

中华人民共和国国家标准

GB/T 25000.10—XXXX
代替 GB/T 16260.1-2006

系统与软件工程 系统与软件质量要求和 评价 (SQuaRE) 系统与软件质量模型

Systems and software engineering — Systems and software Quality Requirements and
Evaluation (SQuaRE) — System and software quality models

(ISO/IEC 25010:2011, MOD)

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

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 质量模型框架	6
5 使用质量模型及特性说明	6
5.1 使用质量模型	7
5.2 使用质量特性说明	7
6 产品质量模型及特性说明	8
6.1 产品质量模型	8
6.2 产品质量特性说明	9
7 质量模型的目标	13
8 质量模型的使用	14
9 来自不同利益相关方视角的质量	15
10 各质量模型间的关系	16
附录 A（资料性附录） 与 GB/T 16260.1-2006 中的质量模型的对比	17
附录 B（资料性附录） 与 ISO/IEC 25010:2011 的差异	20
附录 C（资料性附录） 使用质量模型测量	22
参考文献	27

前 言

GB/T25000在《系统与软件工程 系统与软件产品质量要求和评价(SQuaRE)》总标题下拟分为如下几部分：

- 第1部分：SQuaRE指南
- 第2部分：计划与管理
- 第10部分：系统与软件质量模型
- 第12部分：数据质量模型
- 第20部分：测量参考模型和指南
- 第21部分：质量测量元素
- 第22部分：使用质量测量
- 第23部分：系统和软件产品质量测量
- 第24部分：数据质量测量
- 第30部分：质量需求
- 第40部分：评价过程
- 第41部分：开发方、需方和独立评价方的评价指南
- 第42部分：评价模块
- 第45部分：可恢复性的评价模块
- 第51部分：就绪即用软件产品（RUSP）的质量要求和测试细则
- 第60部分：易用性测试报告行业通用格式（CIF）：易用性相关信息的通用框架
- 第62部分：易用性测试报告行业通用格式（CIF）
- 第63部分：易用性的行业通用格式（CIF）：使用周境描述
- 第64部分：易用性的行业通用格式（CIF）：用户要求报告
- 第65部分：易用性的行业通用格式（CIF）：用户需求规格说明
- 第66部分：易用性的行业通用格式（CIF）：评价报告

本部分是GB/T25000的第10部分。

本部分按照GB/T1.1-2009给出的规则起草。

本部分修改采用国际标准ISO/IEC 25010:2011《系统与软件工程 系统与软件质量要求和评价(SQuaRE) 系统与软件质量模型》（英文版）。本部分与ISO/IEC 25010:2011的差异参见附录A。

本部分代替GB/T 16260.1-2006《软件工程 产品质量 第1部分：质量模型》。新版标准中有关产品质量特性的表述由6大特性修改调整为8大特性，使用质量特性由4大特性修改调整为5大特性。相关的子特性也做了修改、调整和补充。具体的修订细节参见附录B。

本部分由全国信息技术标准化技术委员会（SAC/TC28）提出并归口。

本部分起草单位：上海计算机软件技术开发中心、中国电子技术标准化研究院。

本部分主要起草人：。

本部分于1996年首次发布。2006年第一次修订。

引 言

软件产品和软件密集型的计算机系统正越来越多地用于实现各种各样的企业和个人的功能。个人满足感、业务成功和/或人类安全的目标和目的的实现依赖于高质量的系统与软件。高质量的软件产品和软件密集型计算机系统对于利益相关方是必不可少的，它可以提供价值，并避免潜在的负面影响。

软件产品和软件密集型的计算机系统的利益相关方，包括开发方、需方、使用方或者是使用软件密集型计算机系统的企业客户。对系统与软件质量全面地规范和评估是保证利益相关方利益的关键因素，可以通过定义在系统中与利益相关方的目的和目标相关的必要和期望的质量特性来实现。这包括相关的软件系统和数据，以及该系统对其利益相关方的影响的质量特性。同时，尽可能使用经确认的或被广泛认可的措施和测量方法，对于规定、测量和评价质量特性至关重要。本部分中的质量模型可以用来鉴定相关的质量特性，并进一步用来建立令人满意的需求、准则和相关措施。

本部分的质量模型可以与ISO/IEC 12207和ISO/IEC 15288结合使用，特别是与需求定义、验证和确认相关的过程，该过程重点关注质量要求的规范和评价。ISO/IEC 25030描述了质量模型如何在软件质量需求的过程中发挥作用，而ISO/IEC 25040则描述了质量模型如何在软件质量评价的过程中发挥作用。

本部分可以和ISO/IEC 15504（与软件过程评估有关）一起使用，以提供：

- 客户—供方过程中的软件产品质量定义框架；
- 在支持过程中对于评审、验证和确认的支持以及一个定量的质量评价框架；
- 在管理过程中对于设置组织质量目标的支持。

本部分也可以和GB/T 19001（与质量保证过程有关）一起使用，以提供：

- 对于设立质量目标的支持；
- 对于设计评审、验证和确认的支持。

系统与软件工程 系统与软件质量要求和评价 (SQuaRE) 系统与软件质量模型

1 范围

GB/T25000的本部分定义了：

- a) 使用质量模型，该模型由 5 个质量特性组成，每个质量特性可进一步细分为子特性。这些特性关系到软件产品在特定周境下使用时和用户交互时的结果。该模型可应用于完整的人机系统，包括所涉及的计算机系统和软件产品。
- b) 产品质量模型，该模型由 8 个质量特性组成，每个质量特性可进一步细分为子特性。这些特性与软件的静态属性以及计算机系统的动态属性相关。该模型可以应用于计算机系统和软件产品。

上述两个模型中规定的各种质量特性与所有的软件产品及计算机系统都有关系。这些质量特性和子特性为系统和软件产品质量的细化、测量和评价提供了一致性的术语。同时，本部分提供了一套质量特性集合，通过与这些特性的对比，可以判断系统与软件产品质量需求的完备程度。

虽然产品质量模型的应用范围定义为软件和计算机系统，但其中的很多特性可以应用于更宽泛的系统和 Service 中。

ISO/IEC 25012 中包括的数据质量模型可以作为本部分模型的补充。

质量模型可以用来从获取、需求、开发、使用、评价、支持、维护、质量保证和审核相关的不同视角，对软件和软件密集型计算机系统确定和评价。例如它可以被开发者、需方、质量保证和控制人员和独立评价者，特别是那些对规定和评价软件产品质量负责的人员所使用。在产品开发生活中使用质量模型，可以获得的好处包括：

- 确定系统与软件需求；
- 确认需求定义的完善程度；
- 确定系统与软件的设计目标；
- 确定系统与软件的测试目标；
- 确定质量保证中的质量控制准则；
- 确定软件产品和/或软件密集型计算机系统的验收准则；
- 为支持上述活动而建立质量特性的度量。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25000.1 软件工程 软件产品质量要求和评价(SQuaRE) SQuaRE指南 (GB/T 25000.1-2006, IDT, ISO/IEC 25000:2010)

3 术语和定义

GB/T 25000.1中界定的以及下列术语和定义适用于本文件。

3.1

资产 asset

对个人或组织具有价值的任何东西。

注1：改写 ISO/IEC 13335-1:2004，定义 2.2。

注2：在本标准中，资产是指工作产品，如需求文档、源代码模块、测量定义等。

3.2

属性 attribute

实体的固有性质或特性，可由人工或自动化手段进行定量或定性的辨别。

注1：基于 GB/T 20917-2007。

注2：GB/T 19000 区分两类属性：在事物中固有地存在的一种持久的特性；给产品、过程或系统指派的特性（如产品的价格、产品的拥有者）。指派的特性并不是产品、过程或系统固有的质量特性。

[GB/T 25000.1-2010，定义4.3]

3.3

基准 benchmark

可以对照其进行测量或评估结果的标准。

[ISO/IEC/IEEE 24765:2010，定义3.256]

3.4

组件 component

在一个特定的分析层次上考虑的系统中带有分立结构的实体。诸如一个组合或软件模块。

[GB/T 18492-2001，定义3.1]

3.5

使用周境 context of use

用户、任务、设备（硬件、软件和原材料）以及使用某产品的物理和社会环境。

[GB/T 18978.11-2004，定义3.5]

3.6

直接用户 direct user

与产品交互的个体。

注1：包括主要用户和次要用户。

注2：与 GB/T 18978.11-2004 中“用户”的定义相同。

3.7

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共场合下的临时用户。

[GB/T 25000.1-2010，定义4.14]

3.8

软件质量外部测量 external measure of software quality

用于测量系统及其包含的软件在特定条件下使用时，软件产品使得系统的行为满足明确的和隐含的要求的程度。

注1：在测试和运行期间，通过运行软件产品来验证和/或确认行为的属性。

示例：在测试期间发现的失效数是一种软件质量外部测量，它与计算机系统中产生的故障数有关。这两种测量不一定相同，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可以引起明显不同的失效。

注2：基于 GB/T 25000.1-2010 中“软件外部质量”的定义。

3.9

隐含的要求 implied needs

可能未明确阐述却是实际需要的要求。

注：当软件产品用于特定场合时一些隐含的要求就成为显然的。

示例：隐含的要求包括：尽管没明确阐述但通过其他明确的要求而隐含的需要，以及由于其被认为是显然的或显而易见而未明确阐述的需要。

[GB/T 25000.1-2010，定义4.23]

3.10

间接用户 indirect user

接收系统的输出，但是不与系统进行交互的个人。

3.11

软件质量内部测量 internal measure of software quality

用于测量软件产品在特定条件下使用时，软件产品的一组静态属性满足明确的和隐含的要求的程度。

注1：静态属性包括那些与软件的架构、结构和它的组件有关的属性。

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。

示例：复杂度测量和在走查中发现的故障的数量、严重程度、失效频率等都是作用于软件产品自身的软件质量内部测量。

注3：基于 GB/T 25000.1-2010 中“软件内部质量”的定义。

3.12

测度（名） measure (noun)

作为测量结果被赋予值的变量。

注1：该术语“测度”总体上指的是基本测度、导出测度和指标。

注2：改写 GB/T 25000.1-2010，定义 4.23。

3.13

测量（动） measure (verb)

执行一次测量活动。

[GB/T 25000.1-2010，定义4.33]

3.14

测量 measurement

一组操作，其目的是确定某个测度的值。

[GB/T 20917-2007，基于1993年的计量学中基本术语和通用术语的国际词汇中的定义]

注：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）。

3.15

使用质量 quality in use

在指定的使用周境中，产品或系统能够被特定用户使用的程度，同时满足该特定用户在有效性、效率、抗风险度和满意度特性方面的要求。

3.16

质量测量 quality measure

是一种量度，该量度由两个或以上的质量测量元素通过测量函数计算得来。

[ISO/IEC TR 25021]

3.17

质量测量元素 quality measure element

是一种量度，该量度定义为一种属性以及定量测量该属性的方法，包括数学函数转换公式。

[ISO/IEC TR 25021]

3.18

质量模型 quality model

所定义的一组特性及特性间的关系，以为规定质量需求和评价质量而提供一个框架。

[GB/T 25000.1-2010，定义4.44]

3.19

质量属性 quality property

可以测量的质量要素。

3.20

风险 risk

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。

[GB/T 18492-2001，定义3.12]

3.21

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。

[GB/T 8566-2007]

注1：产品包括中间产品和意图用于开发方和维护方使用的产品。

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。

[GB/T 25000.1-2010，定义4.49]

3.22

软件质量 software quality

在规定条件下使用时，软件产品满足明确的和隐含的要求的能力。

注：该定义不同于GB/T 19000—2008中的质量定义，主要是本标准的定义是讲满足明确和隐含的要求，而GB/T 19000质量定义是讲满足要求。

[GB/T 25000.1-2010，定义4.51]

3.23

软件质量特性 software quality characteristic

软件质量的软件质量属性的种类。

注：软件质量特性可细化为多级子特性并最后到软件质量属性。

[GB/T 25000.1-2010，定义4.52]

3.24

软件质量需求 software quality requirement

对软件中现有的质量属性的要求。

3.25

利益相关方 stakeholder

对某个系统或它拥有的特性有权利、共享、要求或兴趣，以满足其需要和期望的个体或组织。

3.26

系统 system

为达到一个或多个明确目的而组织起来的、相互作用的元素的组合体。

注1：一个系统可被认为是一个产品或它提供的服务。

注2：实际上，对系统含义的解释通常通过使用一个联合名词来阐明，如飞行器系统。或者，单词“系统”可简单地由上下文相关的同义词来替代，如飞行器，虽然这可能使系统的观点不太明显。

[GB/T 22032-2008，定义4.17]

3.27

用户 user

在系统使用过程中，与系统进行交互或从系统中获益的个人或组织。

注：主要用户和次要用户与系统进行交互，并且主要用户和间接用户可以从系统中获益（见第9章）。

3.28

确认 validation

通过检查和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1：“确认过的”一词用来表示相应的状况。

注2：在设计和开发中，确认涉及到检查某个产品以确定是否符合用户需要的过程。

注3：确认通常是对最终产品在规定的使用条件下进行的。在早期阶段，也可能需要进行确认。

注4：如果有不同的预期用途，可以进行多重确认。

[GB/T 25000.1-2010，定义4.62]

3.29

验证 verification

通过检查和提供客观的证据来证实规定需求已经得到满足。

注1：“验证过的”用来表示相应的状况。

注2：在设计和开发中，验证是指对某项规定活动的结果进行检查的过程，以确定该活动对规定需求的符合情况。

[GB/T 25000.1-2010，定义4.64]

4 质量模型框架

系统的质量是系统满足各利益相关方明确的或隐含的要求并因此提供相关价值的的能力。这些明确的和隐含的要求在SQaRE系列标准中以质量模型的形式进行了阐述，这些质量模型将产品质量划分成不同类型的质量特性，有一些进一步划分为子特性或子子特性。这种按层的分解形式可以很容易地将产品质量进行分解细化。但是本标准并未穷举质量特性的所有子特性，而是选择了典型的关注点作为子特性的代表。

与系统质量相关的并可以测量的属性称为系统的质量属性，质量属性都有相关的质量测量。在对质量特性和子特性进行测量时，除非这些特性和子特性是可以直接测量的，否则就需要定义足够的属性来覆盖这些特性和子特性，然后对这些属性逐个测量，并将测量结果进行计算，才能够得到针对这种质量特性和子特性的质量测量结果（参见附录B）。图2给出了质量特性、子特性和质量属性之间的结构关系。

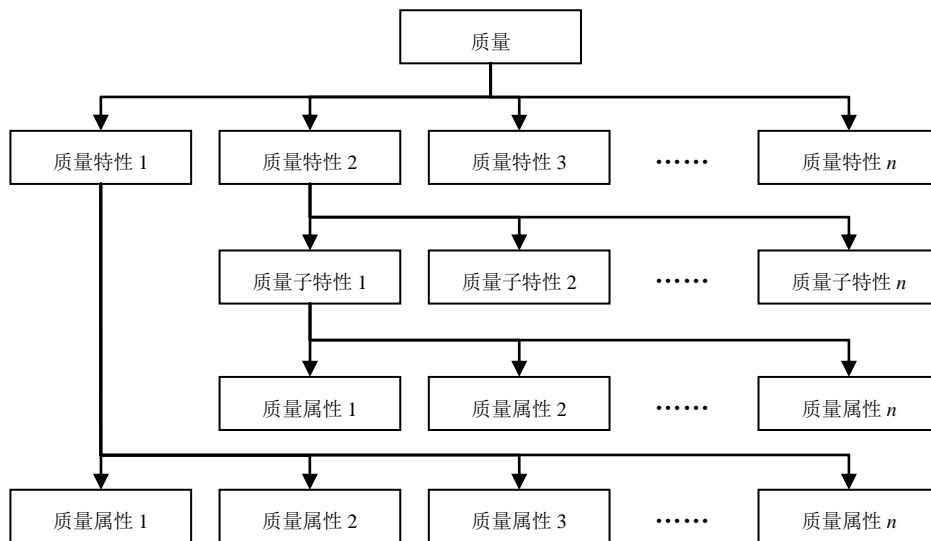


图1 质量模型结构图

SQaRE系列标准提供了三个质量模型，本部分描述了使用质量模型和产品质量模型，ISO/IEC 25012描述了数据质量模型。这三个模型共同组成了质量模型框架，尽可能考虑到质量的所有特性。这三个模型所提供的质量特性可以满足广大的利益相关方对质量的要求，例如软件开发方、系统集成方、需方、所有者、维护者、合同方、质量保证和控制人员、用户等等。

并非这些模型中的所有质量特性都和各个利益相关方都有关系，但是，在确定这些质量特性前，每一种类别的利益相关方都应参与质量特性的相关性的评审和考虑。例如，确立产品和系统的性能需求或评价准则。

5 使用质量模型及特性说明

5.1 使用质量模型

本章定义了与系统交互结果有关的5个特性：有效性、效率、满意度、抗风险性和周境覆盖度，其中满意度、抗风险性和周境覆盖度进一步细分为若干子特性（见图3）。每个特性可以被分配到不同的利益相关方的活动中去，例如操作人员的交互或开发人员的维护。

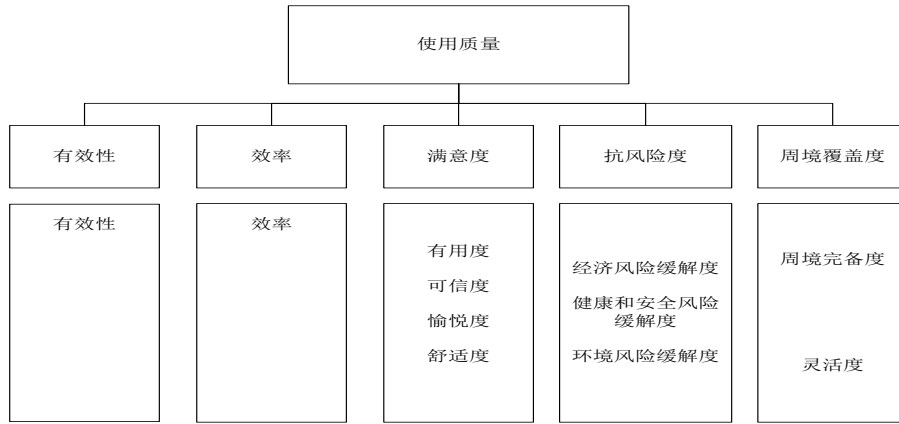


图2 使用质量模型

系统的使用质量体现了产品（系统或软件产品）对利益相关方的影响情况。它由软件、硬件、操作环境的质量、用户、任务及社会环境共同决定。所有的这些因素构成了系统的使用质量。

4.2中给出了使用质量的每个质量特性的定义和解释。

ISO/IEC 25022中给出了使用质量测量的示例。

5.2 使用质量特性说明

5.2.1 有效性

用户实现特定目标的准确性和完备性。

5.2.2 效率

与用户实现目标的准确性和完备性相应的资源消耗。

5.2.3 满意度

产品或系统在指定的使用周境下，使用户的需求得到满足的程度。

注1：对于不直接与产品或系统交互的用户，满意度仅与目标完成情况和可信性相关。

注2：满意度是用户对其与产品或系统交互的反应，还包括对产品使用的态度。

5.2.3.1 有用度

产品或系统使用户对于既定目标的实现感到满意的程度，包括使用的结果和使用后产生的后果。

5.2.3.2 可信度

产品或系统使用户或者其他利益相关方对其行为符合预期有信心的程度。

5.2.3.3 愉悦度

产品或系统使用户因个人需求得到满足而获得愉悦的程度。

注：个人需求包括获得新的知识和技能、个人身份的交流和引发愉快的回忆。

5.2.3.4 舒适度

产品或系统使用户生理上感到舒适的程度。

5.2.4 抗风险度

产品或系统减轻在经济现状、人的生命、健康或者环境方面的潜在风险的程度。

5.2.4.1 经济风险缓解度

产品或系统在预期的使用周境中降低在经济现状、高效运行、商业财产、信誉和其他资源方面的潜在风险的程度。

5.2.4.2 健康和安全风险缓解度

产品或系统在预期的使用周境中降低人员潜在风险的程度。

5.2.4.3 环境风险缓解度

产品或系统在预期的使用周境中降低在财产或者环境方面的潜在风险程度。

5.2.5 周境覆盖度

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、抗风险度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（被称为“特定的条件”）相关。

5.2.5.1 周境完备度

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

注：周境完备性可以被定义或测量为：为达到特定目标，指定用户在使用产品时达到满足有效性、效率、抗风险性和满意度的特定目标的程度，或者通过产品表现出的在所有的预期下都支持使用的产品特性来测量。

示例：由一个非专业人员在小屏幕，低带宽，并且在容错模式（如没有网络连接）下操作时的软件可用程度。

5.2.5.2 灵活度

在超出最初设定需求的周境中，产品或系统在有效性、效率、抗风险度和满意度特性方面能够被使用的程度。

注1：针对额外的用户组、任务和文化，灵活度可通过调整产品（见 6.2.8.1）来达到。

注2：灵活性使软件产品考虑到环境、机会和个人喜好等非预期因素。

注3：如果一个软件产品设计时未考虑灵活性，那么在意外的周境下使用该产品可能是不安全的。

注4：灵活性可以被测量，或者是软件产品使额外的用户类型在额外的使用周境下达到满足有效性、生产率、抗风险性和满意度的额外类型的目标的能力，或者通过对支持产品修改使产品适用于新的用户类型，任务和环境，以及适合 ISO 9241-110 定义的个性化的修改能力的测量。

6 产品质量模型及特性说明

6.1 产品质量模型

产品质量模型将质量属性划分为8个质量特性：功能适合性、性能效率、兼容性、易用性、可靠性、信息安全性、维护性和可移植性，每一个特性由相关的子特性组成（见图4）。

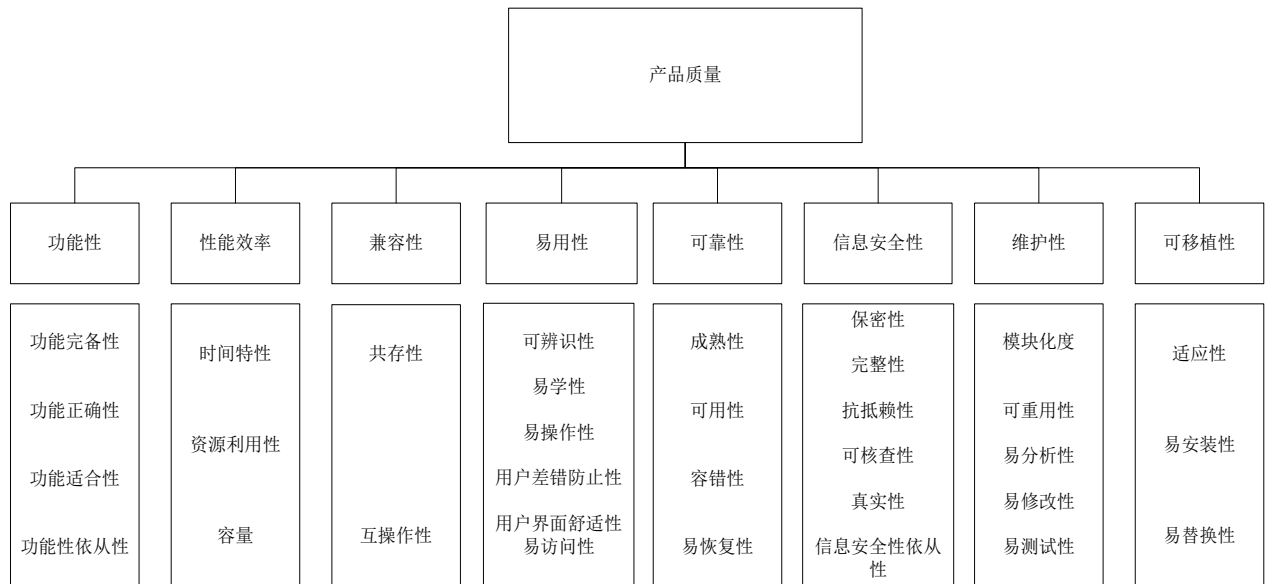


图3 产品质量模型

产品质量模型可以应用于软件产品，或者任何包含软件的计算机系统，因为质量模型中的大部分子特性同时适用于系统与软件。

6.2给出了产品质量的每一个质量特性的定义及解释。

6.2 产品质量特性说明

6.2.1 功能性

在指定条件下使用时，产品或系统的功能满足明确和隐含要求的程度。

6.2.1.1 功能完备性

产品或系统的功能对指定的任务和用户目标的覆盖程度。

6.2.1.2 功能正确性

产品或系统提供具有所需精度的正确结果的程度。

6.2.1.3 功能适合性

产品或系统的功能便于用户实现指定的任务和目标的程度。

示例：用户只需通过必要的步骤就可以完成某项任务，而不需要多余的步骤。

注：功能适合性对应着ISO 9241—110标准中的任务适合性。

6.2.1.4 功能性依从性

产品或系统遵循与功能性相关的标准、约定或法规以及类似规定的程度。

6.2.2 性能效率

性能效率与产品或系统在规定条件下运行时使用的资源数量有关。

注：资源包括其他软件产品、系统的软件和硬件配置、物质材料（如打印纸和存储介质等）。

6.2.2.1 时间特性

产品或系统执行其功能时，响应时间、处理时间及吞吐率符合需求的程度。

6.2.2.2 资源利用性

产品或系统执行其功能时，所使用资源的数量和类别符合需求的程度。

注：人力资源属于效率（5.2.2）的一部分。

6.2.2.3 容量

产品或系统参数最大限度满足需求的程度。

注：参数可包括存储数据项数量、并发用户数、通信带宽、交易吞吐量和数据库规模等。

6.2.3 兼容性

产品、系统或组件在共享同样的硬件或软件环境的条件下，能够和其他产品、系统或组件交换信息，并且/或者实现要求的功能的程度。

6.2.3.1 共存性

在和其他产品共享通用的环境和资源条件下，产品可以有效执行所要求的功能并且不会对其他产品造成负面影响的程度。

6.2.3.2 互操作性

两个及以上的系统、产品或组件之间交换并使用信息的程度。

6.2.4 易用性

在特定的使用周境中，产品或系统具有能使特定用户有效、高效、满意使用的程度。

注：易用性既可以按照产品质量特性和子特性进行描述和测量，也可以按照使用质量的特性和子特性进行描述和测量。

6.2.4.1 可辨识性

产品或系统具有让用户辨识是否符合其要求的程度。

参见功能适合性（6.2.1.3）。

注1：可辨识性需要具有通过产品的初始印象和/或有关的文档对产品或系统功能是否合适进行判断的能力。

注2：产品或系统提供的信息包括演示、指南、文档或网站的主页信息等。

6.2.4.2 易学性

产品或系统具有被特定用户使用并能实现有效果、有效率、无风险、很满意地学习使用产品或系统这一目标的程度。

注：易学性的描述和测量，即可以通过判断产品或系统被特定用户使用并能实现有效果、有效率、无风险、很满意地学习使用产品或系统这一目标的程度来进行，也可以通过ISO 9241-110中产品适合学习的属性来判断。

6.2.4.3 易操作性

产品或系统具有容易操作和控制的属性的程度。

注：易操作性对应ISO 9241—110中可控性、（操作）容错性和用户期望的遵从性。

6.2.4.4 用户差错防止性

系统防止用户犯错误的程度。

6.2.4.5 用户界面舒适性

产品和系统具有在交互过程中，使用户对界面感到愉悦和满足的程度。

注：是指产品或系统可以增强用户乐趣和满意度的属性，例如色彩的使用和图形设计的自然性。

6.2.4.6 易访问性

产品或系统被最大范围用户使用（各种性格和各种能力的用户）并满足特定使用要求和目标的程度。

注1：能力的范围包括与年龄有关的能力障碍。

注2：对具有能力障碍的人而言，易访问性的规定和测量既可以通过产品或系统被特定用户在指定的使用周境中使用并能有效果、有效率、无风险、很满意地实现目标的程度来进行，也可以通过产品属性的存在对易访问性的支持程度来进行。

6.2.5 可靠性

一个系统、产品或者组件在特定环境下、特定的一段时间内完成特定功能的程度。

注：耗损不被用于软件中。可靠性的局限可以归咎于需求、设计和实现阶段中的故障。

6.2.5.1 成熟性

系统、产品或组件在正常运行时满足可靠性要求的程度。

注：成熟性这个概念可以被用于其他质量特性中来表明它们通过正常的操作满足需求的程度。

6.2.5.2 可用性

系统、产品或组件在需要时能够运行和可访问的程度。

注：从外部看，可用性可以通过在上升状态期间的系统、产品或组件总时间比例来进行评估。因此，可用性可以作为成熟性（决定了失效频率）、容错性和易恢复性（决定了每次失效持续的时间）的组合。

6.2.5.3 容错性

在存在故障时，系统、产品或组件的硬件或软件仍可以根据预期运行的程度。

6.2.5.4 易恢复性

在发生了中断或失效的事件下，产品和软件能够直接恢复受损数据并重建正常系统状态的程度。

注：在发生失效后，计算机系统有时会宕机一段时间，这个宕机时间长度由易恢复性决定。

6.2.6 信息安全性

一个产品或系统保护其信息和数据的程度，以使用户或其他产品或系统可以对其进行访问时和类型或授权访问的级别相一致。

注1：需要保护的数据不但包括在系统中存储的数据，还包括在传输中的数据。

注2：存活性（产品或系统在受到攻击后仍旧能够持续提供主要服务并能及时完成任务的程度）包含在易恢复性中（6.2.5.4）。

注3：免疫性（产品或系统避免受到攻击的程度）包含在完整性中（6.2.6.2）。

注4：安全性影响可信度（5.2.3.2）。

6.2.6.1 保密性

产品或系统确保其数据只能被授权用户访问的程度。

6.2.6.2 完整性

系统、产品或组件防止未授权访问、篡改计算机程序或数据的程度。

6.2.6.3 抗抵赖性

活动或事件发生后可以被证实且不可被否认的程度。

6.2.6.4 可核查性

一个实体的活动可以被唯一性地追溯到该实体的程度。

6.2.6.5 真实性

一个主体或资源的身份标识能够被证实和其声明一致程度。

6.2.6.6 信息安全性依从性

系统或软件产品遵循与信息安全性相关的标准、约定或法规以及类似规定的程度。

6.2.7 维护性

约定的维护人员对产品或系统进行调整时，具有效果和效率的程度。

注1：调整包括更正、改进、使软件适应环境变化、需求变化和规格说明书变化。调整包括由专业支持人员进行的调整，也包括由业务人员或操作人员、甚至是最终用户的调整。

注2：维护性包括安装、升级和更新。

注3：维护性既可以作为产品或系统协助维护活动的内部能力，也可以作为维护人员对产品或系统进行维护时所感受到的使用质量。

6.2.7.1 模块化度

由多个独立组件组成的系统或计算机程序，在其一个组件受到修改时对其他组件影响最小化的程度。

6.2.7.2 可重用性

一个资产能够用于多个系统或构建其他资产的程度。

6.2.7.3 易分析性

有效果、有效率地评估行为对产品或系统的影响，这些行为包括修改系统的组成部分、诊断缺陷或失效的原因、标识需要修改的部分等。

注：易分析性的实现包括产品或系统提供分析自身故障的机制，并能够在失效或其他事件发生前提供报告。

6.2.7.4 易修改性

产品或系统能进行有效果、有效率的修改并且不会引入缺陷或降低已有产品质量的程度。

注1：易修改性的实现包括编码、设计、文档和验证变更。

注2：模块化度（6.2.7.1）和易分析性（6.2.7.3）会影响到可修改性。

注3：易修改性是易改变性和稳定性的组合。

6.2.7.5 易测试性

系统、产品或组件能够有效果、有效率地建立测试基准的程度，以及测试是否能有效地执行以确定测试基准是否被满足。

6.2.8 可移植性

系统、产品或组件能够有效果、有效率地从一种硬件、软件、运行环境或使用环境转移到另一种上的程度。

注：可移植性可以作为产品或系统协助移植活动的内部能力，或移植人员对产品或系统进行移植时体验到的使用质量。

6.2.8.1 适应性

产品或系统能够有效果、有效率地适应不同的或演变的硬件、软件、运行环境或使用环境的程度。

注1：适应性包括内部能力的可测量性（屏幕区域、表、事务容量、报告格式等）。

注2：适应性包括由专业支持人员进行的操作，也包括业务人员、操作人员或终端用户等的操作。

注3：如果考虑到系统对最终用户的适应性，适应性可以和 ISO9241—110 中规定的个性化适合性相对应。

6.2.8.2 易安装性

产品或系统在特定环境中能够有效果、有效率地进行安装和/或卸载的程度。

注：如果系统或产品由最终用户安装，易安装性会影响到相关的功能合适性和易操作性。

6.2.8.3 易替换性

一个软件产品在相同作用相同环境的条件下能被另外一个软件产品替换的程度。

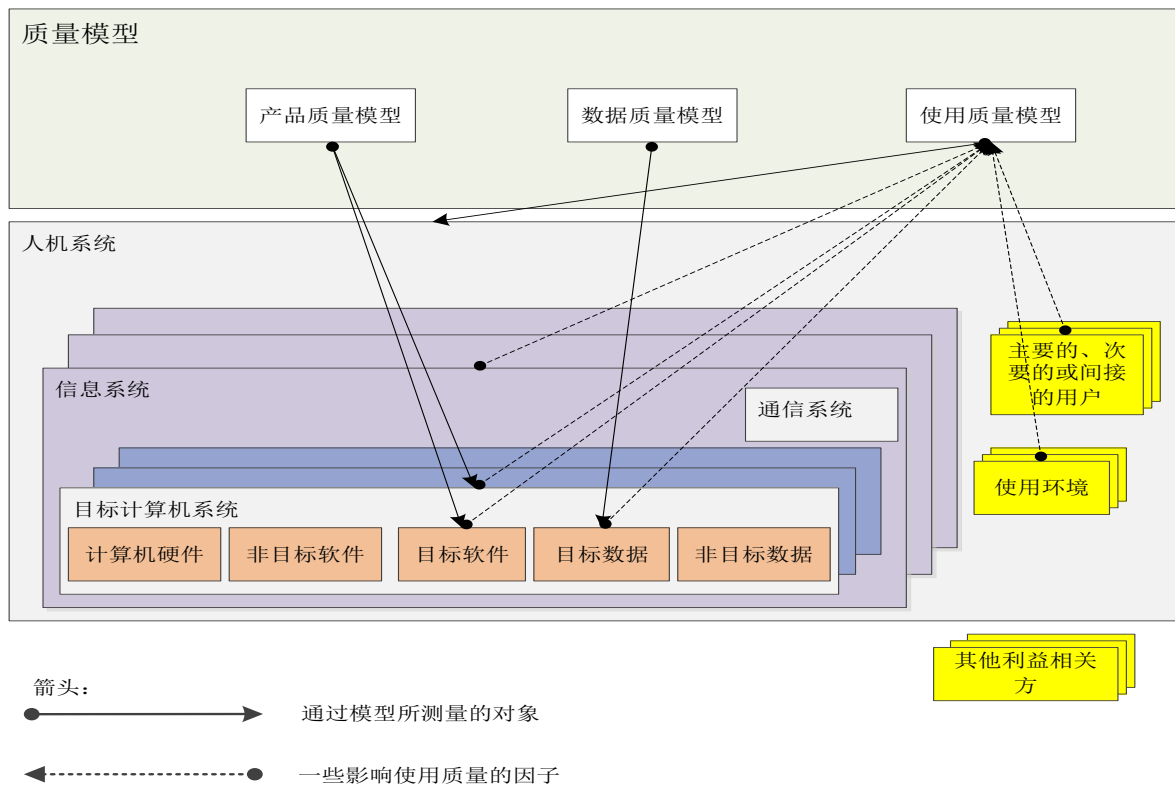
注1：在升级时，软件产品替换成为新版本对用户非常重要。

注2：易替换性也包括了易安装性和适应性的一些属性，把易替换性独立作为一个子特性是因为其重要性。

注3：易替换性能降低锁死风险：可以用另外的产品替换目前的产品，例如使用标准文档格式。

7 质量模型的目标

图5描述了质量模型的目标以及相关的实体。



注：这与ISO/IEC 25012的图2和ISO/IEC 25030的图5在概念上相同，但是它是更侧重于质量模型的不同版本。

图4 质量模型的目标

产品质量模型适用于包含了目标软件的目标计算机系统，使用质量模型适用于包含了目标计算机系统和目标软件产品的人机系统。目标计算机系统中包含了计算机硬件、非目标软件产品、非目标数据以及目标数据（是数据质量模型的适用对象，参见C.7）。目标计算机系统可以是整个信息系统的一个部分，信息系统还可以包括其他的计算机系统、通信系统、局域网和广域网等，而信息系统是更大的人机系统的一个部分（如企业系统、嵌入式系统或者大规模的控制系统等），该系统还包括了用户、技术人员以及物理使用环境。系统的边界需要通过需求或评价的范围来界定，并依赖于使用的用户。

示例：如果一架拥有计算机控制飞行系统的飞机的用户被定义为乘客，则乘客所依赖的系统就包括了机组人员、驾驶员、以及飞行控制系统的硬件和软件，而如果机组人员被定义为用户，则其所依赖的系统就仅包括了驾驶员和飞行控制系统。

其他的利益相关方都与质量有着密切的关系，如软件开发方、系统集成方、需方、所有者、维护者、合同方、质量保证和控制人员等。

8 质量模型的使用

产品质量模型和使用质量模型可以帮助细化质量需求、建立质量测量、执行质量评价（参见附录B）。已定义好的质量特性可以作为质量需求完善程度的检查表，并可以成为估算系统开发最终工作量和活动的一个依据。使用质量模型中的质量特性和产品质量模型中的质量特性可以作为一个质量特性集合共同用于细化或评价计算机系统或软件产品质量。

在一个大的计算机系统或软件产品中使用或度量所有的质量特性是不切实际的，同样，使用质量模型也并不能应用于所有的用户任务场景。质量特性的重要程度需要依据项目的目标层次来确

定，因此宜把模型进行裁剪后再作为需求分解的一个组成部分，以此来确定哪些质量特性和子特性是最重要的，并根据产品的目标和各利益相关方的目的来分配质量测量所需要的资源。

9 来自不同利益相关方视角的质量

质量模型是一个汇集各利益相关方要求的框架。各利益相关方包括以下的用户：

- a) 主要用户：与系统进行交互并达到主要目标的人。
- c) 次要用户：提供支持的人，例如：
 - 1) 内容提供者，系统管理人员，安全管理人员；
 - 2) 维护人员，分析人员，移植人员，安装人员。
- d) 间接用户：接收系统的输出，但不与系统进行交互的人。

表1 使用质量模型和产品质量模型的用户要求示例

用户要求	主要用户	次要用户		间接用户
		内容提供者	维护人员	
	交互	交互	维护或移植	使用输出
有效性	用户使用系统完成任务时的有效性要求	内容提供者在更新系统时的有效性要求	在对系统进行维护或移植时的有效性要求	在使用系统的输出时的有效性要求
效率	用户使用系统完成任务时的效率要求	内容提供者在更新系统时的效率要求	在对系统进行维护或移植时的效率要求	在使用系统的输出时的效率要求
满意度	用户使用系统完成任务时的满意度要求	内容提供者在更新系统时的满意度要求	在对系统进行维护或移植时的满意度要求	在使用系统的输出时的满意度要求
抗风险性	用户使用系统完成任务时的抗风险性要求	内容提供者在更新系统时的抗风险性要求	在对系统进行维护或移植时的抗风险性要求	在使用系统的输出时的抗风险性要求
可靠性	用户使用系统完成任务时的可靠性要求	内容提供者在更新系统时的可靠性要求	在对系统进行维护或移植时的可靠性要求	在使用系统的输出时的可靠性要求
信息安全性	用户使用系统完成任务时的信息安全性要求	内容提供者在更新系统后的信息安全性要求	在对系统进行维护或移植后的信息安全性要求	系统的输出的信息安全性要求
周境覆盖度	在确定的或潜在的使用周境中，系统能达到的有效、效率、安全、满意的程度	在确定的或潜在的使用周境中，内容提供所能达到的有效、效率、安全、满意的程度	在确定的或潜在的使用周境中，维护或移植系统所能达到的有效、效率、安全、满意的程度	在确定的或潜在的使用周境中，使用系统的输出所能达到的有效、效率、安全、满意的程度
易学性	学习使用系统时所能达到的有	学习提供系统内容时所能达到的	学习维护系统或移植系统时所能	学习使用系统的输出时所能达到

	效、效率、安全、满意的程度	有效、效率、安全、满意的程度	达到的有效、效率、安全、满意的程度	的有效、效率、安全、满意的程度
可访问性	在残障人士使用系统时所能达到的有效、效率、安全、满意的程度	在为残障人士提供系统内容时所能达到的有效、效率、安全、满意的程度	在为残障人士进行系统维护或移植时所能达到的有效、效率、安全、满意的程度	在残障人士使用系统的输出时所能达到的有效、效率、安全、满意的程度

上述每个类型的用户在特定环境中使用系统时都需要考虑到使用质量和产品质量，表1提供了用户和质量特性的例子。

注：内容提供者也需要考虑到数据质量的要求。

表 1 中的例子只是用户需求的基本要求，可以作为系统在使用和维护时对质量影响程度的测量基准。

在软件开发或获取之前，就要对各利益相关方的质量要求进行定义，对使用质量要求的分析可以得到产品功能性需求和质量需求，为产品满足使用质量要求奠定基础。

示例：系统可靠性的要求可以导出软件产品成熟性、可用性、容错性和可恢复性的详细需求说明，而系统可靠性也对系统整体的有效性、效率、抗风险性和满意度有很大的影响。

10 各质量模型间的关系

软件产品及计算机系统的特点决定了产品质量受到使用周境的影响（见表2）。

功能适合性、性能效率、易用性、可靠性、信息安全性会对系统的主要用户的使用质量产生极其重要的影响，性能效率、可靠性和信息安全性也会对其他利益相关方在某些领域中产生一定的影响。

兼容性、维护性和可移植性则会对对系统进行维护的次要用户的使用质量产生非常大的影响。

表2 质量特性的影响程度

软件产品属性	计算机系统属性	产品质量特性	对主要用户使用质量的影响	对维护任务质量的影响	信息系统质量对其他利用相关方的影响
○	○	功能适合性	<input type="checkbox"/>		
○	○	性能效率	<input type="checkbox"/>		<input type="checkbox"/>
○	○	兼容性		<input type="checkbox"/>	
○	○	易用性	<input type="checkbox"/>		
○	○	可靠性	<input type="checkbox"/>		<input type="checkbox"/>
○	○	信息安全性	<input type="checkbox"/>		<input type="checkbox"/>
○	○	维护性		<input type="checkbox"/>	
○	○	可移植性		<input type="checkbox"/>	

注：○这些属性影响产品质量。

□产品质量影响这些利益相关方的使用质量。



附录 A (资料性附录)

与 GB/T 16260.1-2006 中的质量模型的对比

GB/T25000 的本部分修订了 GB/T 16260.1-2006。

使用质量模型由 4 大特性修改调整为 5 大特性，具体修订细节如下：

- 删除了安全性特性；
- 增加了抗风险度及其子特性（经济风险缓解度、健康和安全风险缓解度、环境风险缓解度）和周境覆盖度及其子特性（周境完备度、灵活度）。

产品质量模型由 6 大特性修改调整为 8 大特性，具体修订细节如下：

- 将原功能性的一个子特性安全保密性提升为一个独立的特性，并更名为信息安全性，其子特性包括保密性、完整性、抗抵赖性、可核查性、真实性和信息安全性依从性；
- 增加了兼容性特性，其子特性包括共存性和互操作性；
- 在子特性方面增加了几个子特性：功能完备性、容量、用户差错预防性、可访问性、有用性、模块化度和可重用性；
- 删去了性能效率、兼容性、易用性、可靠性、维护性和可移植性的依从性；
- 几个特性和子特性给出了更精确的名称；
- 为了符合 ISO / IEC 9126-1 中的标准或法规的子特性，在质量模型之外的范围可以被定义为对系统需求的一部分。
- 恰当的时候，通用的定义被引入到扩展的计算机系统范围之内，而不是采用特定的软件的定义。

表 A.1 列出了本部分中质量特性和子特性与 GB/T 16260.1-2006 中的对比结果。

表A.1 本标准与 GB/T 16260.1-2006 中的模型的比较

条款	GB/T 25000.10	GB/T 16260.1	注释
5.1	使用质量	使用质量	现在出现的使用质量本质上是一个系统质量
5.2.1	有效性	有效性	
5.2.2	效率	生产率	名字与 ISO/IEC 25062 和 ISO/IEC 9241-11 相对应
5.2.3	满意度	满意度	
5.2.3.1	有用度		新增的子特性
5.2.3.2	可信度		新增的子特性
5.2.3.3	愉悦度		新增的子特性
5.2.3.4	舒适度		新增的子特性
5.2.4	抗风险度	安全性	
5.2.4.1	经济风险缓解度		新增的子特性
5.2.4.2	健康和安全风险缓解度		新增的子特性
5.2.4.3	环境风险缓解度		新增的子特性
5.2.5	周境覆盖度		新增的特性。明确了隐含的质量问题

表 A.1 本标准与 GB/T 16260.1-2006 中的模型的比较 (续)

条款	GB/T 25000.10	GB/T 16260.1	注释
5.2.5.1	周境完备度		新增的子特性(产品能够在所有需要的周境中使用至关重要)
5.2.5.2	灵活度		新增的子特性(保证产品能够在新的使用周境中使用)
6.1	产品质量	内部和外部质量	内部和外部质量合并称为产品质量
6.2.1	功能性	功能性	
6.2.1.1	功能完备性		新增的子特性。覆盖了明确的要求
6.2.1.2	功能正确性	准确性	比准确性更通用
6.2.1.3	功能适合性	适合性	覆盖了隐含的要求
		互操作性	移至兼容性中
		安全保密性	提升为新的特性, 更名为“信息安全性”
6.2.1.4	功能性依从性	功能性依从性	
6.2.2	性能效率	效率	重命名是为了避免与 ISO/IEC 25062 中的效率的定义相互冲突
6.2.2.1	时间特性	时间特性	
6.2.2.2	资源利用率	资源利用性	
6.2.2.3	容量		新增的子特性(特别是与计算机系统相关)
		效率依从性	删除了该子特性
6.2.3	兼容性		新增的特性
6.2.3.2	共存性	共存性	源于可移植性
6.2.3.3	互操作性		源于功能性
6.2.4	易用性		明确了隐含的质量问题
6.2.4.1	可辨识性	易理解性	新的名字更加准确
6.2.4.2	易学性	易学性	
6.2.4.3	易操作性	易操作性	
6.2.4.4	用户差错防止性		新增的特性(对于能够规避风险尤其重要)
6.2.4.5	用户界面舒适性	吸引力	新的名字更加准确
6.2.4.6	易访问性		新增的子特性
		易用性的依从性	删除了该子特性
5.2.5	可靠性	可靠性	
5.2.5.1	成熟性	成熟性	
5.2.5.2	可用性		新增的子特性
4.2.5.3	容错性	容错性	
5.2.5.4	易恢复性	易恢复性	
		可靠性的依从性	删除了该子特性
5.2.6	信息安全性	安全性	新增的子特性
5.2.6.1	保密性		新增的子特性
5.2.6.2	完整性		新增的子特性
5.2.6.3	抗抵赖性		新增的子特性

表 A.1 本标准与 GB/T 16260.1-2006中的模型的比较（续）

条款	GB/T 25000.10	GB/T 16260.1	注释
5.2.6.4	可核查性		新增的子特性
5.2.6.5	真实性		新增的子特性
5.2.6.6	信息安全性依从性		新增的子特性
5.2.7	维护性	维护性	
5.2.7.1	模块化度		新增的子特性
5.2.7.2	可重用性		新增的子特性
5.2.7.3	易分析性	易分析性	
		易改变性	
5.2.7.4	易修改性	稳定性	将易改变性和稳定性相结合的更准确的名称
5.2.7.5	易测试性	易测试性	
		维护性的依从性	删除了该子特性
5.2.8	可移植性	可移植性	
5.2.8.1	适应性	适应性	
5.2.8.2	易安装性	易安装性	
		共存性	移至兼容性
5.2.8.3	易替换性	易替换性	
		可移植性的依从性	删除了该子特性

附录 B
(资料性附录)
与 ISO/IEC 25010:2011 的差异

GB/T25000 的本部分修改采用国际标准 ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation(SQuaRE) - System and software quality models。

基于符合中国国情的需要，本部分对原国际标准的内容结构进行了调整。因此，本部分与国际标准的一致性程度为修改采用。除编辑性修改外，主要技术变化如下：

- a) 增加了规范性引用文件 GB/T 25000.1《软件工程 软件产品质量要求和评价(SQuaRE) SQuaRE 指南》，并在第 3 章中引出；
- b) 将国际标准 ISO/IEC 25010:2011 中的术语和定义进行了整合；
- c) 本部分的产品质量模型与原国际标准基本一致，但是本部分结合国情增加了两个子特性，即功能性依从性和信息安全性依从性，进一步体现了 SQuaRE 国家系列标准的协调性一致性；
- d) 删除了国际标准 ISO/IEC 25010:2011 附录 B 的内容，并将本部分与原国际标准的修订差异作为资料性附录在附录 B 中进行了说明；
- e) 本部分不针对软件属性进行阐述，因此删除了国际标准 ISO/IEC 25010:2011 中的 C.6 及其相关内容。

本部分内容结构的调整情况详见表B.1。

表B.1 结构调整说明

ISO/IEC 25010:2011	调整说明
Foreword	即“前言”
Introduction	即“引言”
1 Scope	即“1 范围”
2 Conformance	删除该条款，并新增“2 规范性引用文件”
3 Quality model framework	
3.1 Quality models	提升至“4 质量模型框架”
3.2 Quality in use model	变更为“5.1 使用质量模型”，与 5.2 构成“5 使用质量模型及特性说明”
3.3 Product quality model	变更为“6.1 使用质量模型”，与 6.2 构成“6 产品质量模型及特性说明”
3.4 Targets of the quality models	提升至“7 质量模型的目标”
3.5 Using a quality model	提升至“8 质量模型的使用”
3.6 Quality from different stakeholder perspectives	提升至“9 来自不同利益相关方视角的质量”
3.7 Relationship between the models	提升至“10 各质量模型间的关系”
4 Terms and definitions	
4.1 Quality in use model	变更为“5.2 使用质量特性说明”，与 5.1 构成“5 使用质量模型及特性说明”

表 B.1 结构调整说明（续）

ISO/IEC 25010:2011	调整说明
4.2 Product quality model	变更为“6.2 产品质量特性说明”，与 6.1 构成“6 产品质量模型及特性说明”
4.3 General	与 4.4 General 合并为“3 术语和定义”
4.4 Terms and definitions from ISO/IEC 25000	与 4.3 Terms and definitions from ISO/IEC 25000 合并为“3 术语和定义”
Annex A(informative) Comparison with the quality model in ISO/IEC 9126-1	即“附录 A（资料性附录）与 GB/T 16260.1-2006 中的质量模型的对比”
Annex B(informative) Example of mapping to dependability	删除该条款，并新增“附录 B（资料性附录）与 ISO/IEC 25010:2011 的差异”
Annex C(informative) Using the quality model for measurement	即“附录 C（资料性附录）使用质量模型测量”，删除了 C.6 Software properties
Bibliography	即“参考文献”

附录 C
(资料性附录)
使用质量模型测量

C.1 概述

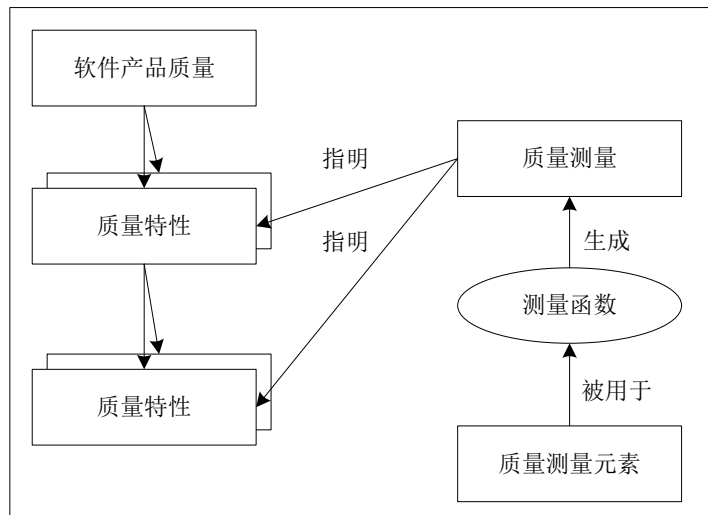
本附录中的信息可能迁移到SQuaRE系列标准的其他部分的未来版本中。

C.2 软件质量测量模型

质量属性是软件所固有的属性，有助于改进软件的质量。质量属性可以归类为一个或多个(子)特性。

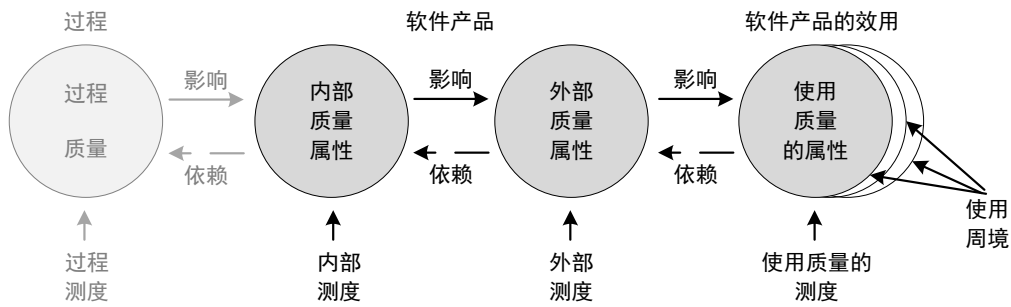
质量属性通过测量方法来测量。测量方法是用来量化属性的逻辑操作序列(相对于特定的标度)。测量方法的结果被称为质量测量元素。质量特性和子特性可以通过测量函数来量化。度量函数是一个用来联系质量测量元素的函数。应用测量函数的结果被称为软件质量测量。这样说来，软件质量测量成为对质量特性和子特性的量化。可以使用多个软件质量测量来测量一个特性或子特性。

图C.1来自ISO/IEC 25020展示了本部分中的质量模型、ISO/IEC 2502n中的测量和ISO/IEC 15939中提出的测量模型间的关系。



图C.1 软件产品质量测量参考模型

C.3 质量途径



图C.2 生存周期中的质量

用户质量要求包括在指定的使用周境下对系统的使用质量的需求。当使用软件产品质量特性和子特性来说明外部质量和内部质量的时候，可以使用这些被确定的要求。

软件产品质量可以通过测量内部属性（典型地是对中间产品的静态测量），也可以通过测量外部属性（典型地是测量代码执行时的行为），或者通过测量使用质量的属性（当产品在实际或模拟中使用）来评价（见图C.2）。

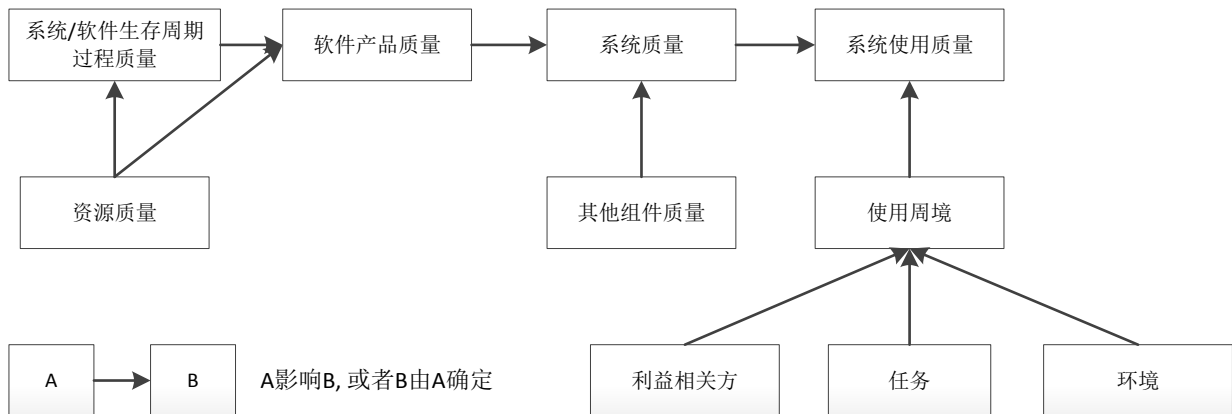
提高过程质量（在ISO/IEC 12207和ISO/IEC 15288中定义的任一生存周期过程的质量）有助于提高产品质量，而产品质量又有助于提高系统使用质量。因此，评估和改进一个过程是提高产品质量的一种手段，而评价和改进产品质量则是提高系统使用质量的方法之一。同样，评价系统使用质量可以为改进产品提供反馈，而评价产品则可以改进为过程提供反馈。

合适的内部属性是获得所需外部行为的先决条件，而适当的外部行为则是获得使用质量的先决条件（见图C.2）。

C.4 质量影响

图C.3说明了质量模型的目标实体之间的关系。软件生存周期过程（如质量需求过程、设计过程和测试过程）影响软件产品和系统的质量。资源的质量，例如人力资源和用于生存周期过程的工具及技术影响过程质量，进而也影响产品质量。

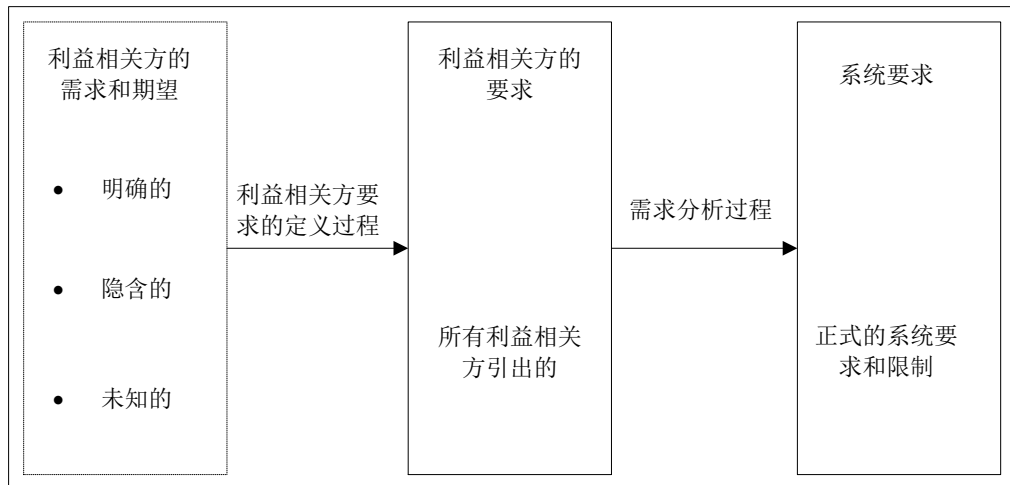
不仅软件产品的质量，而且一个系统的其他组件的质量也影响系统的质量。基于不同的使用周境，系统质量会受到各种各样的影响（效果）。使用周境可以定义为一个用户、任务和环境的集合。在表1中列出了一些使用周境的例子（见第9章）。



图C.3 质量模型的目标实体及其相互间的关系

C.5 质量生存周期模型

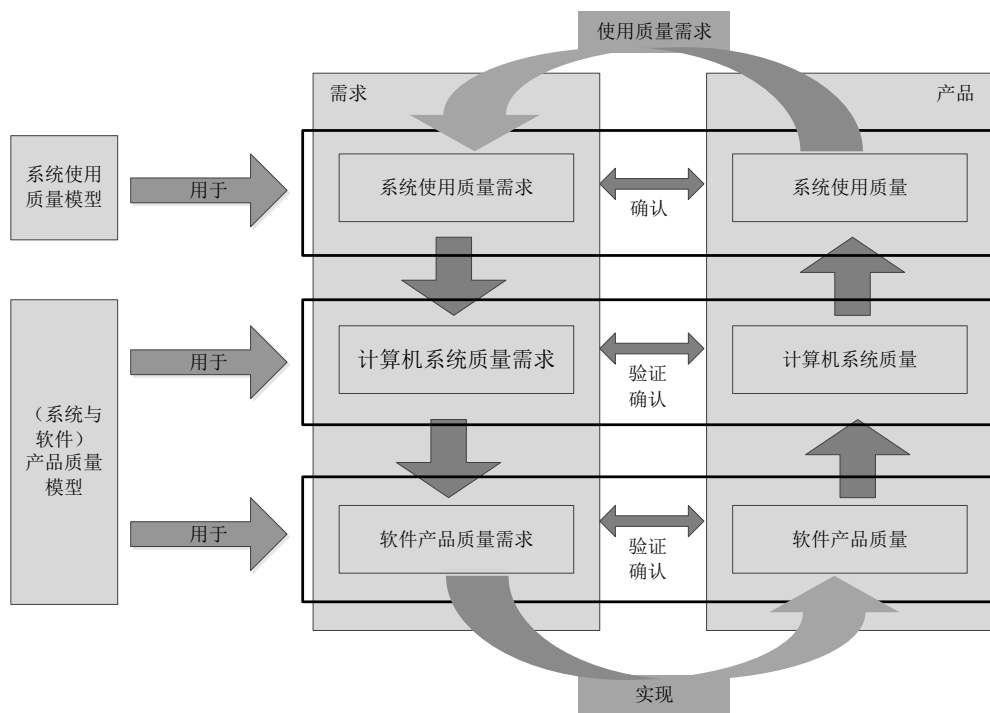
ISO/IEC 25030通过图形解释了质量要求的过程（见图C.4）。图中的“利益相关方的需求”集成了使用质量和产品质量的需求，然后转化为指定的质量要求（利益相关方的要求）。



图C.4 利益相关方要求的定义和分析

质量生存周期模型（见图C.5）把质量放在软件产品生存周期的三个主要阶段。

- 开发阶段的产品即是软件质量内部度量的主体。
- 测试阶段的产品即是软件质量外部度量的主体。
- 使用阶段的产品即是使用质量的主体。



图C.5 系统/软件质量生存周期模型

系统/软件质量生存周期模型还表明，要达到可接受的质量水平，这是一个对各部分对每一种质量：

需求、实现和结果的验证的开发过程的累积。

使用质量要求从用户角度指定了质量需求等级。这些要求来源于用户和其他利益相关方（如软件开发、系统集成商、需求者或拥有者）的需求。使用质量要求被用户作为软件产品验证的目标。当对产品进行评价时，使用质量特性要求应该在质量需求规格说明中，按照使用质量度量标准进行声明。

注1：系统使用质量要求有助于识别和定义外部软件质量要求。

示例1：指定类型的用户能够在指定的时间完成指定的任务。

计算机系统质量的外部度量要求基于外部视角规定了质量需求等级。涵盖了利益相关方的质量要求，包括使用质量要求。外部质量要求被用来作为技术验证的目标和对软件产品进行验证。当对产品进行评价时，外部质量度量要求应该在质量需求规格说明中，按照外部度量标准进行定量声明。

注2：外部度量要求有助于识别和定义软件质量内部度量要求。

注3：外部质量评价可以用来预测系统使用质量。

示例1：用户妥善应对错误信息并成功撤销错误。

软件质量内部度量要求基于内部视角规定了质量需求等级。他们涵盖了外部的质量要求。内部质量需求测量被用来指定中间产品的性能（规格说明、源代码等）。内部软件质量需求也可以被用来指定交付非执行软件产品例如文件和手册的质量。内部软件质量需求测量可以被用来作为各个开发阶段的验证的目标。他们也可以用来定义在开发过程策略、评估的规范和验证。

注4：软件质量内部度量可以被用来预测外部软件质量测量。

示例2：对所有的错误信息指定纠正行为，而且所有的用户输入可以被撤消。

ISO/IEC 25030 描述了软件质量要求，ISO/IEC 25040 描述了软件质量评价过程。

在开发过程中，模型及其相关的度量可以用来管理设计和实施活动来确保达到质量目标。一个质量模型及其相关度量的关键应用是能够较早地洞察软件质量情况。这种洞察力可以用来管理生命周期过程中质量和预测是否满足使用质量。

注5：在有合同的环境或规范的环境中，例如核安全领域，要求是特定的；而在其他的环境中，则应对隐含需求进行识别和定义（ISO 8402:1994, 2.1, 注1）。

C.6 内部测量，外部测量和使用质量的测量

对每一个子特性来说，软件的生产率由一系列的可测量的固定的内部属性决定。GB/T 16260.3 给出了内部测量的实例（将被 GB/T 25000.23 取代）。这些特性及其子特性可以用系统内包含的软件所能提供的容量的程度来测量。GB/T 16260.2 给出了外部测量的实例（将被 GB/T 25000.23 取代）。

系统/软件质量的外部度量向系统/软件提供了一种“黑盒子”的观点并处理与计算机操作系统和硬件上的软件相关的属性。软件质量的内部测量向软件提供了一种“白盒子”的观点并调整了通常可在开发过程中进行评价的软件产品的静态属性。软件的内部测量质量会对系统/软件的外部测量的质量产生影响，同样也会对系统的使用质量产生影响。

基于检测静态属性的内部测量可以用来测量一个软件工作产品的内部属性（见表C.1）。静态分析方法包括检测和自动分析工具。工作产品包括需求和设计文档，源代码，和测试程序。

对动态属性的外部测量可以用来度量计算机系统的外部属性（目标计算机系统见图5），和软件产品的系统相关特性。

使用质量测量（观察或测试真实或模拟使用得到的结果）用来测量系统的内部属性，系统可以包含软件，硬件，通信和用户，以及一个软件密集型系统或者软件产品上的系统相关特性。使用质量测量关系到一个系统对其利益相关方的影响。

表C.1 内部质量测量，外部质量测量以及使用质量测量的差别

属性测量类型	软件产品特性	计算机系统的行为特性	人机系统的影响特性
质量测量类型	内部：静态特性检测	外部：动态特性的检测和建模	使用质量：对实际或模拟使用的结果的测试或观察
软件产品特性类型	固有的	计算机系统相关的	人机系统相关的
计算机系统特性类型		固有的	人机系统相关的
人机系统特性类型			固有的

软件的内部测量可以在开发过程的早期用来预测软件/系统外部测量的质量。同一特性经常同时用外部测量和内部测量来测量，例如，一个内部测量估计预期的响应时间，用以预测外部的时间测量。

GB/T 16260.2 和 GB/T 16260.3 给出了软件产品质量测量的实例（将被 GB/T 25000.23 和 GB/T 25000.22 取代）。

C.7 产品质量和数据质量的关系

ISO/IEC 25012 中的数据质量模型是对产品质量模型的补充。

- 固有的数据质量（见表 C.2）和软件质量的内部测量都有助于衡量计算机系统的总体质量。
- 系统相关数据质量的测量和外部软件质量的测量在对计算机系统的评估方面类似。不同的是系统相关数据质量的测量关注的是数据对计算机系统质量做出的贡献，外部软件质量测量关注的是软件的贡献。但二者实际上测量的都是计算机系统的特性。

表C.2 数据内部属性和计算机系统特性间的关系

属性测量的类型	内部数据质量	计算机系统特性
质量测量的类型	内部数据质量	系统相关数据质量
数据质量的类型	固有的	系统相关的
计算机系统特的类型		固有的

参 考 文 献

- [1] GB/T 16260.1-2006 软件工程 产品质量 第1部分:质量模型(ISO/IEC 9126-1:2001, IDT)
 - [2] GB/T 25000.1-2006 软件工程 软件产品质量要求和评价(SQuaRE) SQuaRE指南(ISO/IEC 25000:2010, IDT)
-