Activities	3. 1260
Define Scope	3. 1261
defined process	3. 1262
defining model	3. 1263
definition of done	3. 1499
definition-use pair	3. 1264
definition-use path	3. 1265
definitive master version	3. 1266
definitive software library	3. 1267
degradation	3. 1509
degree of influence	3. 1268
de-identification	3. 1269
delegation	3. 1270
deleted source statement	3. 1271
delimiter	3. 1272
deliverable	3. 1273
deliverable product	3. 1274
delivered source statement	3. 1275
delivery	3. 1276
delivery performance domain	3. 1277
Delphi method	3. 1278
delta	3. 1279
demand paging	3. 1280
demodularization	3. 1281
demonstration	3. 1282
dependability	3. 1283
dependability measurement	3. 1284
dependability requirement	3. 1285
dependency determination	3. 1286
dependency export	3. 1287
dependent entity	3. 1288, 3. 1289
dependent state class	3. 1290
dependently	3. 1291
deployment	3. 1292
deployment diagram	3. 1293

deployment package	3. 1294
derivation of requirements	3. 1295
derived attribute	3. 1301, 3. 1302
derived class	3. 1296
derived data	3. 1297
derived element	3. 1298
derived measure	3. 1299
derived participant party	3. 1301, 3. 1302
derived property	3. 1301, 3. 1302
derived property.	3. 1300
derived requirement	3. 1303
derived type	3. 1304
description	3. 1305
description standard	3. 1306
descriptive model	3. 1307
design	3. 1308
design analyser	3. 1309
design analysis	3. 1310
design attribute	3. 1311
design characteristic	3. 1312
design concept	3. 1313
design concern	3. 1314
design constraint	3. 1315
design description design document design specification	3. 1316
design document	3. 1317
design element	3. 1318
design entity	3. 1319
design fault	3. 1320
design language	3. 1321
design level	3. 1322
design methodology	3. 1323
design of experiments	3. 1324
design overlay	3. 1325
design pattern	3. 1326
design phase	3. 1327
design rationale	3. 1328
design relationship	3. 1329
design requirement	3. 1330
design review	3. 1331
design specification	3. 1317
design standard	3. 1332
design strategy	3. 1333

design subject	3. 1334
design thinking	3. 1335
Design to cost	3. 1336
Design traceability	3. 1337
design unit	3. 1338
design view	3. 1339
design viewpoint	3. 1340
design-to-cost	3. 1341
desirable consequence	3. 1342, 3. 1343
desk checking	3. 1344
desktop publishing	3. 1345
destination address	3. 1346
destructive read	3. 1347
DET	3. 1157, 3. 1355
detailed design description	3. 1348, 3. 1349, 3. 1350
detailed design phase	3. 1351
detailed design review	3. 1352
detailed design specification	3. 1349, 3. 1350
detailed function point analysis	3. 1353
detailed function point count	3. 1354
detection error trade-off, ...	3. 1355
Determine Budget	3. 1356
deterministic system	3. 1357
develop project charter	3. 1358
develop project management plan	3. 1359
develop schedule	3. 1360
developed source statement	3. 1361
development	3. 1362
development approach	3. 1363
development approach and life cycle performance domain	3. 1364
development branch	3. 1365
development environment	3. 1366
development methodology	3. 1367
development pipeline, ...	3. 1368
development plan	3. 1369
development process	3. 1370
development project	3. 1371
development project function point count (DFP)	3. 1372
development project function point count, ...	3. 1373
development project functional size	3. 1374
development team	3. 1375
development testing	3. 1376

development tool	3. 1377
developmental baseline	3. 1378
developmental configuration	3. 1379
developmental configuration management	3. 1380
deviation	3. 1381
device interface task	3. 1382
DevOps	3. 1383
DF	3. 1162
DFD	3. 1165, 3. 1166
DFP	3. 1372, 3. 1373
DI	3. 1268
diagnostic	3. 1384
diagnostic manual	3. 1385
diagonal microinstruction	3. 1386
diagram boundary	3. 1387
diagram feature	3. 1388
diagram feature reference	3. 1389
diagram number	3. 1390
diagram page	3. 1391
diagram reference	3. 1392
diagram title	3. 1393
diagramming techniques	3. 1394
dialog	3. 1395
differential cash flow	3. 1396
digit numeric character	3. 1397, 3. 1398
digital asset	3. 1399
digital information content asset	3. 1400
digital product	3. 1401
digital signal processing	3. 1402, 3. 1403
digital signal processor	3. 1403, 3. 1404
digital versatile disc.	3. 1405
digraph	3. 1421
DII	3. 1406
dimension	3. 1407
DIP	3. 1514
direct address	3. 1409
direct address one-level address	3. 1408
direct cost	3. 1410
direct instruction	3. 1411
direct labor	3. 1412
direct labour	3. 1413
direct measure	3. 1414

direct memory access	3. 1415
direct memory access controller	3. 1416
direct metric	3. 1417
direct metric value	3. 1418
direct staff-hour	3. 1419
direct user	3. 1420
directed graph	3. 1421
directed SOS	3. 1422
directed system of systems	3. 1422
directory system agent	3. 1423
directory user agent	3. 1424
disassemble	3. 1425
disassembler	3. 1426
disaster recovery	3. 1427
disaster recovery plan	3. 1428
discipline-specific model	3. 1429
discounted payback period	3. 1430
discrete data	3. 1431
discrete effort	3. 1432
discrete type	3. 1433
discretionary dependency	3. 1434, 3. 1435
discrimination (threshold)	3. 1436
discriminator	3. 1437
disjoint substate	3. 1438
disposal	3. 1439
Disposal constraints	3. 1440
Disposal enabling system requirements	3. 1441
Disposal procedure	3. 1442
Disposal record	3. 1443
Disposal report	3. 1444
Disposal strategy	3. 1445
disposed system	3. 1446
distribute information	3. 1447
distributed computing	3. 1448
distributed denial of service attack, ...	3. 1449
distributed processing	3. 1450
distribution copy	3. 1451
distribution transparency	3. 1452
distribution unit	3. 1453
distributional shift	3. 1454
disturbance	3. 1455
DIT	3. 1456

DITA	3. 1132
dividing action	3. 1457
division of standards	3. 1458
DL	3. 1459
DMA	3. 1415
DMAC	3. 1416
DMS	3. 1464
DNN	3. 1253
do nothing alternative	3. 1460
document	3. 1461
document analysis	3. 1462
document control	3. 1463
document management system	3. 1464
document set	3. 1465
document type definition	3. 1466
document type definition (DTD)	3. 1467
documentation	3. 1468
documentation plan	3. 1469
documentation reviews	3. 1470
documentation tree	3. 1471
documented information	3. 1472
domain	3. 1473
domain analysis	3. 1474
domain architecture	3. 1475
domain architecture、product line architecture、reference architecture.....	3. 1476
domain asset	3. 1478
domain assets in requirements	3. 1477
domain component	3. 1479
domain engineering	3. 1480
domain engineering process	3. 1481
domain interface	3. 1482
domain layer	3. 1483
domain model	3. 1484
domain realization	3. 1485
domain requirements analysis	3. 1486
domain requirements elicitation	3. 1487
domain requirements management	3. 1488
domain requirements specification	3. 1489
domain requirements verification and validation	3. 1490
domain scoping	3. 1491
domain supersets	3. 1492
domain test asset	3. 1493

domain test requirement	3. 1494
domain testing	3. 1495
domain variability model	3. 1496
domain-based requirement	3. 1497
dominance	3. 1498
done	3. 1499
do-nothing operation	3. 1500
dot notation	3. 1501
double data rate SDRAM, ...	3. 1502
double data rate x 2	3. 1503
double-operand instruction	3. 1504, 3. 4843
down time	3. 1505
downgrade right	3. 1506
downward compatible	3. 1507
downward compression	3. 1508
DP	3. 1186, 3. 1187, 3. 1294
DPIA	3. 1189
DR	3. 1427
DRAM	3. 1537
drift	3. 1509
driver	3. 1510
DRP	3. 1428
DSA	3. 1423
DSL	3. 1267
DSP	3. 1402, 3. 1403, 3. 1404
DTC	3. 1207, 3. 1336, 3. 1341
DTD	3. 1466, 3. 1467, 3. 1511
DTP	3. 1345
DU	3. 1517
DUA	3. 1424
dual boot	3. 1512
dual coding	3. 1513
dual inline package	3. 1514
dumb terminal	3. 1515
dummy parameter	3. 1516
DUR	3. 1517
duration	3. 1517
duty	3. 1518
duty ethics	3. 1519
DVD	3. 1405
dyadic selective construct	3. 1520
dynamic	3. 1521

dynamic allocation	3. 1522
dynamic analysis	3. 1523
dynamic analyzer	3. 1524
dynamic attribute	3. 1525
dynamic binding	3. 1526
dynamic breakpoint	3. 1527
dynamic buffering	3. 1528
dynamic bus sizing	3. 1529
dynamic classification	3. 1530
dynamic dump	3. 1531
dynamic error	3. 1532
dynamic invocation	3. 1533
dynamic model	3. 1534
dynamic product	3. 1535
dynamic program analysis	3. 1536
dynamic random access memory	3. 1537
dynamic relocation	3. 1538
dynamic resource allocation	3. 1539
dynamic restructuring	3. 1540
dynamic schema	3. 1541
dynamic skeleton	3. 1542
dynamic storage allocation	3. 1543
dynamic testing	3. 1544

E

EA/IS	3. 1580
EAC	3. 1684, 3. 1685
early finish date	3. 1545
early start date (ES)	3. 1546
early-failure period	3. 1547
earned value (EV)	3. 1548
earned value (EV) budgeted cost of work performed (BCWP)	3. 1549
earned value analysis	3. 1550
earned value management (EVM)	3. 1551
earned value technique	3. 1552
echo	3. 1553
ECP	3. 1607
ECR	3. 1608
EDA	3. 1567
EDI	3. 1566
edit	3. 1554
editor	3. 1555
EEF	3. 1626

EEPROM	3. 1565
EF	3. 1545
effective address	3. 1556
effective full license	3. 1557
effective instruction	3. 1558
effective interest rate	3. 1559
effective process	3. 1560
efferent	3. 1561
effort	3. 1562
EFP	3. 1621, 3. 1622
egoless programming	3. 1563
EI	3. 1789
EIF	3. 1791
elasticity	3. 1564
electric erasable programmable read only memory	3. 1565
electronic data interchange	3. 1566
electronic design automation	3. 1567
element	3. 1568
element type	3. 1569
elementary process	3. 1570
ELF	3. 1793
ELSE-rule	3. 1571
embedded computer system	3. 1572
embedded help system	3. 1573
embedded information for users	3. 1574
emergence	3. 1575
emergency	3. 1576
emergency maintenance	3. 1577
emergency support	3. 1578
emitter	3. 1579
empathetic autonomous and intelligent system	3. 1580
empathy	3. 1581
employee	3. 1582
emulated user	3. 1583
emulator	3. 1584
enabled behavior	3. 1585
enabled behaviour	3. 1586
enabling (a transition)	3. 1587
enabling system	3. 1588
Enabling system requirements	3. 1589
enactment	3. 1590
encapsulation	3. 1591

encoding	3. 1592
ENCODING. 1	3. 1593
end item	3. 1594
end of period convention	3. 1595
end user	3. 1596
end user representatives	3. 1597
endeavor	3. 1598
endeavor element	3. 1599, 3. 1600
endeavour	3. 1601
endeavour element	3. 1600, 3. 1602
endurance testing	3. 1603
engineered system	3. 1604
engineering	3. 1605
engineering change	3. 1606
engineering change proposal	3. 1607
Engineering change request	3. 1608
engineering economics	3. 1609
engineering group	3. 1610
engineering interface reference	3. 1611
engineering interface reference management domain	3. 1612
engineering interface reference management domain.	3. 1613
engineering interface reference management policy	3. 1614
engineering interface reference management policy.	3. 1615
engineering viewpoint	3. 1616
enhanced capability	3. 1617
enhancement	3. 1618
enhancement project	3. 1619
enhancement project function point analysis	3. 1620
enhancement project function point count	3. 1621
enhancement project function point count (EFP)	3. 1622
ensure	3. 1623
Ent creator entitlement schema creator	3. 1624
enterprise	3. 1625
enterprise environmental factors	3. 1626
enterprise viewpoint	3. 1627
entitlement	3. 1628
entitlement management	3. 1629
entitlement schema	3. 1630
entity	3. 1631
entity attribute	3. 1632
entity component	3. 1633
entity dependent	3. 1634

entity independent	3. 1635
entity instance	3. 1636
entity of interest	3. 1637
entity-relationship (E-R) diagram	3. 1638
entity-relationship diagram	3. 1201, 3. 1639
entity-relationship map	3. 1639
entrance	3. 1646
entry	3. 1646
entry (type)	3. 1640
entry (-type)	3. 1641
entry action	3. 1642
entry criteria	3. 1643
entry field	3. 1644
entry point	3. 1646
entry profile	3. 1645
enumeration type	3. 1647
environment	3. 1648
environment contract	3. 1649
environment specific inter-ORB protocol	3. 1650
E0	3. 1798
Eol	3. 1637
EP	3. 1570
epic	3. 1651
epilog breakpoint	3. 1652, 3. 1653
epistemic logic	3. 1654
epoch	3. 1655
EPROM	3. 1666, 3. 1667
EQ	3. 1790
equation	3. 1656
equipment	3. 1657
equivalence class	3. 1658
equivalence partition	3. 1659
equivalence partition coverage	3. 1660
equivalence partitioning	3. 1661
equivalent faults	3. 1662
equivalent IDL	3. 1663
equivalent interface	3. 1664
E-R diagram	3. 1638, 3. 1639
ERA	3. 1665
erasable programming read only memory	3. 1666
erasable programming read only memory (EPROM)	3. 1667
errata	3. 1668

error	3. 1669
error analysis	3. 1670
error category	3. 1671
error data	3. 1672
error guessing	3. 1673
error message	3. 1674
error model	3. 1675
error prediction	3. 1676
error prediction model	3. 1675
error processing	3. 1677
error seeding	3. 1678
error tolerance	3. 1679
ES	3. 1759, 3. 1760
escaped	3. 1680
ESIOP	3. 1650
establish and maintain	3. 1681
established requirement	3. 1682
establishing behavior	3. 1683
estimate	3. 1756
estimate at completion	3. 1684
estimate at completion (EAC)	3. 1685
estimate to complete	3. 1686
estimated function point count	3. 1687
estimating methods	3. 1688
estimation	d
ETA	3. 1722
ETC	3. 1686
ethical policy statement	3. 1690
ethical principle	3. 1691
ethical requirement.	3. 1692
ethical risk	3. 1693
ethical value	3. 1694
ethical value requirement (EVR)	3. 1695
EV	3. 1548, 3. 1549
evaluation	3. 1696
evaluation activity	3. 1697
evaluation checklist	3. 1698
evaluation coverage	3. 1699
evaluation group	3. 1700
evaluation level	3. 1701
evaluation method	3. 1702
evaluation module	3. 1703

evaluation procedure	3. 1704
evaluation records	3. 1705
evaluation report	3. 1706
evaluation requester	3. 1707
evaluation sponsor	3. 1708
evaluation stringency	3. 1709
evaluation technology	3. 1710
evaluation tool	3. 1711
evaluator	3. 1712
event	3. 1713
event history	3. 1714
event record	3. 1715
event sequence analysis	3. 1716
event sequence diagram	3. 1717
event sink	3. 1718
event source	3. 1719
event synchronization	3. 1720
event trace	3. 1721
Event tree analysis	3. 1722
event-driven review/activity	3. 1723
event-sequencing logic	3. 1724
evidence	3. 1725
EVM	3. 1551, 3. 1703
examination	3. 1726
exception	3. 1727
exception handling	3. 1728
executable requirements specification	3. 1729
executable source statement	3. 1730
executable statement	3. 1731
execute	3. 1732
executing process group	3. 1733
executing processes	3. 1734
execution	3. 1735
execution efficiency	3. 1736
execution monitor	3. 1737
execution time	3. 1738
execution time theory	3. 1739
execution trace	3. 1740
executive	3. 1020, 3. 1742, 3. 4391
executive program	3. 1020, 3. 1742, 3. 4391
executive state	3. 1741, 3. 4390
executor	3. 1743

exhaustive testing	3. 1744
existence constraint	3. 1745
existence dependency	3. 1746
existing software	3. 1747
exit	3. 1748
exit action	3. 1749
exit criteria	3. 1750
exit point	3. 1751
exit routine	3. 1752
expandability	3. 1753, 3. 1772
expected result	3. 1754
expected value	3. 1755
expected value 50-50	3. 1756
expected value of perfect information	3. 1757
experience-based testing	3. 1758
expert system	3. 1759
expert system (ES)	3. 1760
explainability	3. 1761
explanatory report	3. 1762
explicit address	3. 1763
explicit knowledge	3. 1764
explicit requirement	3. 1765
exploitable	3. 1766
exploratory testing	3. 1767
export	3. 1768
export process	3. 1769
expression	3. 1770
extend	3. 1771
extendability	3. 1772
extended component	3. 1773
extended element	3. 1774
extended entry table	3. 1775
extended process set	3. 1776
extensible markup language	3. 1777
EXtensible Style Language Transformations	3. 1778
EXtensible Style Language Transformations (XSLT)	3. 1779
extensional set	3. 1780, 3. 1781
extensive logical operation	3. 1782
extensive mathematical operation	3. 1783
external	3. 1784
external attribute	3. 1785
external dependency	3. 1786

external event	3. 1787
external I/O device	3. 1788
external input	3. 1789
external inquiry	3. 1790
external interface file	3. 1791
external interface requirement	3. 1792
external logical file	3. 1793
external measure	3. 1794
external measure (of system or software quality)	3. 1795
external measure of software quality	3. 1796
external measure of system or software quality	3. 1797
external output	3. 1798
external quality	3. 1799
external software quality	3. 1800
external variability	3. 1801
extractive approach	3. 1802
extranet	3. 1803
extreme programming	3. 1804

F

(FTR)	3. 1862
F1 score	3. 1805
F1-score	3. 1805
facet	3. 1806
faceted search	3. 1807
facilitated workshops	3. 1808
facility	3. 1809
factoring	3. 1810
factory	3. 1811
fail safe	3. 1812
fail soft	3. 1813
failure	3. 1814
failure category	3. 1815
failure data	3. 1672
failure intensity	3. 1816
failure intensity objective	3. 1817
failure mode	3. 1818
failure mode and effect analysis	3. 1819
Failure Mode and Effects Analysis	3. 1820
failure rate	3. 1821
failure ratio	3. 1821
failure recovery	3. 1822
failure severity	3. 1823

failure transparency	3. 1824
fallback plan	3. 1825
false negative	3. 1826
false positive	3. 1827
families of programs	3. 1828
family of platforms	3. 1829
fast tracking	3. 1830
fatal error	3. 1831
fault	3. 1832, 3. 1840
fault category	3. 1671
fault dictionary	3. 1833
fault insertion	3. 1837
fault isolation	3. 1834
fault masking	3. 1835
fault secure	3. 1836
fault seeding	3. 1678, 3. 1837
fault tolerance	3. 1679, 3. 1838
Fault tree analysis	3. 1839
faultlessness	3. 1841
FCA	3. 1945, 3. 1946
FDC	3. 1949
feasibility study	3. 1842
feature	3. 1843
feature branch	3. 1844
feature catalog	3. 1845
feature catalogue	3. 1845, 3. 1846
feature constraint	3. 1847
feature engineering	3. 1848
feature freeze	3. 1849
feature language	3. 1850
feature reference	3. 1851
feature selection	3. 1848
feature set	3. 1852
FEO page	3. 1890
FF	3. 1853, 3. 1865
FFP contract	3. 1868
FHA	3. 1950
field	3. 1178
field of application (of a specification)	3. 1854
field programmable gate array	3. 1855
fieldbus	3. 1856
FIFO	3. 1857

fifth-generation language	3. 1858
fifth-generation language (5GL)	3. 1859
figurative constant	3. 1860
file type referenced	3. 1861, 3. 1862
final accounts	3. 1863
final function point analysis	3. 1864
finish-to-finish	3. 1853, 3. 1865
finish-to-start	3. 1866
finite state machine	3. 1867
FIO	3. 1817
firm-fixed-price contract	3. 1868
firmware	3. 1869
first input routine	3. 1870
first normal form	3. 1871
first-line software manager	3. 1872
fitness for purpose	3. 1873
fitness-for-use test	3. 4904
fixed cost	3. 1874
fixed duration	3. 1875
fixed price with economic price adjustment contract	3. 1876
fixed-price-incentive-fee contract	3. 1877
flag	3. 1878
flash memory	3. 1879
flip-flop	3. 1880, 3. 1881
float	3. 1882
flow	3. 1883
flow chart	3. 1885
flow diagram	3. 1885
flow of control	3. 1884
flowchart	3. 1885
flowcharter	3. 1886
flowcharting	3. 1887
FMEA	3. 1819, 3. 1820
FN	3. 1826
focus class	3. 1888
focus of control	3. 1889
for Exposition Only page	3. 1890
For Exposition Only page (FEO page)	3. 1891
foreground	3. 1892
foreground processing	3. 1893
foreign key	3. 1895
foreign key migrated key	3. 1894

forking action	3. 1896
form	3. 1897
form, fit, and function	3. 1898
formal design	3. 1899
formal evaluation process	3. 1900
formal language	3. 1901
formal parameter	3. 1902
formal qualification review	3. 1903
formal requirements language	3. 1904, 3. 1905
formal review	3. 1906
formal specification	3. 1907
formal testing	3. 1908
formal verification	3. 1909
formalization	3. 1910
formative construct	3. 1911
formative evaluation	3. 1912
forward pass	3. 1913
forward recovery	3. 1914
fountain model	3. 1915
four-address instruction	3. 1916
four-plus-one address instruction	3. 1917
fourth generation language (4GL)	3. 1918
FP	3. 1934, 3. 1939
FPA	3. 1935, 3. 1936
FP-EPA contract	3. 1876
FPGA	3. 1855
FPIF contract	3. 1877
F-profile	3. 1919
FQR	3. 1903
frame	3. 1920
frame a concern	3. 1921
framework	3. 1922
free float	3. 1923
freedom from economic risk	3. 1924
freedom from environmental and societal risk	3. 1925
freedom from health risk	3. 1926
freedom from human life risk	3. 1927
freedom from risk	3. 1928
frozen branch	3. 1929
FS	3. 1866, 3. 1953
FSM	3. 1955
FSMM	3. 1954

FTA	3. 1839
FTR	3. 1861, 3. 1862, 3. 1930
full duplex	3. 1931
function	3. 1932
function field	3. 1933
function point	3. 1934
function point analysis	3. 1935, 3. 2032
function point analysis, ...	3. 1936
function point count	3. 1937
function point table	3. 1938
function point, ...	3. 1939
function type	3. 1940
functional adaptability	3. 1941
functional analysis	3. 1942
functional appropriateness	3. 1943
functional baseline	3. 1944
Functional configuration audit	3. 1945
functional configuration audit (FCA)	3. 1946
functional configuration identification	3. 1947
functional design	3. 1948
functional domain categorization, ...	3. 1949
Functional hazard analysis	3. 1950
functional process	3. 1951
functional requirement	3. 1952
functional size	3. 1953
functional size measurement method	3. 1954
functional size measurement, ...	3. 1955
functional specification	3. 1956
functional system design	3. 1957
functional testing	3. 4204
functional user	3. 1958
functional user requirements (FUR)	3. 1959
functional user requirements, ...	3. 1960
FUR	3. 1959, 3. 1960

G

GDPR	3. 1963
gemba	3. 1961
general AI	3. 1962
general artificial intelligence	3. 1962
General Data Protection Regulation	3. 1963
general purpose input/output port (GPIO)	3. 1964
general purpose input/output port, ...	3. 1965

general register	3. 1966
general system characteristics	3. 1967
generality	3. 1968
generalization	3. 1969
generalization structure	3. 1970
generalization taxonomy	3. 1971
generally accepted	3. 1972
general-purpose input/output port	3. 1973
generated address	3. 1974
generation	3. 1975
generic entity	3. 1976
generic practice	3. 1977
generic profile group	3. 1978
generic program unit	3. 1979
glass box	3. 1980, 3. 5110
glass-box testing	3. 1981, 3. 4328, 3. 4329
global attribute	3. 1982
global compaction	3. 1983
global data	3. 1984
global label	3. 1985
globally unique identifier	3. 1986
goal	3. 1987
Government-off-the-Shelf	3. 1988
GPIO	3. 1964, 3. 1965, 3. 1973
GPU	3. 1992
grade	3. 1989
granularity	3. 1990
graphical information	3. 1991
graphical processing unit	3. 1992
graphical symbol	3. 1993
GraphicSymbol	3. 1994
greenfield system engineering	3. 1995
ground rules	3. 1996
ground term	3. 691
ground truth	3. 1997
group	3. 1998
groups	3. 1998
GSCs	3. 1967
GUI	3. 1999
GUID	3. 1986
guide	3. 2000
guideline	3. 2001

H

half duplex	3. 2002
HAM	3. 2005
hard copy	3. 2003
hard failure	3. 2004
hardware	3. 2012
hardware asset management, ...	3. 2005
hardware configuration item	3. 2006
hardware configuration item(HWCI)	3. 2007
hardware configuration item, ...	3. 2008
hardware engineering	3. 2009
hardware identification creator, ...	3. 2010
hardware identification schema, ...	3. 2011
harm	3. 2013
harm from use	3. 2014
Harvard architecture	3. 2015
hazard	3. 2016
hazard identification	3. 2017
hazard warning	3. 2018
hazardous event	3. 2019
hazardous situation	3. 2020
HCI	3. 2006, 3. 2007, 3. 2008
header	3. 2021
heavyweight process	3. 2022
help system	3. 2023
heuristic evaluation	3. 2024
hierarchical decomposition	3. 2025
hierarchical modeling	3. 2026
hierarchically consecutive	3. 2027
hierarchy	3. 2028
hierarchy chart	3. 2029, 3. 4331
high level	3. 2032
high level model	3. 2030
high level modeling method	3. 2031
higher-level management	3. 2033
high-level design	3. 2034
high-level keyword	3. 2035
high-level net	3. 2036
high-level Petri Net graph	3. 2037
high-order language	3. 2038
high-order language (HOL)	3. 2039
historical information	3. 2040

history file	3. 2041
HMI	3. 2042
HOL	3. 2038
homogeneous redundancy	3. 2043
host computer	3. 2044
host machine	3. 2045
hostile backout	3. 2046
HRS	3. 2047
HSI	3. 2051
HTML	3. 2059
HTTP	3. 2060
human factors	3. 2048
human resource planning	3. 2049
human systems engineering	3. 2050
Human systems integration	3. 2051
human-centered design	3. 2052, 3. 2053
human-centred design	3. 2054
human-centred quality	3. 2055
HW	3. 2012
HWID creator	3. 2010
HWID schema	3. 2011
hybrid approach	3. 2056
hybrid coupling	3. 2057
hyperlink	3. 2058
hypertext markup language	3. 2059
HyperText Transfer Protocol	3. 2060

I

I/O	3. 2061
I/O task-structuring criteria	3. 2062
I2C	3. 2282
IaC	3. 2191
IAP	3. 2103
IBD	3. 2188
IC	3. 2245, 3. 2246
ICD	3. 2063, 3. 2272
ICE	3. 2109
ICOM code, ...	3. 2064
ICOM label	3. 2065
icon	3. 2066
ICS	3. 2067
ICT product, ...	3. 2068
ICT requirement, information and communication technology requirement	3. 2069

IDD	3. 2070, 3. 2273
IDE	3. 2247, 3. 2248
idea mapping	3. 2071, 3. 2072
idea/	3. 2072
idea/mind mapping, ...	3. 2071
ideal time	3. 2073
IDEF	3. 2251
identification transaction	3. 2074
identifier-dependent entity	3. 1289
identifying relationship	3. 2075
identity	3. 2076
identity-style view	3. 2077
IDIQ	3. 2126
IFB	3. 2321
if-then-else	3. 2078
IIC	3. 2282
ILF	3. 2079, 3. 2292
illustration	3. 2080
IM	3. 2107
immunity provision	3. 2081
immutable class	3. 2082
impact analysis	3. 2083
impact mapping	3. 2084
impairment	3. 2085
impediment	3. 2086
imperfect debugging	3. 2087
implementable standard	3. 2088
implementation	3. 2089, 3. 2090
implementation class	3. 2091
implementation phase	3. 2092
implementation process	3. 4618
implementation rating module	3. 2093
implementation requirement	3. 2094
implied need	3. 2095
import	3. 2096
import process	3. 2097
importer	3. 2098
impossible zone	3. 2099
improvability	3. 2100
improvement backlog	3. 2101
Improvement plan	3. 2102
in-application programming	3. 2103

in-application programming (IAP)	3. 2104
INC	3. 2105
incident	3. 2106
incident manager (IM)	3. 2107
incident report	3. 2108
in-circuit emulator	3. 2109
in-circuit emulator (ICE)	3. 2110
include	3. 2111
inclusivity	3. 2112
income function	3. 2113
incomplete process	3. 2114
inconsistency ratio	3. 2115
increment	3. 2116
incremental analysis	3. 2117
incremental approach	3. 2118
incremental benefit	3. 2119
incremental compiler	3. 1037, 3. 2120, 3. 2121
incremental development	3. 2122
incremental investment	3. 2123
incremental life cycle	3. 2124
incremental productivity	3. 2125
indefinite delivery indefinite quantity	3. 2126
independence	3. 2127
independent	3. 2128
independent estimates	3. 2129
independent evaluator	3. 2130
independent state class	3. 2131
independent verification and validation	3. 2132
independent verification and validation (IV&V)	3. 2133
independent verification and validation(IV&V)	3. 2134
index	3. 2135
indicative function point analysis	3. 2136
indicative function point count	3. 2137
indicator	3. 2138
indicator value	3. 2139
indigenous error	3. 2140
indirect cost	3. 2141
indirect labour	3. 2142
indirect measure	3. 2143
indirect user	3. 2144
individual interface	3. 2145
inductive assertion method	3. 2146

inference	3. 2147
influence diagram	3. 2148
informal group review	3. 2149
informal review	3. 2150
information	3. 2151
information analysis	3. 2152
information and communication technology product	3. 2068
information architecture	3. 2153
information content	3. 2154
information design	3. 2155
information development	3. 2156
information for use	3. 2157
information for users	3. 2158
information gathering techniques	3. 2159
information hiding	3. 2160
information item	3. 2161
information item content	3. 2162
information item type	3. 2163
Information management plan	3. 2164
information management system	3. 2165
information model	3. 2166
information need	3. 2167
information part	3. 2168
information processing requirements	3. 2169
information processing system	3. 2170
information product	3. 2171
information provisioning	3. 2172
information radiator	3. 2173
information retrieval	3. 2174
information security	3. 2175
information security incident	3. 2176
information security policy	3. 2177
information structure	3. 2178
information structure creator	3. 2179
information structure id	3. 2180
information system needs	3. 2181
information technology asset	3. 2182
information technology asset management	3. 2333
information technology asset management plan	3. 2330
information technology asset management system	3. 2183, 3. 2332
information technology asset portfolio	3. 2334
information technology infrastructure management	3. 2336

information technology project	3. 2184
information technology project dataset	3. 2337
information technology service	3. 2185
information technology service adaptability	3. 2339
information technology service function	3. 2340
information type	3. 2186
information viewpoint	3. 2187
information-based domain	3. 2188
informed consent	3. 2189
infrastructure	3. 2190
infrastructure as code	3. 2191
infrastructure ecosystem	3. 2192
inheritance	3. 2090
inherited attribute	3. 2193
inhibitor arc	3. 2194
initial Ent	3. 2195
initial function point analysis	3. 2196
initial function point count	3. 2197
initial investment	3. 2198
initial marking (of the net)	3. 2199
initial marking of a place	3. 2200
initial risk	3. 2201
initialization section	3. 2202
initialize	3. 2203
initiating object	3. 2204
initiating process group	3. 2205
initiating processes	3. 2206
initiation trigger	3. 2207
initiative	3. 2208
inner cardinality	3. 2209
input	3. 2210
input arc (of a transition)	3. 2211
input argument	3. 2212
input assertion	3. 2213
input loopback	3. 2214
input place (of a transition)	3. 2215
input primitive	3. 2216
input routine	3. 2217
input-output coupling	3. 1147, 3. 2218
input-process-output	3. 2219
input-process-output (IPO) chart	3. 2220
input-process-output chart	3. 2221

input-process-output chart/IP0 chart	3. 2222
in-service	3. 2223
in-service period	3. 2224
in-service reference model	3. 2225
inspection	3. 2226
inspection-based evaluation	3. 2227
inspections and audits	3. 2228
installability	3. 2229
installability testing	3. 2230
installation and checkout phase	3. 2231
installation and check-out phase	3. 2232
installation manual	3. 2233
installed function point count	3. 2234
instance data	3. 2675
instance of benchmarking	3. 2235
instance-level attribute	3. 2236
instance-level operation	3. 2237
instantiated trace link	3. 2238
instantiation	3. 2239
institutional knowledge	3. 2240
instructional information	3. 2241
instrument	3. 2242
instrumentation	3. 2243
in-system programming	3. 2244
integrated circuit	3. 2245, 3. 2246
integrated development environment	3. 2247
integrated development environment (IDE)	3. 2248
integrated repository	3. 2249
integration	3. 2250
integration definition for functional modeling	3. 2251
Integration plan	3. 2252
integration testing	3. 2253
integrity level	3. 2254
integrity level claim	3. 2255
integrity level definition authority	3. 2256
integrity level requirements	3. 2257
integrity level scheme	3. 2258
intelligent agent	3. 2259
intelligent vision system	3. 2260
intended use	3. 2261
interaction capability	3. 2262
interaction group	3. 2263

interaction point	3. 2264
interaction sequence	3. 2265
interactive	3. 2266
interactive compiler	3. 2121
interactive language	3. 2267
interactive system	3. 2268
interchange reference point	3. 2269
interface	3. 2270
interface control	3. 2271
Interface control document	3. 2272
interface design document	3. 2273
interface repository	3. 2274
interface requirement	3. 2275
interface requirements specification	3. 2276
interface role	3. 2277
interface signature	3. 2278
interface specification	3. 2279
interface task	3. 2280
interim function point analysis	3. 2281
inter-integrated circuit bus	3. 2282
inter-integrated circuit bus (I2C)	3. 2283
interleave	3. 2284
intermediate product	3. 2285
intermediate software product	3. 2285
intermediate software product need	3. 2286
intermediate system product	3. 2285
intermittent fault	3. 2287
internal action	3. 2288
internal arrow	3. 2289
internal attribute	3. 2290
internal dependency	3. 2291
internal logical file	3. 2292
internal measure (of software quality)	3. 2293
internal service provider	3. 2294
internal task-structuring criteria	3. 2295
internal variability	3. 2296
internet of things	3. 2297
interoperability	3. 2298
interoperability testing	3. 2299
interoperating system	3. 2300
interpret	3. 2301
interpretability	3. 2302

interpreter	3. 2303
interpretive code	3. 2304
interrelationship digraphs	3. 2305
interrogation	3. 2306
interrupt controller	3. 2307
interrupt latency	3. 2308
interrupt mask	3. 2309
interrupt priority	3. 2310
interrupt request	3. 2311
interrupt service routine ISR	3. 2312
interval scale	3. 2313
intervenability	3. 2314
interviews	3. 2315
interworking reference point	3. 2316
intrinsic	3. 2317
intrinsic relationship	3. 2318
invariant	3. 2319
invariant schema	3. 2320
invitation for bid	3. 2321
invitation to tender	3. 2322
invocation deliver	3. 2323
invocation submit	3. 2324
IoT	3. 2297
IPO chart	3. 2221
IPO diagram	3. 2325
IR	3. 2174, 3. 2274
IRS	3. 2276
isochronicity	3. 2326
isolation	3. 2327
ISP	3. 2224, 3. 2244
ISRM	3. 2225
issue	3. 2328
IT asset	3. 2182
IT asset management	3. 2329, 3. 2333
IT asset management plan	3. 2330
IT asset management system	3. 2331, 3. 2332
IT asset portfolio	3. 2334
IT infrastructure	3. 2335
IT infrastructure management	3. 2336
IT project dataset	3. 2337
IT service	3. 2338
IT service adaptability	3. 2339

IT service function	3. 2340
IT service system	3. 2341
ITAM	3. 2329, 3. 2333
ITAMS	3. 2183, 3. 2331, 3. 2332
iteration	3. 2342
iteration backlog	3. 2343
iteration plan	3. 2344
iteration planning	3. 2345
iteration review	3. 2346
iterative approach	3. 2347
iterative construct	3. 2348
iterative development	3. 2349
iterative life cycle	3. 2350
iterative process	3. 2351
ITT	3. 2322
IV&V	3. 2132
IVS	3. 2260

J

JCL	3. 2353
job	3. 2352
job control language	3. 2353
job function	3. 2354
job stream	3. 2355
JSON (JavaScript Object Notation) web token	3. 2356
jump	3. 2357
JWT	3. 2356

K

KA	3. 2378
kanban board	3. 2358, 3. 5132
K-base	3. 2379
KDM entity	3. 2359
KDM instance	3. 2360
KDM model	3. 2361
KDM relationship	3. 2362
kernel	3. 2363
kernel resident control program	3. 2364
key migration	3. 2365
Key Performance Parameter	3. 2366
key practices	3. 2367
key process area	3. 2368
key-style view	3. 2369
keyword	3. 2370

keyword dictionary	3. 2371
keyword execution code	3. 2372
keyword library	3. 2371
keyword test case	3. 2373
keyword-driven testing	3. 2374
keyword-driven testing framework	3. 2375
kickoff meeting	3. 2376
kilo-operations per second	3. 2377
knowledge area	3. 2378
knowledge base	3. 2379
KOPS	3. 2377
KPA	3. 2368
KPP	3. 2366

L

label	3. 2380
LAN	3. 2440
language processor	3. 2381
language standard	3. 2382
last responsible moment	3. 2383
latch	3. 1880, 3. 1881
latency	3. 2384
latent variable	3. 2385
lateral compression	3. 2386
launcher	3. 1510
layer	3. 2387
layout	3. 2388
LCA	3. 2417
LCC	3. 2418
LCL	3. 2441
LCM	3. 2419
lead startup canvas	3. 2389
lead time	3. 2390
lead time chart	3. 2391
leading decision	3. 2392
left-shift	3. 2393
legacy hardware	3. 2394
legacy software	3. 2395
legend	3. 2396
lessons learned	3. 2397
lessons learned register	3. 2398, 3. 2399
lessons learned repository	3. 2399
letter of understanding	3. 2400

level	3. 2401
level of abstraction	3. 2402
level of documentation	3. 2403
level of effort	3. 2404
Level of Repair Analysis	3. 2405
level of risk	3. 2406
level of service	3. 2407
Levenshtein distance	3. 2408
lexicography	3. 2409
liaison	3. 2410
library	3. 2411
license	3. 2412
license compliance audit	3. 2413
license model	3. 2414
licensing standard	3. 2415
life cycle	3. 2416
life cycle assessment	3. 2417
life cycle cost	3. 2418
Life cycle management	3. 2419
life cycle model	3. 2420
life cycle processes	3. 2421
lightweight process	3. 2422
likelihood	3. 2423
limit	3. 2424
limited entry table	3. 2425
line of code	3. 2426
linear interpolation	3. 2427
link	3. 2428
linking loader	3. 2429
list	3. 2430
list function	3. 2431
listing	3. 2432
literal	3. 2433
load	3. 2434
load map	3. 2435
load module	3. 2436
load-and-go	3. 2437
loaded origin	3. 2438
loader	3. 2439
local area network	3. 2440
local coincidence logic	3. 2441
local compaction	3. 2442

local customization	3. 2443
local data	3. 2444
local SAM owner	3. 2445
local variable	3. 2446
locality of quality responsibility	3. 2447
localization	3. 2448
location facility	3. 2449
location in space	3. 2450
location in time	3. 2451
location reference	3. 2452
location transparency	3. 2453
lock	3. 2454
lockout	3. 2455
LOE	3. 2404
log	3. 2456
log in	3. 2458
log off	3. 2457
log on	3. 2458
log out	3. 2457
logic	3. 2459
logic programming language	3. 2460
logical cohesion	3. 2461
logical file	3. 2462
logical interface	3. 2463
logical layout	3. 2464
logical record	3. 2465
logical relationship dependency	3. 2466
logical source statement	3. 2467
logical trace	3. 2468
logical transaction	3. 2469
logical type	3. 2470
loop	3. 2471
loop assertion	3. 2472
loop body	3. 2473
loop control	3. 2474
loop control variable	3. 2475
loosely coupled message communication	3. 359, 3. 360
LORA	3. 2405
lossage	3. 2476
LOU	3. 2400
low level	3. 2477
lowclass	3. 2478

low-level design	3. 2479
low-level keyword	3. 2480
low-level language, ...	3. 306
LSS	3. 2467
LTC	3. 2481
LTL	3. 2482

M

machine learning, ...	3. 2483
machine readable	3. 2484
machine translation	3. 2485
machine-readable	3. 2486
machine-readable passport	3. 2689
macro	3. 2487
macro definition	3. 2488
macro generating program	3. 2489
macro library	3. 2490
macroassembler	3. 2491
macrogenerator	3. 2492
macroinstruction	3. 2493
macroprocessor	3. 2494
macroprogramming	3. 2495
MADM	3. 2696
main program	3. 2496
main schedule	3. 2497
maintain repository activity	3. 2498
maintainability	3. 2499
maintenance	3. 2500
maintenance manual	3. 2501
maintenance phase	3. 2502
maintenance plan	3. 2503
Maintenance procedure	3. 2504
maintenance program maintenance infrastructure	3. 2505
make or buy analysis	3. 2506
managed and controlled	3. 2507
management process	3. 2508
management reserve	3. 2509
management review	3. 2510
management skills	3. 2511
management system	3. 2512
managerial independence	3. 2513
managing role	3. 2514
mandatory	3. 2515

mandatory dependency	3. 2516
mandatory nonidentifying relationship	3. 2517
manual testing	3. 2518
manufacture	3. 2519
manufacturing phase	3. 2520
many-sorted algebra	3. 2521
many-to-many relationship	3. 2522
map program	3. 2523
mapping	3. 2524
mapping completeness	3. 2525
mapping document	3. 2526
marking	3. 2527
marking (of a net)	3. 2528
marking of a place	3. 2529
markup language	3. 2530
MARR	3. 2627, 3. 2628
mask	3. 2531
mask ROM	3. 2532
master state	3. 4390
material	3. 2533
matrix diagrams	3. 2534
maturity	3. 2535
maturity level	3. 2536
maturity model	3. 2537
maturity questionnaire	3. 2538
maximax rule	3. 2539
maximin rule	3. 2540
MBSE	3. 2660
MBSSE	3. 2659
MC/DC testing	3. 2666
MCDA	3. 2700
MCDM	3. 2708
MCU	3. 2608
MD5	3. 2582
MDA	3. 2645
mean	3. 2541
mean execution time	3. 2542
mean execution time rating value	3. 2543
mean execution time reference value	3. 2544
mean time between failures	3. 2545
mean time between failures (MTBF)	3. 2546
mean time between failures, ...	3. 2547

mean time between software failures	3. 2691
mean time between support	3. 2692
mean time to detection	3. 2548
mean time to failure	3. 2694
mean time to impairment recovery	3. 2693
mean time to repair	3. 2549
mean time to repair (MTTR)	3. 2550
mean time to x	3. 2695
meaningful	3. 2551
measure (noun)	3. 2552
measure of effectiveness	3. 2553
measure of performance	3. 2554
measurement	3. 2555
measurement analyst	3. 2556
measurement experience base	3. 2557
measurement function	3. 2558
measurement librarian	3. 2559
measurement method	3. 2560
measurement model	3. 2561
measurement performance domain	3. 2562
measurement procedure	3. 2563
measurement process	3. 2564
measurement process owner	3. 2565
Measurement repository	3. 2566
measurement source	3. 2567
measurement sponsor	3. 2568
measurement standard	3. 2569
measurement user	3. 2570
measuring instrument	3. 2571
mechanism loopback	3. 2572
member product	3. 2573
memory	3. 2574
memory capacity	3. 2575
memory map	3. 2576
menu	3. 2577
menu structure	3. 2578
merge	3. 2579
merge from current	3. 2580
message	3. 2581
message digest 5	3. 2582
Message-Digest algorithm 5	3. 2582
meta	3. 2583

meta-attribute	3. 2584
metaclass	3. 2585
metacompiler	3. 2586
metadata	3. 2587
meta-entity	3. 2588
metalanguage	3. 2589
meta-meta-attribute	3. 2590
meta-meta-entity	3. 2591
meta-metamodel	3. 2592
meta-meta-relationship	3. 2593
metamodel	3. 2594
metamodel element	3. 2595
metamorphic relation	3. 2596, 3. 2690
meta-object	3. 2597
meta-object facility	3. 2598
meta-object facility MOF	3. 2599
meta-relationship	3. 2600
method	3. 2601
method engineer	3. 2602
method standard	3. 2603
methodology	3. 2604
methodology element	3. 2605
metric	3. 2606
micro code assembler	3. 2607
microchip	3. 2246
microcontroller unit	3. 2608
microgenerator	3. 2609
microinstruction	3. 2610
microoperation	3. 2611
microprogram	3. 2612
microprogrammable computer	3. 2613
microprogrammed computer	3. 2614
microprogramming	3. 2615
middleware	3. 2616
migratability	3. 2617
migrated attribute	3. 2618
migrated key	3. 1895
migration	3. 2619
migration transparency	3. 2620
milestone list	3. 2621
milestone review	3. 2622
milestone schedule	3. 2623

mind mapping	3. 2072
Mind-Mapping	3. 2624
minimalism	3. 2625
minimax regret rule	3. 2626
minimum attractive rate of return	3. 2627, 3. 2628
minimum delay programming	3. 2629
minimum tasks	3. 2630
minimum viable product	3. 2631
mirror site	3. 2632
mission	3. 2633
mistake	3. 2634
mixed entry table	3. 2635
mixed mode	3. 2636
ML	3. 2483
mobile device	3. 2637
mobility schema	3. 2638
mock object	3. 2639
mock-up	3. 2640
mode	3. 2641
model	3. 2642
model baseline	3. 2643
model configuration item	3. 2644
model driven architecture	3. 2645
model element	3. 2646
model element library	3. 2647
model glossary	3. 2648
model hierarchy	3. 2649
model kind	3. 2650
model layers	3. 2651
model name	3. 2652
model name abbreviation	3. 2653
model note	3. 2654
model note number	3. 2655
model page	3. 2656
model pattern	3. 2657
model repository	3. 2658
model-based systems and software engineering, ...	3. 2659
model-based systems engineering	3. 2660
modeling	3. 2661
modeling tool	3. 2662
modifiability	3. 2663
modifiable	3. 2664

modification request	3. 2665
modified condition/decision coverage testing	3. 2666
modified source statement	3. 2667
modified-off-the-shelf	3. 2668
modular decomposition	3. 2669
modular programming	3. 2670
modularity	3. 2671
modularization	3. 2672
module	3. 2673
module data	3. 2674, 3. 2675
module strength	3. 2676
MOE	3. 2553
MOF	3. 1263, 3. 2599
monitor	3. 2677
monitor and control project work	3. 2678
monitor and control risks	3. 1024, 3. 2679
monitor communications	3. 2680
monitor risks	3. 2681
monitor stakeholder engagement	3. 2682
monitorability	3. 2683
monitoring	3. 2684
monitoring and controlling process group	3. 2685
monolithic executor	3. 2686
Monte Carlo simulation	3. 2687
mood chart	3. 2688
MOP	3. 2554
MOTS	3. 2668
MR	3. 2690
MRa	3. 2498
MRP	3. 2689
MT	3. 2485
MTBF	3. 2545, 3. 2546, 3. 2547
MTBS	3. 2692
MTBSF	3. 2691
MTIR	3. 2693
MTTD	3. 2548
MTTF	3. 2694
MTTR	3. 2549, 3. 2550
MTX	3. 2695
multi-attribute decision	3. 2696
multi-component system	3. 2697
multi-core	3. 2698

multi-core processor	3. 2699
multi-criteria decision analysis	3. 2700
multidimensional construct	3. 2701
multidisciplinary collaborative simulation	3. 2702
multilevel security	3. 2703
multilevel storage	3. 5075
multi-modal biometric system	3. 2704
multiple inheritance	3. 2705
multiple instance approach	3. 2706
multiple readers and writers	3. 2707
multiple-criteria decision making	3. 2708
multiple-hit decision table	3. 2709
multipoint estimating	3. 2710
multiset	3. 2712
multiset cardinality	3. 2711
multi-valued property	3. 2713
mutable class	3. 2714
mutual exclusion	3. 2715
mutually exclusive	3. 2716
mutually exclusive clustering	3. 2717
mutually inclusive	3. 2718
MVP	3. 2631

N

N 2 diagram, N2 diagram	3. 2719
N2 diagram	3. 2778
name resolution	3. 2720
name space	3. 2721
named constant	3. 2722
named constraint	3. 2723
naming action	3. 2724
naming context	3. 2725
naming domain	3. 2726
naming graph	3. 2727
narrow artificial intelligence	3. 2728
N-ary relationship	3. 2729
natural unit	3. 2730
navigation	3. 2731
navigational aids	3. 2732
NDA	3. 2733
NDI	3. 2734
need statement	3. 2735
negotiation	3. 2736

nest	3. 2737
network open end	3. 2738
network path	3. 2739
network video recorder	3. 2783
networking	3. 2740
neural network	3. 2741
neuron coverage	3. 2742
new source statements	3. 2743
NFR	3. 2757
NFSM	3. 2744
node letter	3. 2745
node number	3. 2746
nominal scale	3. 2747
noncompensatory decision technique	3. 2748
non-compensatory model	3. 2749
nonconformity	3. 2750
non-deliverable item	3. 2751
nondelivered source statement	3. 2752
non-deterministic system	3. 2753
nondeveloped source statement	3. 2754
nondevelopmental	3. 2755
nondimensional scaling	3. 2756
non-disclosure agreement	3. 2733
non-functional requirement	3. 2757
non-functional requirements for software	3. 2758
non-functional size	3. 2759
non-functional size measurement	3. 2744
non-human resource cost	3. 2760
nonidentifying relationship	3. 2761
nonintrinsic relationship	3. 2762
nonkey attribute	3. 2763
non-primary entity	3. 2764
nonprocedural language	3. 2765
nonprocedural programming language	3. 2766
nonprogrammable terminal	3. 1515, 3. 2767
non-repudiation	3. 2768
nontechnical requirement	3. 2769
non-terminal failure	3. 2780
non-terminal symbol	3. 2770
non-time-critical computationally intensive task	3. 2771
non-volatile memory	3. 2772
non-volatile random access memory	3. 1879

no-op	3. 1500
no-operation	3. 1500
NOR	3. 2775
norm	3. 2773
normal run response	3. 2776
normal run termination	3. 2777
normal support response	3. 2779
normalization	3. 2774
notice of revision	3. 2775
NRR	3. 2776
NRT	3. 2777
N-squared diagram	3. 2778
NSR	3. 2779
NTF	3. 2780
number of data element types	3. 2781
numeric	3. 2782
numeric character	3. 1397, 3. 1398
NVR	3. 2783
NVRAM	3. 1879

0

OA	3. 2799
OAUTH	3. 2784
object	3. 2785
object diagram	3. 2786
object flow state	3. 2787
object identifier	3. 2788
object implementation	3. 2789
object language	3. 2790
object lifeline	3. 2791
object program	3. 2792
object reference	3. 2793
object set	3. 2794
object type	3. 2795
object type set	3. 2796
objective function	3. 2797
object-oriented design	3. 2798
object-oriented systems engineering method	3. 2812
OBS	3. 2867
OD	3. 2851
ODP	3. 2813
OEM	3. 2886
OF	3. 2803

office automation	3. 2799
offset	3. 2800
off-the-shelf	3. 2801
off-the-shelf product	3. 2802
on-chip oscillator	3. 2804
on-demand scheduling	3. 2805
one-level address	3. 1409
one-time password/one-time programming	3. 2806
one-time programming	3. 2807
online compiler	3. 2121
online documentation	3. 2808
online help	3. 2809
on-screen documentation	3. 2810
onscreen information for users	3. 2811
OOSEM	3. 2812
open authentication	3. 2784
open distributed processing	3. 2813
open software foundation	3. 2814
open source software	3. 2815
open system	3. 2816
open system environment	3. 2817
open systems interconnection	3. 2818
operand	3. 2819
operating environment (software)	3. 2820
operating mode	3. 2821
operating system	3. 2822
operation	3. 2823
operation and maintenance costs	3. 2824
operation and maintenance phase	3. 2825
operation code	3. 2826
Operation constraints	3. 2827
Operation enabling system requirements	3. 2828
operation exception	3. 2829
operation field	3. 2830
operation interface	3. 2831
Operation record	3. 2832
Operation report	3. 2833
Operation strategy	3. 2834
operational availability	3. 194
operational concept	3. 2835
operational constraint	3. 2836
operational environment	3. 2837

operational failure	3. 2803
operational product	3. 2838
operational reliability	3. 2839
operational scenario	3. 2840
operational software	3. 2841
operational test and evaluation	3. 2842
operations	3. 2843
operations manual	3. 2845
operator manual	3. 2844, 3. 2845
Operator/maintainer training materials	3. 2846
OPM	3. 2847
opportunity	3. 2848
opportunity cost	3. 2849
OpsCon	3. 2835, 3. 2850
optical disc	3. 2851
optimistic duration	3. 2852
optimization analysis	3. 2853
optimizing process	3. 2854
optional	3. 2855
optional attribute	3. 2856
optional nonidentifying relationship	3. 2857
optional requirement	3. 2858
optional task	3. 2859
ordinal scale	3. 2860
Organization infrastructure	3. 2861
Organization infrastructure needs	3. 2862
Organization lessons learned	3. 2863
Organization portfolio direction and constraints	3. 2864
Organization strategic plan	3. 2865
Organization tailoring strategy	3. 2866
organizational breakdown structure	3. 2867
organizational management	3. 2868
organizational management profile	3. 2869
organizational maturity	3. 2870
organizational plan	3. 2871
Organizational policies, procedures, and assets	3. 2872
organizational policy	3. 2873
organizational process assets	3. 2874
organizational process maturity	3. 2875
Organizational process performance measures data	3. 2876
Organizational process performance measures needs	3. 2877
organizational profile	3. 2878

organizational project management maturity	3. 2879
organizational risk	3. 2880
organizational test policy	3. 4661
organizational test practice	3. 2881
organizational test process	3. 2882
organizational test specification	3. 2883
organizational test strategy	3. 2884
origin attribute	3. 2885
original equipment manufacturer	3. 2886
original source statement	3. 2887
orphan page	3. 2888
orthogonal variability model	3. 2889
OS	3. 2822
OSE	3. 2817
OSF	3. 2814
OSI	3. 2818
osmotic communication	3. 2890
OSS	3. 2815
OTE	3. 2842
OTP	3. 2806, 3. 2807
OTS	3. 2801
outcome	3. 2891
outer cardinality	3. 2892
output	3. 2893
output arc (of a transition)	3. 2894
output assertion	3. 2895
output place (of a transition)	3. 2896
output primitive	3. 2897
output product	3. 2898
outsider's viewpoint	3. 2899
outsource	3. 2900
overall project risk	3. 2901
overfitting	3. 2902
overflow exception	3. 2903
overhead operation	3. 2904
overhead time	3. 2905
overlay	3. 2906
overlay supervisor	3. 2907
overload	3. 2908
override	3. 2909
overriding property	3. 2910
OVM	3. 2889

owned attribute	3. 2911
owner of the FSM method	3. 2912
P	
P&D	3. 3268
P&P	3. 3043
PA	3. 3158
PaaS	3. 3035, 3. 3036
package	3. 2913
package diagram	3. 2914
padding	3. 2915
page	3. 2916
page breakage	3. 2917
page frame	3. 2918
page reference	3. 2919
page swapping	3. 2920
page table	3. 2921
page type letter	3. 2922
page zero	3. 2923
page-by-page reviewing	3. 2924
pager	3. 2925
paging	3. 2926
pair programming	3. 2927
pair review	3. 2928
PAM	3. 3156
parallel	3. 2929
parallel classes	3. 2930
parallel construct	3. 2931
parallel run operation	3. 2932
parameter	3. 2933
parameterized collection class	3. 2934
parameterized high-level net graph	3. 2935
parameter-value pair	3. 2936
parametric diagram	3. 2937
parametric estimating	3. 2938
parking lot	3. 2939
parking lot diagram	3. 2940
parse	3. 2941
parser	3. 2942
partial	3. 2943
partial cluster	3. 2944
participant property	3. 2945
participatory design	3. 2946

pass/fail criteria	3. 2947
passive interconnection	3. 2948
passive object	3. 2949
patch	3. 2950
path	3. 2951
path analysis	3. 2952
path condition	3. 2953
path expression	3. 2954
pathological coupling	3. 2955
pattern-sensitive fault	3. 1220, 3. 2956
payoff matrix	3. 2957
PBS	3. 3226
PC	3. 2991
PCA	3. 3002, 3. 3296
PCB	3. 3112
PCBA	3. 3114
PCO	3. 3040
PDL	3. 3279
PDM	3. 3068
PDPC	3. 3169
PDR	3. 3080
peer review	3. 2958
peer software	3. 2959
percentile	3. 2960
perceptual reference point	3. 2961
perfective maintenance	3. 2962
perfective support	3. 2963
perform integrated change control	3. 2964
perform qualitative risk analysis	3. 2965
perform quality assurance	3. 2966
perform quality control	3. 2967
perform quantitative analysis	3. 2968
perform quantitative risk analysis	3. 2969
performance	3. 2970
performance analysis	3. 2971
performance baseline	3. 2972
performance measurement baseline	3. 2973
performance reports	3. 2974
performance requirement	3. 2975
performance reviews	3. 2976
performance specification	3. 2977
performed process	3. 2978

periodic task	3. 2979
peripheral equipment	3. 2980
permanence	3. 2981
permission	3. 2982
perpetual license	3. 2983
persistence	3. 2984
persistence schema	3. 2985
persistent object	3. 2986
persistent storage	3. 2987
persistent URI	3. 2988
person hours per functional size unit	3. 2989
persona	3. 2990
personal computer	3. 2991
personal computer (PC)	3. 2992
personal data	3. 2993
personal protective equipment	3. 2994
personally identifiable information	3. 2995
personally identifiable information principal	3. 1204
perspective-based reading	3. 2996
PERT	3. 3280, 3. 3281
pessimistic duration	3. 2997
PESTEL	3. 3045
Petri net	3. 2998
PG	3. 3273
phase	3. 2999
phase gate	3. 3000
physical architecture	3. 3001
physical configuration audit	3. 3002
physical configuration audit (PCA)	3. 3003
physical interface	3. 3004
physical requirement	3. 3005
physical source statement	3. 3006
PI	3. 3232
PICS	3. 3385
PII	3. 2995
PIPEDA	3. 3007
pipeline	3. 3008
PJM	3. 3339
place	3. 3009
plan	3. 3010
plan communications	3. 3011
plan communications management	3. 3011

plan cost management	3. 3012
plan procurement	3. 3013
plan procurement management	3. 3013
plan quality	3. 3014
plan quality management	3. 3014
plan resource management	3. 3015
plan risk management	3. 3016
plan risk responses	3. 3017
plan schedule management	3. 3018
plan scope management	3. 3019
plan stakeholder engagement	3. 3020
plan stakeholder management	3. 3021
plan standard	3. 3022
planned process	3. 3023
planned value	3. 3024, 3. 3025
planning	3. 3026
planning horizon	3. 3027
planning performance domain	3. 3028
planning poker	3. 3029
planning process group	3. 3030
plant	3. 3031
plastic leaded chip carrier	3. 3032
plastic leaded chip carrier (PLCC)	3. 3033
platform	3. 3034
platform as a service	3. 3035
platform as a service (PaaS)	3. 3036
playback	3. 3708
PLCC	3. 3032, 3. 3033
PM	3. 3049, 3. 3339
PMB	3. 2973
PMO	3. 3332
PMP	3. 3333
PN	3. 2998
PNML high-level net document	3. 3037
PNML symmetric net document	3. 3038
point design	3. 3039
point of control and observation	3. 3040
point.	3. 3041
pointer	3. 3042
policies and procedures	3. 3043
policy-setting behaviour	3. 3044
political, economic, social, technological, environmental, and legal	3. 3045

pop-up	3. 3046
port	3. 3047
portability	3. 3048
portfolio management	3. 3049
Portfolio management plan	3. 3050
Portfolio management record	3. 3051
Portfolio management report	3. 3052
portfolio.	3. 3053
positive consequence	3. 1343
post project evaluation	3. 3054
postamble breakpoint	3. 1653, 3. 3055
post-closure activities	3. 3056
post-compile time	3. 3057
postcondition	3. 3058
post-development review	3. 4474
post-development review	3. 3059, 3. 4473
post-implementation review	3. 4474
post-implementation review	3. 4473
postmortem dump	3. 3060
post-probe	3. 3061
postprocessor	3. 3062
post-tested iteration	3. 3063, 3. 4888
PPE	3. 2994
PR	3. 3140
practice	3. 3064
practices	3. 3065
practitioner	3. 3066
pragma	3. 3391
preamble breakpoint	3. 3371
precautionary principle	3. 3067
precedence diagramming method	3. 3068
precedence relationship	3. 3069
precompiler	3. 3070
precondition	3. 3071
predicate	3. 3072
predicate data use	3. 3073
prediction.	3. 3074
predictive approach	3. 3075
predictive life cycle	3. 3076
predictive metric	3. 3077
predictive metric value	3. 3078
preferential logic	3. 1435

preferred logic	3. 1435
preliminary design	3. 3079
preliminary design review	3. 3080
Preliminary interface definition	3. 3081
Preliminary life cycle concepts	3. 3082
Preliminary MOE data	3. 3083
Preliminary MOE needs	3. 3084
Preliminary TPM data	3. 3085
Preliminary TPM needs	3. 3086
Preliminary validation criteria	3. 3087
preparation time	3. 3088
pre-probe	3. 3089
pre-processing	3. 3090
preprocessor	3. 3091
prescription	3. 3092
presentable	3. 3093
presentation device	3. 3094
prestore	3. 3095
pretested iteration	3. 3096
prettyprinting	3. 3097
preventive action	3. 3098
preventive maintenance	3. 3099
preventive support	3. 3100
previously developed software	3. 3101
primary data	3. 3102
primary entitlement	3. 3103
primary entitlement schema	3. 3104
primary entity-type	3. 3105
primary information structure	3. 3106
primary intent	3. 3107
primary key	3. 3108
primary user	3. 3109
primitive	3. 3110
primitive type	3. 362, 3. 3111
principal	3. 1204
printed circuit board	3. 3112
printed circuit board (PCB)	3. 3113
printed circuit board assembly	3. 3114
printed circuit board assembly (PCBA)	3. 3115
printed documentation	3. 3116
printed information for users	3. 3117
prioritization matrices	3. 3118

prioritization scheme	3. 3119
priority interrupt	3. 3120
priority inversion	3. 3121
privacy capabilities	3. 3122
privacy control	3. 3123
privacy framework	3. 3124
privacy principles	3. 3125
privacy requirement	3. 3126
privacy risk	3. 3127
private	3. 3128
private type	3. 3129
privileged instruction	3. 3130
privileged state	3. 1741, 3. 3131, 3. 4390
PRM	3. 3204
proactive approach	3. 3132
probabilistic estimating	3. 3133
probabilistic system	3. 3134
probability	3. 3135
probability and impact matrix	3. 3136
probe	3. 3137
problem	3. 3138
problem definition	3. 3139
problem description	3. 3139
problem report	3. 3140
problem state	3. 3141
problem-oriented language	3. 3142
procedural cohesion	3. 3143
procedural language	3. 3144
procedural programming language	3. 3145
Procedure	3. 3146
procedure testing	3. 3147
procedure-oriented language	3. 3144
process	3. 3148
process action plan	3. 3149
process action team	3. 3150
process analysis	3. 3151
process architect	3. 3152
process architecture	3. 3153
process area	3. 3154
process assessment	3. 3155
process assessment model	3. 3156
process asset library	3. 3157

process attribute	3. 3158
process attribute outcome	3. 3159
process attribute rating	3. 3160
process capability	3. 3161
process capability baseline	3. 3162
process capability determination	3. 3163
process capability level	3. 3164
process capability rating	3. 3165
process category	3. 3166
process component	3. 3167
process context	3. 3168
process decision program chart	3. 3169
process definition	3. 3170
process description	3. 3171
process development	3. 3172
process dimension	3. 3173
process evidence	3. 3174
process group	3. 3175
process improvement	3. 3176
process improvement action	3. 3177
process improvement objective	3. 3178
process improvement plan	3. 3179
process improvement programme	3. 3180
process improvement project	3. 3181
process improvement support element	3. 3182
process infrastructure	3. 3183
process instance	3. 3184
process management	3. 3185
process maturity	3. 3186
process measurement	3. 3187
process measurement framework	3. 3188
process metric	3. 3189
process outcome	3. 3190
process owner	3. 3191
process performance	3. 3192
process performance baseline	3. 3193
process performance indicator	3. 3194
process profile	3. 3195
process purpose	3. 3196
process quality	3. 3197
process quality attribute	3. 3198
process quality characteristic	3. 3199

process quality determination	3. 3200
process quality dimension	3. 3201
process quality indicator	3. 3202
process quality level	3. 3203
process reference model	3. 3204
process risk determination	3. 3205
process standard	3. 3206
process tailoring	3. 3207
process trustworthiness	3. 3208
process view	3. 3209
process-evidence element	3. 3210
processing logic	3. 3211
processor	3. 3212
process-related risk	3. 3213
procurement audit	3. 3214
procurement documents	3. 3215
procurement management plan	3. 3216
procurement performance review	3. 3217
procurement statement of work	3. 3218
producer	3. 4399
producer object	3. 3219
product analysis	3. 3220
product asset instance	3. 3221
product authority	3. 3222
product backlog	3. 3223
product baseline	3. 3224, 3. 3225
Product breakdown structure	3. 3226
product certification	3. 3227
product configuration baseline	3. 3224, 3. 3225
product configuration identification	3. 3228
product description	3. 3229
product engineering	3. 3230
product identification	3. 3231
Product Improve	3. 3232
product instances	3. 3233
product library	3. 3234
product life cycle	3. 3235
product line	3. 3236
product line adoption plan	3. 3237
product line adoption scenario	3. 3238
product line architecture	3. 3239
product line configuration	3. 3240

product line configuration delta	3. 3241
product line configuration management	3. 3242
product line engineering factory	3. 3243
product line engineering factory configurator	3. 3244
product line engineering factory development environment	3. 3245
product line institutionalization	3. 3246
product line platform	3. 3247
product line product family	3. 3248
product line reference model	3. 3249
product line scoping	3. 3250
product line test strategy	3. 3251
product line-configuration item	3. 3252
product management	3. 3253
product metric	3. 3254
product owner	3. 3255
product quality	3. 3256
product requirement	3. 3257
product risk	3. 3258
product roadmap	3. 3259
product safety label	3. 3260
product scope	3. 3261
product scope description	3. 3262
product scoping	3. 3263
product specification	3. 3264
product standard	3. 3265
product support	3. 3266
production	3. 3267
production and deployment	3. 3268
production configuration	3. 3224, 3. 3225
production library	3. 3269
production plan	3. 3270
production rate	3. 3271
profile	3. 3272
profile group	3. 3273
program	3. 3274
program architecture	3. 3275
program block	3. 3276
program construct	3. 3277
program counter	3. 3278
program design language	3. 3279
Program Evaluation and Review Technique	3. 3280
Program Evaluation Review Technique	3. 3281

program extension	3. 3282
program instruction	3. 3283
program instrumentation	3. 3284
program library	3. 3285, 3. 4084
program listing	3. 3286
program maintenance manual	3. 3287
program management	3. 3288
program mutation	3. 3289
program network chart	3. 3290
program protection	3. 3291
program specification	3. 3292
program status word	3. 3293
program structure chart	3. 4331
program support library	3. 4064
program synthesis	3. 3294
programmable breakpoint	3. 3295
programmable counter array	3. 3296
programmable logic device	3. 3297
programmable logic device software	3. 3298
programmable pulse generator	3. 3299
programmable read-only memory	3. 3300
programmable reload timer	3. 3301
programmable terminal	3. 3302
programmable terminal intelligent terminal	3. 3303
programmatic reference point	3. 3304
programmer manual	3. 3305
programming	3. 3306
programming language	3. 3307
programming support environment	3. 3308
programming system	3. 3309
program-sensitive fault	3. 3310
progressive elaboration	3. 3311
prohibition	3. 3312
project	3. 3313
project balance	3. 3314
project brief	3. 3315
project calendar	3. 3316
project communications management	3. 3317
project control	3. 3318
project cost management	3. 3319
project file	3. 3320
project file project notebook	3. 3321

project function point count	3. 3322
project functional size	3. 3323
project funding requirements	3. 3324
project governance	3. 3325
project initiation	3. 3326
project integration management	3. 3327
project lead	3. 3328
project library	3. 4064
project life cycle	3. 3329
project management (PM)	3. 3330
project management knowledge area	3. 3331
project management office	3. 3332
project management plan	3. 3333
project management process group	3. 3334
project management software	3. 3335
project management staff	3. 3336
project management system	3. 3337
project management team	3. 3338
project manager	3. 3339
project performance	3. 3340
project phase	3. 3341
project plan	3. 3342
project planning	3. 3343
project portfolio	3. 3344
project procurement management	3. 3345
project quality management	3. 3346
project resource constraint	3. 3347
project resource management	3. 3348
project risk	3. 3349
project risk management	3. 3350
project risk profile	3. 3351
project schedule	3. 3352
project schedule management	3. 3353
project schedule network diagram	3. 3354, 3. 3355
project scope	3. 3356
project scope management	3. 3357
project scope statement	3. 3358
project specification	3. 3359
project stakeholder management	3. 3360
project status report	3. 3361
project tailoring strategy	3. 3362
project team	3. 3363

project team directory	3. 3364
project time management	3. 3365
project vision statement	3. 3366
project work performance domain	3. 3367
project-based organizations	3. 3368
projectized organization	3. 3369
project's defined software process (PDSP)	3. 3370
prolog breakpoint	3. 3371
prologue breakpoint	3. 3372
PROM	3. 3300
prompt	3. 3373
proof of concept	3. 3374
proof of correctness	3. 3375
property to quantify	3. 3376
property-of-interest	3. 3377
proposal	3. 3378
proposal evaluation techniques	3. 3379
proposed change	3. 3380
proposition	3. 3381
protected	3. 3382
protection exception	3. 3383
protocol	3. 3384
Protocol Implementation Conformance Statement	3. 3385
protocol object	3. 3386
prototype	3. 3387
prototyping	3. 3388
provide instruments activity	3. 3389
provision	3. 3390
PRT	3. 3301
pseudo code	3. 3392
pseudo instruction	3. 3391
pseudocode	3. 3392
pseudo-op	3. 3391
pseudo-operation	3. 3391
pseudo-oracle	3. 3393
pseudo-state	3. 3394
pseudostatic random access memory	3. 3395
PSRAM	3. 3395
PSS	3. 3006
PSW	3. 3293
psychometrics	3. 3396
PT	3. 3066

public	3. 3397
public blockchain	3. 3398
public cloud	3. 3399
published model	3. 3400
publishing pipeline	3. 3401
pulse width modulation	3. 3402
purchaser	3. 1117
purpose of the count	3. 3403
p-use	3. 3073
pushdown storage	3. 3404
PV	3. 3024, 3. 3025
PWM	3. 3402

Q

QA	3. 3420
QAP	3. 3422
qauality assurance report	3. 3405
QC	3. 3428
QFD	3. 3434
QM	3. 3437, 3. 3445
QME	3. 3446
qualification	3. 3406
qualification (system)	3. 3407
qualification body	3. 3408
qualification limit	3. 3409
qualification margin	3. 3410
qualification requirement	3. 3411
qualification scheme	3. 3412
qualification testing	3. 3413
qualification/qualified (person)	3. 3414
qualified personnel	3. 3415
qualifier	3. 3416
qualitative risk analysis	3. 3417
quality	3. 3418
quality analysis	3. 3419
quality assurance	3. 3420
quality assurance evaluation report	3. 3421
quality assurance plan	3. 3422
quality assurance record	3. 3423
quality attribute	3. 3424
quality characteristic	3. 3425
quality characteristic (of software product or system)	3. 3426
quality checklist	3. 3427

quality control	3. 3428
quality control measurements	3. 3429
quality evaluation	3. 3430
quality factor	3. 3431
quality factor sample	3. 3432
quality factor value	3. 3433
quality function deployment	3. 3434
quality in use (measure)	3. 3435
quality in use measure	3. 3436
quality management	3. 3437
quality management and control tools	3. 3438
quality management corrective action	3. 3439
quality management evaluation report	3. 3440
quality management guideline	3. 3441
quality management plan	3. 3442
quality management record	3. 3443
quality management report	3. 3444
quality measure	3. 3445
quality measure element	3. 3446
quality measure on external property	3. 3447
quality measure on internal property	3. 3448
quality metric	3. 3449
quality model	3. 3450
quality of service	3. 3451
quality policy	3. 3452
quality property	3. 3453
quality rating module	3. 3454
quality record report	3. 3455
quality report	3. 3456
quality requirement	3. 3457
quality subcharacteristic	3. 3458
quality subfactor	3. 3459
quantitative control	3. 3460
quantitative risk analysis	3. 3461
quantum computing	3. 3462
query language	3. 3463
queue	3. 3464
quiescing	3. 3465

R

RAM	3. 3468, 3. 3676, 3. 3677
random failure	3. 3466
random testing	3. 3467

random-access memory	3. 3468
rapid prototyping	3. 3469
rate-monotonic algorithm	3. 3470
rating	3. 3471
rating interval	3. 3472
rating level	3. 3473
ratio scale	3. 3474
rational unified process	3. 3475
RBS	3. 3659, 3. 3727
RDBMS	3. 3549
reachability graph	3. 3476
reachability set	3. 3477
reachable marking	3. 3478
reactivation	3. 3479
reactive approach	3. 3480
reactiveness	3. 3481
read	3. 3483
read arc	3. 3482
read type	3. 3483
readability	3. 3484
reader note	3. 3485
reading reference	3. 3486
read-only	3. 3487
read-only memory	3. 3488
ready	3. 3489
real address	3. 3490
real storage	3. 3491
real type	3. 3492
realization	3. 3493
real-time	3. 3494
real-time clock	3. 3495
real-time operating system	3. 3496
real-time operating system (RTOS)	3. 3497
real-time scheduling theory	3. 3498
real-world object	3. 3499
reasonably foreseeable misuse	3. 3500
reasoning technique	3. 3501
recall	3. 3502
receptacle	3. 3503
recommendation	3. 3504
record element type	3. 3505
record element type (RET)	3. 3506

record type	3. 3507
records management system	3. 3508
recoverability	3. 3509
recursion	3. 3510
recursive	3. 3511
recursive routine	3. 3512
redundancy	3. 3513
reengineering	3. 3514
reenterable	3. 3515
reentrant	3. 3515
reentry point	3. 3516
refactor	3. 3517
reference	3. 3518
reference architecture	3. 3519
reference body of knowledge	3. 3520
reference framework	3. 3521
reference FSM method	3. 3522
reference information	3. 3523
reference mode	3. 3524
reference node	3. 3525
reference place	3. 3526
reference point	3. 3527
reference transition	3. 3528
reference user requirement collection	3. 3529
reference user requirements	3. 3530, 3. 3788
referential integrity	3. 3531
refinement	3. 3532
reflective construct	3. 3533
reflexive	3. 3534
reflexive ancestor (of a class)	3. 3535
refresh	3. 3536
Regex	3. 3544
regid	3. 3539
register	3. 3537
register bank	3. 3538
registration identifier	3. 3539
registry	3. 3540
regression	3. 3541
regression analysis	3. 3542
regression test	3. 3543
regular expression	3. 3544
regular expression (Regex)	3. 3545

regulatory context	3. 3546
reinforcement learning	3. 3547
relation	3. 3548
relational database management system	3. 3549
relationship	3. 3550
relationship instance	3. 3551
relationship manager	3. 3552
relationship name	3. 3553
relationship type	3. 3554
relative address	3. 3555
relative chain frequency	3. 3556
relative estimating	3. 3557
relative loader	3. 3558, 3. 3583
relative time	3. 3559
release	3. 3560
release backlog	3. 3561
release map	3. 3562
release package	3. 3563
release plan	3. 3564
release planning	3. 3565
relevant stakeholder	3. 3566
reliability	3. 3567
reliability assessment	3. 3568
reliability class	3. 3569
reliability data	3. 3570
reliability growth	3. 3571
reliability model	3. 3572
reliability numeric	3. 3573
reliability testing	3. 3574
relocatable	3. 3575
relocatable address	3. 3576
relocatable code	3. 3577
relocatable machine code	3. 3578
relocate	3. 3579
relocating address	3. 3580
relocating assembler	3. 3581
relocating code	3. 3582
relocating loader	3. 3583
relocation dictionary	3. 3584
relocation factor	3. 3585
relocation transparency	3. 3586
relocator	3. 3587

remote batch entry	3. 3589
remote job entry	3. 3588, 3. 3589
remote terminal emulator	3. 3590
repair	3. 3591
reparent	3. 3592
repeatability	3. 3593
repeatability (of results of measurement)	3. 3594
repetitive addressing	3. 3595
replaceability	3. 3596
replay	3. 3597, 3. 3708
replication	3. 3598
replication schema	3. 3599
replication transparency	3. 3600
repo bloat	3. 3601
repo surgery	3. 3602
report	3. 3603
report performance	3. 3604
report standard	3. 3605
repository	3. 3606
representation	3. 3607
representation property	3. 3608
representation standard	3. 3609
reproducibility	3. 3610
reproducibility (of results of measurement)	3. 3611
request	3. 3612
request for change	3. 3613, 3. 3717
request for information	3. 3718
request for proposal	3. 3614
request for quotation	3. 3615
request form	3. 3616
requested change	3. 3617
required inputs	3. 3618
required outputs	3. 3619
requirement	3. 3620
requirement review	3. 3621
requirement standard	3. 3622
requirements allocation	3. 3623
requirements analysis	3. 3624
requirements attributes	3. 3625
requirements derivation	3. 3626
requirements document	3. 3627
requirements documentation	3. 3628

requirements elicitation	3. 3629
requirements engineering	3. 3630
requirements flow-down	3. 3631
requirements management	3. 3632
requirements management plan	3. 3633
requirements partitioning	3. 3634
requirements phase	3. 3635
requirements review	3. 3621
requirements specification	3. 3636
requirements specification language	3. 3637
requirements test matrix	3. 4686
requirements traceability	3. 3638
requirements traceability matrix	3. 3639
requirements traceability tool	3. 3640
requirements validation	3. 3641
requirements verification	3. 3642
Requirements Verification and Traceability Matrix	3. 3643
requirements verification table	3. 4686
requirements-based testing	3. 3644
rescue point	3. 3682
reserve	3. 3645, 3. 3647
reserve analysis	3. 3646
reserved word	3. 3648
reset	3. 3649
reset arc	3. 3650
resident control program	3. 3651
residual control	3. 3652
residual risk	3. 3653
resilience	3. 3654
resiliency	3. 3655
resistance	3. 3656
resource	3. 3657
resource allocation	3. 3658
resource breakdown structure	3. 3659
resource calendar	3. 3660
resource histogram	3. 3661
resource leveling	3. 3662
resource management	3. 3663
resource manger	3. 3664
resource monitor task	3. 3665
resource pooling	3. 3666
resource smoothing	3. 3667

resource utilization	3. 3668
resource utilization measurement	3. 3669
resource utilization measurement (RUM)	3. 3670
resource-limited schedule	3. 3671
respecialize	3. 3672
responding object	3. 3673
response time	3. 3674
responsibility	3. 3675
responsibility assignment matrix (RAM)	3. 3676
responsibility assignment matrix (RAM)	3. 3677
responsive web design	3. 3678
responsiveness	3. 3679
restart	3. 3680
restart point	3. 3681, 3. 3682
result	3. 3683
RET	3. 3505, 3. 3506
retainage	3. 3684
retesting	3. 3685, 3. 3686
retirement	3. 3687
retirement phase	3. 3688
retrospective	3. 3689
retrospective trace	3. 3690
return	3. 3691
return code	3. 3692
Return on investment	3. 3693
return on investment (ROI)	3. 3694
return value	3. 3695
reusability	3. 3696
reusable	3. 3697
reusable product	3. 3698
reusable software product	3. 3699
reuse	3. 3700
reused source statement	3. 3701
revealed requirement	3. 3702
revenue alternative	3. 3703
reverse engineering	3. 3704
reverse execution	3. 3705, 3. 3708
reversibility	3. 3706
reversible execution	3. 3707, 3. 3708
review	3. 3709
review folder	3. 3710
review output	3. 3712, 3. 3713

review/audit output	3. 3711, 3. 3712, 3. 3713
revocation	3. 3714
reward hacking	3. 3715
rework	3. 3716
RFC	3. 3613, 3. 3717
RFI	3. 3718
RFP	3. 3614
RFQ	3. 3615
right	3. 3719
ripple clock	3. 3720
risk acceptance	3. 3721
risk action request	3. 3722
risk analysis	3. 3723
risk appetite	3. 3724
risk audit	3. 3725
risk avoidance	3. 3726
risk breakdown structure	3. 3727
risk category	3. 3728
risk criteria	3. 3729
risk enhancement	3. 3730
risk escalation	3. 3731
risk exploiting	3. 3732
risk exposure	3. 3733
risk factor	3. 3734
risk handling	3. 3735
risk identification	3. 3736
risk leverage factor	3. 3737
risk management	3. 3738
risk management plan	3. 3739
risk management process	3. 3740
risk metric	3. 3741
risk mitigation	3. 3742
risk monitoring and control	3. 3743
risk owner	3. 3744
risk profile	3. 3745
risk reassessment	3. 3746
risk reduction	3. 3747
risk reduction measure	3. 3748
risk register	3. 3749
risk report	3. 3750
risk review	3. 3751
risk sharing	3. 3752

risk source	3. 3753
risk threshold	3. 3754
risk tolerance	3. 3755
risk transfer	3. 3756
risk transference	3. 3757
risk treatment	3. 3758
risk treatment measure	3. 3759
risk trigger	3. 3760
risk urgency assessment	3. 3761
risk-adjusted backlog	3. 3762
risk-based testing	3. 3763
risk-free MARR	3. 3764
RJE	3. 3588, 3. 3589
rlf	3. 3737
RM	3. 3552, 3. 3738
roadmap	3. 3765
robot	3. 3766
robustness	3. 3767
ROI	3. 3693, 3. 3694
role	3. 3768
role name	3. 3769
role-based reviewing	3. 3770
role-playing session	3. 3771
roll in	3. 3772
roll out	3. 3773
rolling wave planning	3. 3774
ROM	3. 3488
root cause	3. 3775
root cause analysis	3. 3776
root compiler	3. 3777
routine	3. 3778
RT	3. 3674, 3. 3784
RTE	3. 3590
RTM	3. 3639
RTOS	3. 3496, 3. 3497
rule	3. 3779
rule-based language	3. 3780
RUM	3. 3669, 3. 3670
run	3. 3781
run mode	3. 3782
run stream	3. 3783
run time	3. 3784, 3. 3785

running time	3. 3785
runtime environment	3. 3786
runtime platform	3. 3787
RUP	3. 3475
RUR	3. 3530, 3. 3788
RVTM	3. 3643
RWD	3. 3678

S

<sevo> data failure (DF)	3. 1162
SaaS	3. 4029
safe integration	3. 3789
safety	3. 3790
safety criteria	3. 3791
safety critical function	3. 3792
safety demonstration	3. 3793
safety injection signal	3. 3794
safety integrity requirement	3. 3795
safety life cycle	3. 3796
safety note	3. 3797
safety requirement	3. 3798
safety sign	3. 3799
safety validation	3. 3800
safety-critical software	3. 3801
safety-critical system	3. 3802
SAIV	3. 3824
SAM	3. 4031
SAM program scope	3. 3803
SAM tool	3. 3804, 3. 3805, 3. 3806
SAMP	3. 4312
sample instance diagram	3. 3807
sample instance table	3. 3808
satisficing	3. 3809
SBI	3. 3921
SBS	3. 4456
scaffolding	3. 3810
scalability	3. 3811
scalability testing	3. 3812
scalar	3. 3813
scalar property	3. 3816
scalar-valued class	3. 3814
scalar-valued property	3. 3815, 3. 3816
scale	3. 3817

scaliability	3. 3818
scatter diagram	3. 3819
scenario	3. 3820
scenario script	3. 3821
scenario testing	3. 3822
scenario-based reviewing	3. 3823
schedule as independent variable	3. 3824
schedule baseline	3. 3825
schedule compression	3. 3826
schedule data	3. 3827
schedule development	3. 3828
schedule forecast	3. 3829
schedule management plan	3. 3830
schedule model	3. 3831
schedule network analysis	3. 3832
schedule performance index	3. 3833
schedule performance index (SPI)	3. 3834
schedule variance	3. 3835
schedule variance (SV)	3. 3836
scheduled finish date	3. 3837
scheduled start date	3. 3838
scheduler	3. 3839
schema	3. 3840
Schematron	3. 3841
SCI	3. 3922, 3. 4042
SCM	3. 3842, 3. 4043
SCMP	3. 3844
SCMPI	3. 3843
SCN	3. 3845, 3. 4199
scope baseline	3. 3846
scope creep	3. 3847
scope management plan	3. 3848
scope of the FSM	3. 3849
screen capture	3. 3850, 3. 3851
screen dump	3. 3851
scripted testing	3. 3852
SCRM	3. 3853
scrum	3. 3854
scrum lead	3. 3855
scrum report	3. 3856
scrum team	3. 3857
SCS	3. 3802

SCU	3. 4094
S-curve	3. 3858
SDD	3. 4052
SDF	3. 4060, 3. 4061
SDLC	3. 3859
SDP	3. 4066
SDR	3. 3860
SDRAM	3. 4431
SE	3. 3971
search algorithm	3. 3861
second normal form	3. 3862
secondary data	3. 3863
secondary user	3. 3864
secondary window	3. 3865
second-generation language	3. 306
secure hash algorithm	3. 3866
secure processing requirement	3. 3867
security	3. 3868
security accreditation	3. 3869
security and intellectual property scheme	3. 3870
security authority	3. 3871
security branch	3. 3872
security control	3. 3873
security domain	3. 3874
security failure	3. 3875
security interaction policy	3. 3876
security kernel	3. 3877
security level	3. 3878
security policy	3. 3879
security responsibility	3. 3880
security testing	3. 3881
SEE	3. 3886
SEE service	3. 3884
SEE service acquirer	3. 3882
SEE service supplier	3. 3883
seed	3. 3885
SEG	3. 4071
segment	3. 3887
segmented executor	3. 3888
SEI	3. 3925
selective dump	3. 3889
selective trace	3. 3890

self-assembly product	3. 3891
self-contained	3. 3892
self-descriptiveness	3. 3893
self-documented	3. 3894
self-learning system	3. 3895
self-provisioning	3. 3896
self-relative address	3. 3897
seller	3. 3898
seller proposals	3. 3899
semantic agreement	3. 3900
semantic error	3. 3901
semantic label	3. 3902
semantic transfer language	3. 3903
Semantic Transfer Language (STL)	3. 3904
semantic variation point	3. 3905
semantics	3. 3906
semaphore	3. 3907
SEMDM	3. 3908
semi-formal verification	3. 3909
SEMP	3. 3910, 3. 4518
sensitivity analysis	3. 3911
sentiment	3. 3912
SEP	3. 3914, 3. 4518
separate documentation	3. 3913
SEPG	3. 4072
sequence diagram	3. 3915
sequential	3. 3916
sequential clustering	3. 3917
sequential cohesion	3. 3918
sequential construct	3. 3924
sequential processes	3. 3919
serial	3. 3920
serial bus interface	3. 3921
serial communication interface	3. 3922
serial construct	3. 3923, 3. 3924
serial expansion interface	3. 3925
serial peripheral interface	3. 3926
servant leadership	3. 3927
server object	3. 3928
server-side	3. 3929
service agreement	3. 3930
service alternative	3. 3931

service catalogue	3. 3932
service change request	3. 3933
service compatibility	3. 3934
service component	3. 3935
service delivery policy	3. 3936
service delivery profile	3. 3937
service design	3. 3938
service desk	3. 3939
service export	3. 3940
service import	3. 3941
service level agreement	3. 3942
service level agreement information completeness	3. 3943
service level agreement satisfaction	3. 3944
service level objective	3. 3945
service maintainability	3. 3946
service management	3. 3947
service management system	3. 3948, 3. 3950
service management system (SMS)	3. 3949
service manager (SM)	3. 3951
service measurability	3. 3952
service offer	3. 3953
service performance efficiency	3. 3954
service primitive	3. 3955
service provision system	3. 3956
service provisionability	3. 3957
service qualitative objective	3. 3958
service quality characteristic	3. 3959
service reliability	3. 3960
service request	3. 3961
service requirement	3. 3962
service security	3. 3963
service supportiveness	3. 3964
service usability	3. 3965
service-based license	3. 4370, 3. 4371
services (修改为 service)	3. 3966
session component	3. 3967
settlement	3. 3968
set-up time	3. 3969
severity	3. 3970
SF	3. 3837, 3. 3972, 3. 4251
SFR	3. 4189
SGML	3. 3973

SGRAM	3. 4432
SHA	3. 3866
shadow class	3. 3974
shared asset	3. 3975
shared asset superset	3. 3976
shell	3. 3977
short message service	3. 3950
should-cost estimate	3. 3978
shrink small outline package	3. 3979
side effect	3. 3980
sign change coverage	3. 3981
signal	3. 3982
signal interface	3. 3983
signal interface signature	3. 3984
signature	3. 3985
signpost	3. 3986
sign-sign coverage	3. 3987
SILS	3. 4080
SIM	3. 4480
simple buffering	3. 3988
simple knowledge organization system	3. 3989
simplicity	3. 3990
simulation	3. 3991
simulation component	3. 3992
simultaneous	3. 3993
simultaneous recursion	3. 3994
single boot	3. 3995
single chip microcomputer	3. 2608
single inheritance	3. 3996
single instance approach	3. 3997
single-address instruction	3. 3998
single-hit decision table	3. 3999
single-level encoding	3. 4000
single-operand instruction	3. 4001
single-point estimating	3. 4002
single-step execution	3. 4003, 3. 4004
single-step operation	3. 4004
single-valued	3. 4005
single-valued property	3. 4006
SIS	3. 3794
site map	3. 4007
sizing	3. 4008

skeleton	3. 4009
skill	3. 4010
SKOS	3. 3989
SKU	3. 4297
SLA	3. 3942
SLA information completeness	3. 3943
SLA satisfaction	3. 3944
slack	3. 1882
slave state	3. 3141
SLC	3. 4088
sleep mode	3. 4247
slice	3. 4011
SLO	3. 3945
SLOC	3. 4179
slush	3. 4012
small and medium enterprise	3. 4013
small outline integrated circuit	3. 4014
small outline package	3. 4015, 3. 4241
SMART	3. 4197
SME	3. 4013, 3. 4358
SMS	3. 3948, 3. 3949, 3. 3950
SNAP category	3. 4093
SNAP sub-category	3. 4096
snapshot dump	3. 4016
snoop	3. 4017
SoC	3. 4489
societal and ethical risk mitigation	3. 4018
society	3. 4019
soft copy image	3. 4020
soft error	3. 4021
soft failure	3. 4022
soft logic	3. 1435
software	3. 4167
software acquisition	3. 4023
software acquisition process	3. 4024
software agent	3. 4025
software and systems product line	3. 4026
software architecture	3. 4027
software artifact	3. 4028
software as a service	3. 4029
software asset	3. 4030
software asset management	3. 4031

software asset management tool	3. 3805, 3. 3806
software baseline	3. 4032
software baseline audit	3. 4033
software baseline library	3. 4034
software behavior concepts	3. 4035
software build	3. 4036
software capability evaluation	3. 4037
software characteristic	3. 4038
software component	3. 4039
software configuration	3. 4040
software configuration control board	3. 4041
software configuration item	3. 853, 3. 4042
software configuration management	3. 4043
software configuration management item	3. 4044
software configuration management plan	3. 3844
Software Configuration Management Planned Information	3. 3843
software consumer	3. 4045
software correction	3. 4046
software creator	3. 4047
software database	3. 4048
software design	3. 4049
software design audit	3. 4050
software design concept	3. 4051
software design description	3. 4052
software design notation	3. 4053, 3. 4054
software design representation	3. 4054
software design verification	3. 4055
software developer	3. 4056
software development cost	3. 4057
software development cycle	3. 4058
software development environment	3. 4059
software development file	3. 4060, 3. 4061
software development folder	3. 4061
software development income	3. 4062
software development library	3. 4063, 3. 4064
software development method	3. 4065
software development notebook	3. 4061
software development plan	3. 4066
software development process	3. 4067
software diversity	3. 4068
software element	3. 4069
Software Engineering	3. 3971

software engineering economics	3. 4070
Software Engineering Environment	3. 3886
Software Engineering Environment Service	3. 3884
software engineering group	3. 4071
software engineering process group	3. 4072
software entitlement	3. 4073
software entitlement reconciliation	3. 4074
software entitlement schema	3. 1630
software failure	3. 4075
software feature	3. 4076
software function	3. 4077
software hazard	3. 4078
software identification tag	3. 4079
software in the loop simulation	3. 4080
software integration	3. 4081
software item	3. 4082
software library	3. 4083, 3. 4084
software license	3. 4085
software licensee	3. 4086
software licensor	3. 4087
software life cycle	3. 4088
software maintenance	3. 4089
software non-functional assessment process (SNAP) category	3. 4090
Software Non-functional Assessment Process (SNAP) point	3. 4091
Software Non-functional Assessment Process (SNAP) sub-category	3. 4092
software non-functional assessment process category	3. 4093
software non-functional assessment process counting unit	3. 4094
software non-functional assessment process point	3. 4095
Software Non-functional Assessment Process sub-category	3. 4096
software package	3. 4097
software packager	3. 4098
software piracy	3. 4099
software plans	3. 4100
software process	3. 4101
software process assessment	3. 4102
software process description	3. 4103
software process element	3. 4104
software process improvement plan	3. 4105
software process improvement proposal	3. 4106
software process maturity	3. 4107
software process-related documents	3. 4108
software product	3. 4109

software product evaluation	3. 4110
software project	3. 4111
software project life cycle	3. 4112
software project life cycle (SPLC)	3. 4113
software project life cycle model	3. 4114
software project life cycle model (SPLCM)	3. 4115
software project life cycle process	3. 4116
software project life cycle process (SPLCP)	3. 4117
Software Project Management Planned Information	3. 4118
software quality	3. 4119
software quality assurance	3. 4120
software quality characteristic	3. 4121
software quality control	3. 4122
software quality control (SQC)	3. 4123
software quality goal	3. 4124
software quality in use	3. 4125
software quality management	3. 4126
software quality measure	3. 4127
software quality metric	3. 4128
software quality requirement	3. 4129
software release management	3. 4130
software reliability	3. 4131
software reliability engineering	3. 4132
software reliability management	3. 4133
software repository	3. 4134
software requirement	3. 4135
software requirement review	3. 4136
software requirements analysis	3. 4137
software requirements engineering	3. 4138
software requirements management	3. 4139
software requirements phase	3. 4140
software requirements review (SRR)	3. 4141
software requirements specification	3. 4142
software requirements verification	3. 4143
software safety	3. 4144
software safety program	3. 4145
software security	3. 4146
software service	3. 4147
software specification review	3. 4148
software support	3. 4149
software sustainment	3. 4150
software system	3. 4151

software system element	3. 4152
software test environment	3. 4153
software test incident	3. 4154
software testing	3. 4155
software tool	3. 4156
software unit	3. 4157
software usage	3. 4158
software user documentation	3. 4159
software version ID	3. 4160
software version ID (software version identification)	3. 4161
software work product	3. 4162
software/system element	3. 4163
software-based system	3. 4164
software-related group	3. 4165
software release management planned information	3. 4166
S0I	3. 4486
S0IC	3. 4014
solution domain	3. 4168
S00	3. 4263
S0P	3. 4015, 3. 4241
SoPC	3. 4490
sort	3. 4169
sort generator	3. 4170
sorting	3. 4171
S0S	3. 4487
SoSE	3. 4488
source address	3. 4172
source code	3. 4173
source code generator	3. 4174
source code organization	3. 4175
source document	3. 4176
Source documents	3. 4177
source language	3. 4178
source lines of code	3. 4179
source node	3. 4180
source program	3. 4181
source statements	3. 4182
S0W	3. 4264
SP	3. 4095
space very small entity	3. 4183
space VSE	3. 4183
span	3. 4184

spawn action	3. 4185
special cause	3. 4186
special cause /assignable cause	3. 4187
special cause of a defect	3. 4188
special function register	3. 4189
specialization	3. 4190
specialize	3. 4191
specialty engineering	3. 4192
specific address	3. 4193
specific code	3. 4194
specific profile group	3. 4195
specific symbol	3. 4196
specific, measurable, achievable, relevant, and timely	3. 4197
specification	3. 4198
specification change notice	3. 3845, 3. 4199
specification language	3. 4200
specification tree	3. 4201
specification verification	3. 4202
specification-based testing	3. 4203
specification-basedtesting	3. 4204
SPI	3. 3833, 3. 3834, 3. 3926
spike testing	3. 4205
spiral model	3. 4206
SPLC	3. 4112, 3. 4113
SPLCM	3. 4114, 3. 4115
SPLCP	3. 4116, 3. 4117
split key	3. 4207
split-run testing	3. 1
SPMPI	3. 4118
sponsor	3. 4208
SPQM—RM	3. 4209
SPR	3. 3867
spreadsheet program	3. 4210
sprint	3. 4211
sprint backlog	3. 4212
SPS	3. 4213
SQA	3. 4120
SQC	3. 4122, 3. 4123
SQL	3. 4214
SQO	3. 3958
squiggle	3. 4215
SRAM	3. 4279

SRD	3. 4216
SRE	3. 4132
SRMPI	3. 4166
SRR	3. 4136, 3. 4141, 3. 4494, 3. 4499
SRS	3. 4142
SS	3. 3838, 3. 4182, 3. 4252
SSE	3. 4217
SSOP	3. 3979
SSPL	3. 4026
SSR	3. 4148
SSS	3. 4218
stability schema	3. 4219
stabilization phase	3. 4220
stable branch	3. 4221
stable process	3. 4222
stack pointer	3. 4223
staged representation	3. 4224
stages	3. 4225
stakeholder	3. 4226
stakeholder engagement assessment matrix	3. 4227
stakeholder engagement plan	3. 4228
stakeholder equity	3. 4229
stakeholder need	3. 4230
Stakeholder needs	3. 4231
Stakeholder needs and requirements definition record	3. 4232
Stakeholder needs and requirements definition strategy	3. 4233
stakeholder performance domain	3. 4234
stakeholder perspective	3. 4235
Stakeholder requirements	3. 4236
stakeholder requirements specification	3. 4237
Stakeholder requirements traceability	3. 4238
stakeholder satisfaction	3. 4239
staleness	3. 1509
stand-alone	3. 4240
standard operating procedure	3. 4241
standard process	3. 4242
standard technical English	3. 4243
standardized profile	3. 4244
standards enforcer	3. 4245
standby mode	3. 4246, 3. 4247
standby redundancy	3. 4248
stand-up meeting	3. 4249

starting address	3. 4250
start-to-finish	3. 4251
start-to-start	3. 4252
state	3. 4253
state class	3. 4254
state data	3. 4255
state diagram	3. 4256
state invariant condition	3. 4257
state machine	3. 4258
state name	3. 4259
state transition diagram	3. 4260
state transition testing	3. 4261
statement coverage	3. 4262
statement of objectives	3. 4263
statement of work	3. 4264
statement testing	3. 4265
StateTransition	3. 4266
static	3. 4267
static analysis	3. 4268
static analyzer	3. 4269
static attribute	3. 4270
static binding	3. 4271
static breakpoint	3. 4272
static classification	3. 4273
static dump	3. 4274
static error	3. 4275
static model	3. 4276
static product	3. 4277
static program analysis	3. 4278
static random access memory	3. 4279
static random access memory (SRAM)	3. 4280
static schema	3. 4281
static testing	3. 4282
statistical process control	3. 4283
statistical sampling	3. 4284
statistical test model	3. 4285
statistically managed process	3. 4286
status code	3. 4287
status meeting	3. 4288
status report	3. 4289
STE	3. 4153, 3. 4243
steering committee	3. 4290

step	3. 4291
step-by-step instructions	3. 4292
step-by-step operation	3. 4004, 3. 4293
stepwise refinement	3. 4294
stereotype	3. 4295
stimulus	3. 4296
STL	3. 3903, 3. 3904
stock keeping unit	3. 4297
stop	3. 4298
storage	3. 4299, 3. 4304
storage allocation	3. 4300
storage breakpoint	3. 1141, 3. 4301
storage capacity	3. 4302
storage device	3. 4304
storage efficiency	3. 4303
store	3. 4305
story map	3. 4306
story point	3. 4307
story-mapping	3. 4308
STP	3. 4309
straight-line code	3. 4310
straight-line coding	3. 4311
strategic asset management plan	3. 4312
strategic IT asset management plan	3. 4313
strategic IT asset management plan.	3. 4314
strategic plan	3. 4315
strategy artifacts	3. 4316
Strategy documents	3. 4317
stratified language	3. 4318
stream interface	3. 4319
stream interface signature	3. 4320
strengths, weaknesses, opportunities, and threats	3. 4321
strengths, weaknesses, opportunities, and threats analysis	3. 4322
stress testing	3. 4323
strong typing	3. 4324
structural feature	3. 4325
structural model aspect	3. 4326
structural testing	3. 4327, 3. 4328, 3. 4329
structure chart	3. 4330, 3. 4331
structure clash	3. 4332
structure model	3. 4333
structure-based testing	3. 4329, 3. 4334

structured authoring	3. 4335, 3. 4341
structured design	3. 4336
structured program	3. 4337
structured programming	3. 4338
structured programming language	3. 4339
structured walkthrough	3. 4340
structured writing	3. 4341
stub	3. 4342
style	3. 4343
subactivity	3. 4344
subactivity state	3. 4345
subclass	3. 4346
subclass cluster	3. 4347, 3. 4348
subclass responsibility	3. 4349
subdomain	3. 4350
subject	3. 4351
subject area	3. 4352
subject domain	3. 4353
subject domain responsibility	3. 4354
subject software	3. 4355
subject tool	3. 4356
subject unit	3. 4357
subject-matter expert	3. 4358
submachine state	3. 4359
subnetwork	3. 4360
subpackage	3. 4361
sub-path	3. 4362
subphase	3. 4363
subprogram	3. 4364
subproject	3. 4365
subroutine	3. 4366
subroutine trace	3. 4367, 3. 4368
subscription-based license	3. 4369, 3. 4370, 3. 4371
subsetting	3. 4372
substate	3. 4373
substitutability	3. 4374
sub-subcharacteristic	3. 4375
subsystem	3. 4376
subtype	3. 4377
success criteria	3. 4378
successor activity	3. 4379
suitability	3. 4380

summary activity	3. 4381
summative evaluation	3. 4382
sunk cost	3. 4383
superclass	3. 4384
supercomputer	3. 4385
supertype	3. 4386
supervised learning	3. 4387
supervisor	3. 1020, 3. 1742, 3. 4388, 3. 4391
supervisor state	3. 4389, 3. 4390
supervisory program	3. 1020, 3. 1742, 3. 4391
supervisory state	3. 1741
supplemental Ent	3. 4392, 3. 4393
supplemental entitlement schema	3. 4392, 3. 4393
supplemental information structure	3. 4394
supplementary run	3. 4395
Supplied system	3. 4396
supplier	3. 4397, 3. 4399
Supplier management procedure	3. 4398
Supply agreement	3. 4400
Supply payment	3. 4401
Supply record	3. 4402
Supply report	3. 4403
Supply response	3. 4404
Supply strategy	3. 4405
support activity group	3. 4406
support manager	3. 4407
support manual	3. 4408
support software	3. 4409
support staff-hour	3. 4410
support stage	3. 4411
support vector machine	3. 4412
supportability	3. 4413
supported product	3. 4414
surveillance	3. 4415
survivability	3. 4416
suspension criteria	3. 4417
sustainment	3. 4418
SUT	3. 4510
SV	3. 3835
SVM	3. 4412
SW	3. 4167
swap	3. 4419

SWCI	3. 853
SWE	3. 3971
SWID	3. 4079
swimlane	3. 4420
SWOT	3. 4321
SWOT analysis	3. 4322
symbol	3. 4421
symbol table	3. 4422
symbolic address	3. 4423
symbolic execution	3. 4424
symbolic language	3. 4425
symbolic trace	3. 4426
symmetric net	3. 4427
synch state	3. 4428
synchronize	3. 4429
synchronous	3. 4430
synchronous dynamic random access memory	3. 4431
synchronous graphics random access memory (SGRAM)	3. 4432
synchronous message communication	3. 4433, 3. 4438
synchronous message communication with reply	3. 4434, 3. 4435
synchronous message communication without reply	3. 4436, 3. 4437
synchronous request	3. 4439
syntactic agreement	3. 4440
syntactic error	3. 4441, 3. 4442
syntax	3. 4443
syntax error	3. 4442
syntax testing	3. 4444
synthesis	3. 4445
SyRS	3. 4500
system	3. 4446
System analysis record	3. 4447
System analysis report	3. 4448
System analysis strategy	3. 4449
system and software product quality	3. 4450
system architect	3. 4451
system architecture	3. 4452
System architecture description	3. 4453
system architecture rationale	3. 4454
system boundary	3. 4455
system breakdown structure	3. 4456
system characteristic	3. 4457
system description	3. 4458

system design	3. 4459
System design description	3. 4460
system design rationale	3. 4461
system design review	3. 3860
system development	3. 4462
system development cycle	3. 4463
system documentation	3. 4464
system effectiveness	3. 4465
system effectiveness analysis	3. 4466
system element	3. 4467
system element description	3. 4468
system element documentation	3. 4469
system engineering group	3. 4470
system entity	3. 4471
system follow-up	3. 4472, 3. 4473, 3. 4474
system function definition	3. 4475
system function identification	3. 4476
system functional interface identification	3. 4477
system hazard	3. 4478
system integration	3. 4479
system integration module	3. 4480
system interface task	3. 4481
system library	3. 4482
system life cycle	3. 4483
system maintenance	3. 4484
system model	3. 4485
system of interest	3. 4486
system of systems	3. 4487
system of systems engineering	3. 4488
system on a chip	3. 4489
system on a programmable chip	3. 4490
system profile	3. 4491
system reliability	3. 4492
system requirement	3. 4493
system requirement review	3. 4494
system requirements	3. 4495
system requirements allocated to software	3. 4496
system requirements definition record	3. 4497
system requirements definition strategy	3. 4498
system requirements review	3. 4136, 3. 4494, 3. 4499
system requirements specification	3. 4500
System requirements traceability	3. 4501

system safety	3. 4502
system software	3. 4503
system specification	3. 4504
system structure	3. 4505
system support	3. 4506
system table	3. 4507
system test of user documentation	3. 4508
system testing	3. 4509
system under test	3. 4510
systematic failure	3. 4511
systematic reuse	3. 4512
system-of-interest	3. 4513
systems analysis	3. 4514
systems development life cycle	3. 3859, 3. 4515
systems engineering	3. 4516
systems engineering effort	3. 4517
Systems Engineering Management Plan	3. 3910, 3. 4518
Systems Engineering Plan	3. 3914, 3. 4518
systems integration testing	3. 4519

T

T&E	3. 4520
T&M contract	3. 4725
table	3. 4521
table heading	3. 4522
table-driven method	3. 4523
tag	3. 4524
tag slide	3. 4525
tagid	3. 4526
tailored process	3. 4527
tailoring	3. 4528
tailoring guideline	3. 4529
tangibility	3. 4530
target audience	3. 4531
target capability	3. 4532
target computer	3. 4533
target entity	3. 4534
target language	3. 4535
target language object language	3. 4536
target machine	3. 4537
target node	3. 4538
target of process	3. 4539
target of verification	3. 4540

target process profile	3. 4541
target risk	3. 4542
task	3. 4543
task behaviour specification	3. 4544
task board	3. 4545
task completion	3. 4546
task interface	3. 4547
task inversion	3. 4548
task kick-off meeting	3. 4549
task mode	3. 4550
task priority criteria	3. 4551
task structuring	3. 4552
task submission	3. 4553
task type	3. 4554
taskboard	3. 4555
task-clustering criteria	3. 4556
task-structuring criteria	3. 4557
taxonomy	3. 4558
TBD	3. 4750
TCPI	3. 4559, 3. 4751
TDD	3. 4560, 3. 4690
team	3. 4561
team performance domain	3. 4562
team selection plan	3. 4563
technical complexity adjustment	3. 4564
technical complexity adjustment factors	3. 4565
technical debt	3. 4566
technical independence	3. 4567
technical management	3. 4568
technical performance measure	3. 4569
technical product quality requirement	3. 4570
technical requirement	3. 4571
technical review	3. 4572
technical standard	3. 4573
technology roadmap	3. 4574
technology used for evaluation	3. 1710
technology viewpoint	3. 4575
TEMP	3. 4576
template	3. 4577
template <X> template	3. 4578
template class	3. 4579
template rating module	3. 4580

template type	3. 4581
temporal clustering	3. 4582
temporal cohesion	3. 4583
term	3. 4584
term evaluation	3. 4585
term-based license	3. 4370, 3. 4371
terminal	3. 4586
terminal symbol	3. 4587
terminating behaviour	3. 4588
termination construct	3. 4589
termination deliver	3. 4590
termination proof	3. 4591
termination submit	3. 4592
terminology management system	3. 4593
test approach	3. 4594
test basis	3. 4595
test bed	3. 4596
test case	3. 4597
test case generator	3. 4598, 3. 4599
test case specification	3. 4600
test class	3. 4601
test completion process	3. 4602
test completion report	3. 4603, 3. 4604
test condition	3. 4605
test coverage	3. 4606
test coverage item	3. 4607, 3. 4608
test criteria	3. 4609
test data	3. 4610
test data generator	3. 4599, 3. 4611
test data readiness report	3. 4612
test description	3. 4613
test design	3. 4614, 3. 4618
test design and implementation process	3. 4615
test design specification	3. 4616
test design technique	3. 4617
test documentation	3. 4619
test driver	3. 4620
test effort	3. 4621
test environment	3. 4622
test environment and data management process	3. 4623
test environment item	3. 4624
test environment readiness report	3. 4625

test environment requirements	3. 4626
test environment set-up process	3. 4627
test execution	3. 4628
test execution engine	3. 4629
test execution log	3. 4630
test execution process	3. 4631
test framework	3. 4632
test generator	3. 4599, 3. 4633
test incident	3. 4634
test incident report	3. 4635
test incident reporting process	3. 4636
test independence	3. 4637
test interface	3. 4638
test interface layer	3. 4639
test item	3. 4640, 3. 4642
test item transmittal report	3. 4641
test lead	3. 4643
test level	3. 4644, 3. 4645
test log	3. 4646
test management	3. 4647
test management process	3. 4648
test model	3. 4649
test model specification	3. 4650
test monitoring and control process	3. 4651
test object	3. 4642
test objective	3. 4652
test oracle	3. 4653
test oracle problem	3. 4654
test organization	3. 4655
test participant	3. 4656
test phase	3. 4645, 3. 4657
test plan	3. 4658
test planning process	3. 4659
test policy	3. 4661
test policy, organizational test policy	3. 4660
test practice	3. 4662
test procedure	3. 4663
test process	3. 4664
test protocol	3. 4665
test readiness review	3. 4666
test repeatability	3. 4667
test report	3. 4668

test requirement	3. 4669
test result	3. 4670
test scenario	3. 4671
test script, test procedure specification	3. 4672
test set	3. 4673
test set architecture	3. 4674
test specification	3. 4675
test specification technique	3. 4676
test status report	3. 4677
test strategy	3. 4678
test strategy and planning process	3. 4679
test sub-process	3. 4680
test suite	3. 4681
test summary report	3. 4604, 3. 4682
test target version	3. 4683
test technique	3. 4684
test traceability matrix	3. 4685, 3. 4686
test type	3. 4687
test unit	3. 4688
testability	3. 4689
test-driven development	3. 4690
testing	3. 4691
testing description	3. 4692
testing pyramid	3. 4693
testing task iteration	3. 4694
testing tool	3. 4695
test-validity	3. 4696
testware	3. 4697
text	3. 4698
text editor editor	3. 4699
text page	3. 4700
text processing	3. 4701, 3. 4702
text processing word processing	3. 4703
texture	3. 4705
texture, architectural texture	3. 4704
theme	3. 4706
think time	3. 4707
third normal form	3. 4708
thrashing	3. 4709
thread	3. 4710
thread [of control]	3. 4711
threat modeling	3. 4712

three-address instruction	3. 4713
three-plus-one address instruction	3. 4714
threshold	3. 4715
threshold coverage	3. 4716
throughput	3. 4717
throughput chart	3. 4718
throughput rate	3. 4719
throughput rating value	3. 4720
throughput reference value	3. 4721
thumbnail	3. 4722
tick	3. 4723
tier chart	3. 4724
tightly coupled message communication	3. 4438
tightly coupled message communication with reply	3. 4435
tightly coupled message communication without reply	3. 4437
time and materials contract	3. 4725
time behaviour	3. 4726
time class	3. 4727
time event	3. 4728
time expression	3. 4729
time out	3. 4730
time Petri net	3. 4731
time sharing	3. 4732
time slicing	3. 4733
timebox	3. 4734
time-critical task	3. 4735
timeliness	3. 4736
timeliness function	3. 4737
timeliness rating value	3. 4738
timely throughput	3. 4739
timer event	3. 4740
timer pulse unit	3. 4741
time-scaled schedule network diagram	3. 4742
timing	3. 4743
timing analyser	3. 4744
timing analysis	3. 4745
timing diagram	3. 4746
TINA	3. 4747
tinderbox	3. 4748
TMRR	3. 4749
to be determined	3. 4750
to-complete performance index	3. 4751

TOE	3. 4752
token	3. 4753
tolerable risk	3. 4754
tolerance	3. 4755
tool	3. 4756
tool bridge	3. 4757
toolbox	3. 4758
tool-specific information	3. 4759
top level	3. 4760
top-down	3. 4761
top-down decomposition	3. 4763
top-down design	3. 4763
top-down design, top-down decomposition	3. 4762
top-down testing	3. 4764
topic	3. 4765
tornado diagram	3. 4766
total	3. 4767
total cluster	3. 4770
total cluster complete cluster	3. 4768
total cluster, complete cluster	3. 4769
total correctness	3. 4771
total quality management	3. 4772
TOV	3. 4540
TPM	3. 4569
T-profile	3. 4773
TPU	3. 4741
TQM	3. 4772, 3. 4774
TR	3. 4571
TRA	3. 4775
trace	3. 4776
trace link	3. 4777
trace link semantics	3. 4778
traceability	3. 4779
traceability matrix	3. 4780
traceability with different abstraction level	3. 4781
traceable	3. 4782
tracer	3. 4783
trade study	3. 4784
trade-off	3. 4785
trade-off analysis	3. 4786
trailer	3. 4787
trailing decision	3. 4788

training data	3. 4789
training group	3. 4790
training program	3. 4791
transaction	3. 4792
transaction analysis	3. 4794
transaction analysis, transaction-centred design	3. 4793
transaction file	3. 4795
transaction matrix	3. 4796
transaction schema	3. 4797
transaction transparency	3. 4798
transactional function	3. 4799
transactional function type	3. 4800
transaction-centered design	3. 4794, 3. 4801
transfer	3. 4802
transfer file	3. 4803
transfer of an entitlement	3. 4804
transform analysis	3. 4805, 3. 4806
transformation analysis	3. 4805, 3. 4806
transform-centered design	3. 4806, 3. 4807
transform-centred design	3. 4805
transient error	3. 4808
transient object	3. 4809
transition	3. 4810
transition occurrence transition rule	3. 4811
transition scenario	3. 4812
transition variables	3. 4813
translator	3. 4814
transparency	3. 4815
transparent box	3. 5110
transportability	3. 3048
trap	3. 4816
tree diagram	3. 4817
tree-structured chart	3. 4818
trend analysis	3. 4819
trigger condition	3. 4820
triggering event	3. 4821, 3. 4822
triggering event type	3. 4822
triggers	3. 4823
triple bottom line	3. 4824
triple constraint	3. 4825
TRR	3. 4666
true negative	3. 4826

true positive	3. 4827
trunk	3. 4828
trustworthiness	3. 4829
trustworthiness principle	3. 4830
trustworthy data	3. 4831
trustworthy process area	3. 4832
T-shirt sizing	3. 4833
tunnel notation	3. 4834
tunneled arrow	3. 4835
tunneling	3. 4836
tuple	3. 4837
turnaround time	3. 4838
turnkey	3. 4839
turnkey system	3. 4840
tutorial	3. 4841
two-address instruction	3. 1504, 3. 4842, 3. 4843
two-level address	3. 4844
two-level encoding	3. 4845
two-phase acquisition	3. 4846
two-plus-one address instruction	3. 4847
type	3. 4848
type expression	3. 4849

U

UART	3. 4881
ULA	3. 4850
UML	3. 4865, 3. 4866
unbinding behavior	3. 4851
unbinding behaviour	3. 4852
unbundle	3. 4853
uncertainty	3. 4854
uncertainty performance domain	3. 4855
unconditional branch	3. 4856
unconditional jump	3. 4857
underfitting	3. 4858
underflow exception	3. 4859
underlying license	3. 4860
understandability	3. 4861
undeveloped argument	3. 4862
undirected graph	3. 4863
unidimensionality	3. 4864
Unified Modeling Language	3. 4865
unified modeling language™	3. 4866

uniform resource identifier	3. 4867
Uniform Resource Locator	3. 4868
unintended user	3. 4869
uninterpreted	3. 4870
unique function	3. 4871
uniqueness constraint	3. 4872
unit	3. 4873
unit (of measure)	3. 4874
unit development folder	3. 4061
unit of measurement	3. 4875
unit price of size	3. 4876
unit requirements documentation	3. 4877
unit test	3. 4878
unit test framework	3. 4879
unit testing	3. 4880
universal asynchronous receiver/transmitter	3. 4881
universal serial bus	3. 4882
unpack	3. 4883
unscripted testing	3. 4884
unspecialize	3. 4885
unstratified language	3. 4886
unsupervised learning	3. 4887
UNTIL	3. 4888
unwind	3. 4889
up	3. 4890
up time	3. 4891
updatable argument	3. 4892
upload	3. 4893
upward compatible	3. 4894
upward compression	3. 4895
UR	3. 4936
URa	3. 4953
URI	3. 4867
URL	3. 4868
usability	3. 4896
usability defect	3. 4897
usability finding	3. 4898
usability inspection	3. 4899
usability laboratory	3. 4900
usability objective	3. 4901
usability problem	3. 4902
usability test	3. 4903, 3. 4904

usability testing	3. 4905
usability walkthrough	3. 4906
usage mode	3. 4907
USB	3. 4882
use case	3. 4908
use case diagram	3. 4909
use case instance	3. 4910
use case model	3. 4911
use case specification	3. 4912
use error	3. 4913
use of IT	3. 4914
usefulness	3. 4915
user	3. 4916
user assistance	3. 4917
user controllability	3. 4918
user documentation	3. 4919
user engagement	3. 4920
user error protection	3. 4921
user experience	3. 4922
user group	3. 4923
user guide	3. 4924
user interaction	3. 4925
user interface	3. 4926
user interface aesthetics	3. 4927
user interface element	3. 4928, 3. 4930
user interface element user interface object	3. 4929
user interface object	3. 4930
user interface task	3. 4931
user manual	3. 4932
user need	3. 4933
User notification	3. 4934
user profile	3. 4935
user requirements	3. 4936
user state	3. 3141
user story point	3. 4937
user system interaction requirements	3. 4938
user terminal	3. 4939
user type	3. 4940
user view	3. 4941
user-based evaluation	3. 4942
user-centered design	3. 2053
user-defined reliability class	3. 4943

use-related quality requirements	3. 4944
user-friendly	3. 4945
user-recognizable	3. 4946
user-system interaction	3. 4947
utilitarianism	3. 4948
utility	3. 4949
utility software	3. 4950
utilization	3. 4951
utilization bound theorem	3. 4952
utilize benchmarking results activity	3. 4953

V

V&V	3. 5034
V&V effort	3. 5035, 3. 5043
VAC	3. 5015
valid	3. 4954
validate scope	3. 4955
validated deliverables	3. 4956
validated metric	3. 4957
validation	3. 4958
validation action	3. 4959
validation data	3. 4960
validation effort	3. 5043
Validation procedure	3. 4961
Validation report	3. 4962
validation test	3. 4963
value at risk	3. 4964
value baseline	3. 4965
value bearer	3. 4966
value benefit	3. 4967
value chain analysis	3. 4968
value change coverage	3. 4969
value class	3. 4970
value cluster	3. 4971
value delivery office	3. 4972
value delivery system	3. 4973
value demonstrator	3. 4974
value disposition	3. 4975
value engineering	3. 4976
value harm	3. 4977
value lead	3. 4978
value list constraint	3. 4979
value proposition	3. 4980

value range constraint	3. 4981
value register	3. 4982
value stream map	3. 4983
value stream mapping	3. 4984
value trace	3. 1206, 3. 5013
value-added reseller	3. 4985
value-based system requirement	3. 4986
vanity metric	3. 4987
VAR	3. 4985
variability	3. 4988
variability binding	3. 4989
variability constraint	3. 4990
variability dependency	3. 4991
variability documentation	3. 4992
variability implementation	3. 4993
variability in requirements	3. 4994
variability in space	3. 4995
variability in test cases	3. 4996
variability in time	3. 4997
variability index	3. 4998
variability management	3. 4999
variability mechanism	3. 5000
variability mechanism operationalization	3. 5001
variability mechanism operationalization (VMO)	3. 5002
variability model	3. 5003
variability modeling plan	3. 5004
variability modeling strategy	3. 5005
variability modelling plan	3. 5006
variability modelling strategy	3. 5007
variability traceability	3. 5008
variable	3. 5009
variable configuration item	3. 5010
variable cost	3. 5011
variable definition	3. 1152
variable strength	3. 5012
variable trace	3. 1206, 3. 5013
variance analysis	3. 5014
variance at completion	3. 5015
variance at completion (VAC)	3. 5016
variant	3. 5017
variant selection	3. 5018
variation	3. 5019

variation in space	3. 5020
variation in time	3. 5021
variation point	3. 5022
variability modeling plan	3. 5023
VCRM	3. 5024
VDD	3. 5025, 3. 5049
VDO	3. 4972
VDT	3. 5056
VDU	3. 5083
VE	3. 4976
Vee model (建议删除)	3. 5085
velocity	3. 5026
velocity chart	3. 5027
vendor	3. 4399
vendor branch	3. 5028
vendor service context codeset ID	3. 5029
verb phrase	3. 5030
verifiable	3. 5031
verifiable requirements language	3. 1904, 3. 1905
verification	3. 5032, 3. 5043
verification action	3. 5033
verification and validation	3. 5034
verification and validation effort	3. 5035
verification and validation plan	3. 5036
verification cross reference matrix	3. 4686
verification method	3. 5037
Verification report	3. 5038
verification sponsor	3. 5039
verification system	3. 5040
verification test	3. 5041
verification tool	3. 5042
verified deliverable	3. 5044
verify scope	3. 5045
Verilog hardware description language	3. 5046
version	3. 5047
version control	3. 5048
version description document	3. 5049
version identifier	3. 5050
versioning	3. 5051
vertex	3. 5052
vertical microinstruction	3. 5053
very small and micro enterprise	3. 5054

very small entity	3. 5055
VHDL	3. 5046
video display terminal	3. 5056
video random access memory	3. 5057
view	3. 5058
view component	3. 5059
view diagram	3. 5060
view element	3. 5061
view projection	3. 5062
viewpoint	3. 5063
viewpoint (on a system)	3. 5064
viewpoint correspondence	3. 5065
viewpoint statement	3. 5066
violation	3. 5067
virtual	3. 5068
virtual address	3. 5069
virtual machine	3. 5070
virtual memory	3. 5075
virtual prototype	3. 5071
virtual prototype system	3. 5072
virtual reference	3. 5073
virtual SoS	3. 5076
virtual space	3. 5074
virtual storage	3. 5075
virtual system of systems	3. 5076
virtual team	3. 5077
virtual test environment	3. 5078
virtualization	3. 5079
visibility	3. 5080
vision statement	3. 5081
visual data and information	3. 5082
visual display unit	3. 5083
vital business service	3. 5084
VM	3. 5070
VMO	3. 5001
V-model	3. 5085
vocabulary	3. 5086
volatile memory	3. 5087
volatility	3. 5088
volume testing	3. 5089
VRAM	3. 5057
VSCID	3. 5029

VSE	3. 5055
VSME	3. 5054
vulnerability	3. 5090
VVP	3. 5036

W

waiver	3. 5091
walk-through	3. 5092
WAP	3. 5116
war room	3. 5093
warning	3. 5094
warning message	3. 5095
waste	3. 5096
watchdog timer	3. 5097
waterfall model	3. 5098
WAV	3. 5099
Waveform audio file format	3. 5099
WBS	3. 5122
WCAG	3. 5101
WDT	3. 5097
wearout-failure period	3. 5100
Web Content Accessibility Guidelines	3. 5101
web lead	3. 5102
weighted milestone method	3. 5103
well-being	3. 5104
well-being impact assessment	3. 5105
well-being metrics	3. 5106
well-defined process	3. 5107
what-if scenario analysis	3. 5108
WHILE	3. 3096, 3. 5109
white box	3. 5110
white-box testing	3. 4328, 3. 4329
whitespace	3. 5111
whole process of software development	3. 5112
WIA	3. 5105
wideband Delphi	3. 5113
wiki	3. 5114
window	3. 5115
Wireless Application Protocol	3. 5116
without prejudice	3. 5117
wizard	3. 5118
word	3. 5119
word processing	3. 4702

work authorization	3. 5120
work authorization system	3. 5121
work breakdown structure	3. 5122
work breakdown structure component	3. 5123
work effort	3. 5124
work package	3. 5125
work performance data	3. 5126
work performance information	3. 5127
work product	3. 5128
work unit	3. 5129
workaround	3. 5130
workflow board	3. 5131, 3. 5132
working agreement	3. 5133
working area	3. 5134, 3. 5137
working metamodel	3. 5135
working set	3. 5136
working space	3. 5137
working storage	3. 5137, 3. 5138
workload	3. 5139
workload model	3. 5140
write	3. 5141
write type	3. 5142
write-type	3. 5142
writing reference	3. 5143
X	
XML	3. 1777
XML schema definition	3. 5144
XP	3. 1804
XSD	3. 5144, 3. 5145
XSLT	3. 1778, 3. 1779
Y	
yesterday's weather	3. 5146
Z	
zero-address instruction	3. 5147
#	
24765 中是 “ alternatives analysis”	3. 164
50-50 estimate	3. 1756
5GL	3. 1858, 3. 1859