



中山大學
SUN YAT-SEN UNIVERSITY

Chapter 1

Overview of Software Engineering

Software Testing: Approaches & Technologies

School of Data & Computer Science, Sun Yat-sen University

Outline

- 1.1. 软件与软件危机
- 1.2. 软件开发
- 1.3. 软件生命周期
- 1.4. 软件质量模型
- 1.5. 敏捷开发



1.4 软件质量模型

1.4.1 软件质量的概念

- 软件质量是软件工程开发的关键问题，也是软件工程生产中的核心问题
 - 软件质量问题是导致软件项目进度延误、预算超支或项目失败、项目终止等软件危机的根本原因。
 - 提高软件质量可以降低项目开发的总成本，如：降低维护成本并延长软件的生命期、降低软件失效导致的成本损失等。
 - 通过减少并及早检测纠正实际的软件开发过程和软件开发结果与预期不符的情况，降低错误成本。

1.4 软件质量模型

1.4.1 软件质量的概念

- 软件质量的定义
 - 现代质量管理中，质量被认为是“用户的满意程度”
 - ANSI/IEEE Std 729-1983，软件质量是“与软件产品满足规定的和隐含的需求能力有关的特征和特性的全体”。
 - ISO/IEC 9001-1999，软件质量是“软件产品满足需求的能力”。
 - 1. 软件产品中能满足给定需求的性质和特性的总体（例如符合规格说明）。
 - 2. 软件具有所期望的各种属性组合的程度。
 - 3. 顾客或用户觉得软件满足其综合期望的程度。
 - 4. 软件的合成特性，它确定软件在使用中将满足顾客预期要求的程度。

1.4 软件质量模型

1.4.1 软件质量的概念

- 软件质量的定义 (续)
 - A. V. Feigenbaum (Total Quality Control, 1983)
 - ✧ Quality is a customer determination, not an engineer's determination, not a marketing determination, nor a general management determination. It is based on the customer's actual experience with the product or service, measured against his or her requirements -- stated or unstated, conscious or merely sensed, technically operational or entirely subjective -- and always representing a moving target in a competitive market.

1.4 软件质量模型

1.4.1 软件质量的概念

- 看待软件质量的不同观点
 - 先验论观点
 - ✧ 质量是产品一种可以认识但不可定义的性质
 - 用户观点
 - ✧ 质量是产品满足使用目的之程度
 - 制造者观点
 - ✧ 质量是产品性能和规格要求的符合度
 - 产品观点
 - ✧ 质量是联结产品固有性能的纽带
 - 价值观点
 - ✧ 质量依赖于顾客愿意付给产品报酬的数量

1.4 软件质量模型

1.4.1 软件质量的概念

- 软件质量范畴 (3A)
 - Accountability (可说明性)
 - ✧ 用户可以基于产品或服务的描述和定义来使用产品
 - 例如: 市场需求说明书, 功能设计说明书
 - Availability (有效性)
 - ✧ 产品或服务对于 99.999% 的客户总是有效的
 - 例如: 性能测试和恢复测试
 - Accessibility (易用性/可用性)
 - ✧ 对于用户, 产品或服务容易使用并且具备有用的功能
 - 例如: 确认测试和用户可用性测试

1.4 软件质量模型

1.4.1 软件质量的概念

- 高质量软件标准体系

- 产品质量

- ✧ 是人们实践产物的属性和行为，是可以认识和可以科学描述的。并且可以通过一些方法和人类活动，来进行改进。

- ✧ 质量模型: McCall 模型, Boehm 模型, ISO 9126 模型

- 过程质量

- ✧ 软件能力成熟度模型 CMM (Capability Maturity Model).

- ✧ 国际标准过程模型 ISO 9000

- ✧ 软件过程改进和能力决断 SPICE (Software Process Improvement and Capability Determination)

1.4 软件质量模型

1.4.1 软件质量的概念

- 软件产品质量标准
 - 功能性 Functionality
 - ✧ 软件所实现的功能达到它的设计规范和满足用户需求的程度
 - 可用性 Usability
 - ✧ 对于一个软件，用户学习、操作、准备输入和理解输出所作的努力的程度 (安装简单；使用轻松；界面友好)
 - 可靠性 Reliability
 - ✧ 在规定的的时间和条件下，软件所能达到保持其正常的功能操作、性能水平的程度
 - 性能 Performance
 - ✧ 在指定条件下，用软件实现某种功能所需的计算机资源的有效程度

1.4 软件质量模型

1.4.1 软件质量的概念

- 软件产品质量标准 (续)
 - 能力/容量 Capacity
 - ✧ 系统的接受力、容纳或吸收的能力，或某项功能的最大量或最大限度
 - 可测量性 Scalability
 - ✧ 系统某些特性可以通过一些量化的数据指标描述其当前状态或理想状态
 - 可维护性 Service manageability
 - ✧ 在一个运行软件中，当环境改变或软件发生错误时，进行相应修改所做的努力的程度
 - 兼容性 Compatibility
 - 可扩展性 Extensibility

1.4 软件质量模型

1.4.2 软件质量特性

- 软件质量的三种不同视面
 - 使用者 (用户) 角度
 - ✧ 软件使用、软件性能和软件效果
 - 是否具有所需要的功能、可靠程度如何、效率如何、使用是否方便、环境开放的程度如何 (对环境、平台的限制, 以及与其他软件连接的限制)
 - 开发者角度:
 - ✧ 中间过程的质量
 - ✧ 最终产品的质量

1.4 软件质量模型

1.4.2 软件质量特性

- 软件质量的三种不同视面 (续)
 - 管理者角度
 - ◇ 质量的总体，而不是某一特性
 - ◇ 从管理角度对软件质量进行度量，把影响软件质量的主要因素划分为三类，分别反映用户在使用软件产品时的三种观点：
 - 产品运行：正确性、健壮性、效率、完整性、可用性、风险
 - 产品修改：可理解性、可维修性、灵活性、可测试性
 - 产品转移：可移植性、可重用性、互运行性
 - 软件质量特性是从使用者的观点引入的面向管理的观点，它的实际价值在于它体现了用户的观点。

1.4 软件质量模型

1.4.2 软件质量特性

- 软件质量特性的定义
 1. 功能性 Functionality
 - ✧ 软件所实现的功能达到它的设计规范和满足用户需求的程度
 2. 效率 Efficiency
 - ✧ 在规定条件下，用软件实现某种功能所需的计算机资源 (包括时间) 的有效程度
 3. 可靠性 Reliability
 - ✧ 在满足一定条件的应用环境中，软件能够正常维持其工作的能力
 4. 安全性 Safety&Security
 - ✧ 为了防止意外或人为的破坏，软件应具备的自身保护能力

1.4 软件质量模型

1.4.2 软件质量特性

- 软件质量特性的定义 (续)

- 5. 易使用性 Usability

- ✧ 用户在学习、操作和理解软件使用的过程中所需要付出的努力的程度

- 6. 可维护性 Maintainability

- ✧ 当环境改变或软件运行发生故障时，为使其恢复正常运行所需要付出的努力的程度

- 7. 可扩充性 Extensibility/Scalability

- ✧ 在功能改变和扩展情况下，软件能够正常运行的能力

- 8. 可移植性 Portability

- ✧ 将软件从现有运行平台向另一个运行平台过度所需要付出的努力的程度

1.4 软件质量模型

1.4.2 软件质量特性

- 软件质量特性的定义 (续)

9. 可重用性 Reusability

- ✧ 整个软件或其中一部分能作为软件包而被再利用的程度

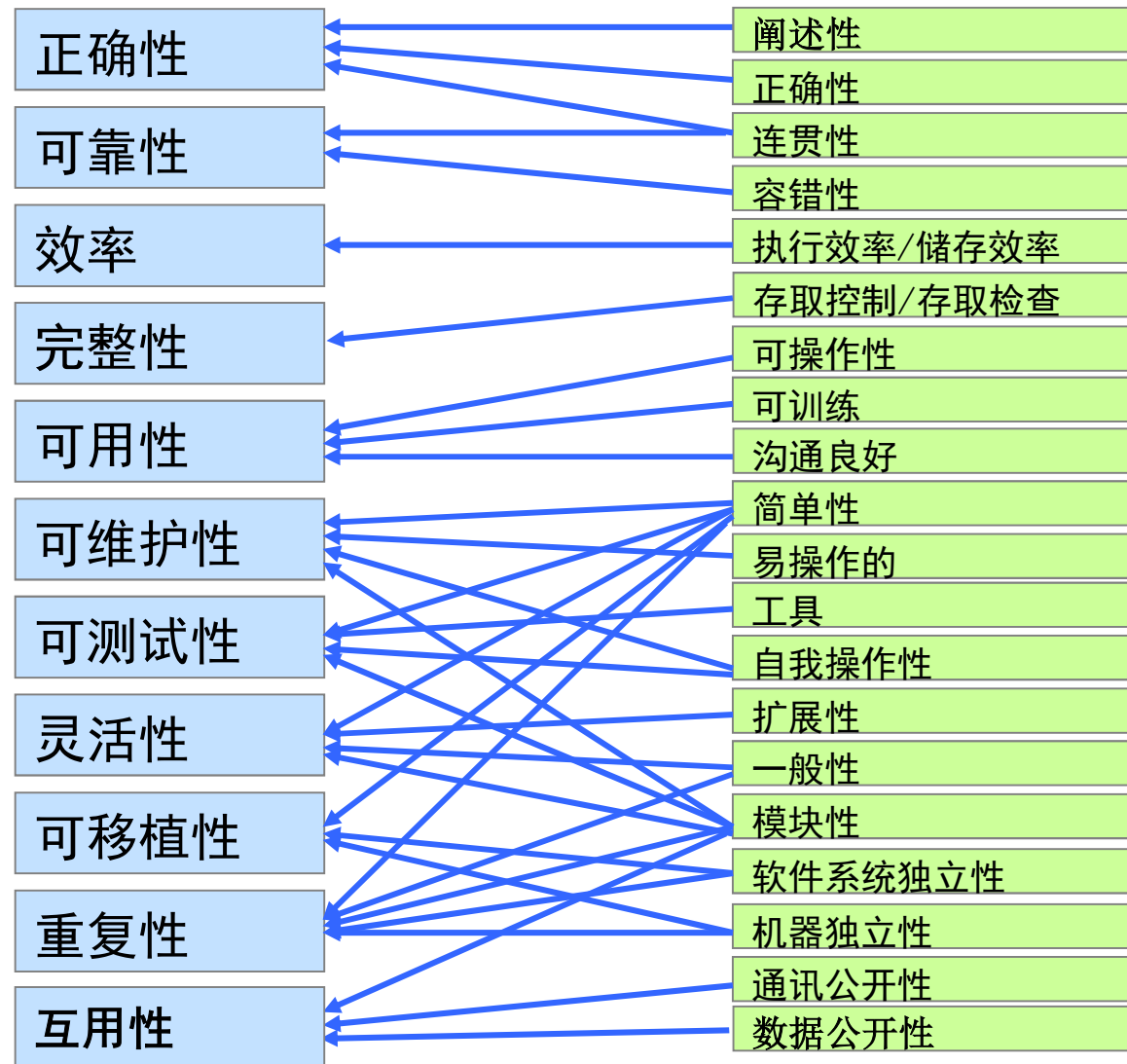
1.4 软件质量模型

1.4.3 软件质量分层模型

- *Barry W. Boehm* 在1976年首次提出了软件质量层次模型
 - 软件产品的质量基本上可从软件的可用性、软件的可维护性和软件的可移植性三个方面来考虑。
 - 软件质量在概念上分解为若干层次，对于最低层的软件质量概念引入量化指标，以便得到软件质量的整体评价。
 - 软件质量可以分为23个质量特性，它们是：
 - ✧ 可访问性、可说明性、准确性、可扩充性、通信性、完备性、简洁性、一致性、设备独立性、效率、人类工程、可读性、可维护性、可修改性、可移植性、可靠性、健壮性、自包含性、自描述性、结构性、可测试性、可理解性和可用性。

1.4 软件质量模型

Boehm 软件质量层次模型



1.4 软件质量模型

1.4.3 软件质量分层模型

- *McCall* 等将软件质量分解为 FCM 3层模型
 - FCM
 - ✧ 软件质量要素 (Factor)
 - ✧ 衡量标准 (Criteria)
 - ✧ 量度标准 (Metrics)
 - FCM 模型包括11个标准，分为三大类
 - ✧ 产品操作 (Product Operation)
 - ✧ 产品修正 (Product Revision)
 - ✧ 产品转移 (Product Transition)
 - 能够对 FCM 模型进行度量

1.4 软件质量模型

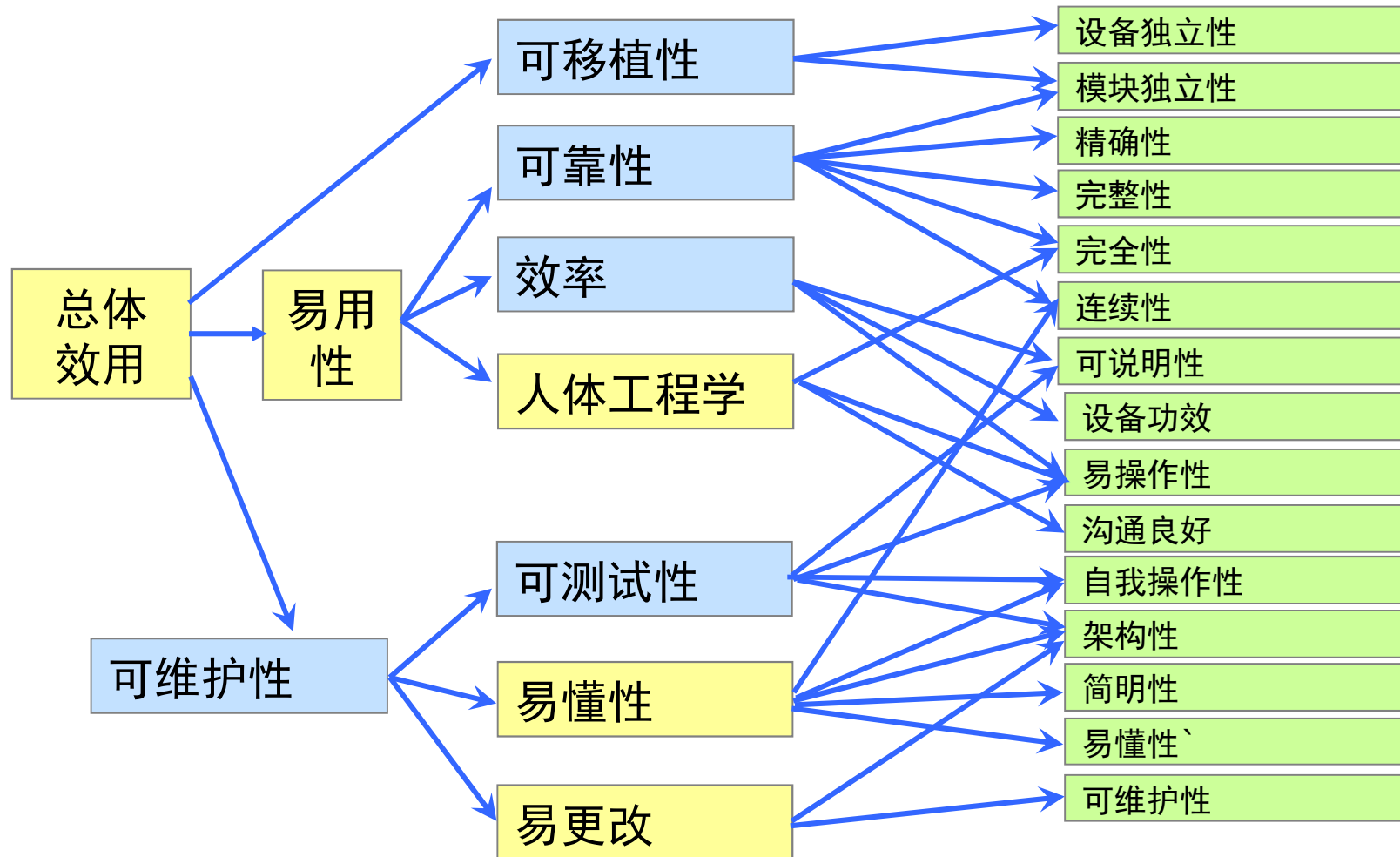
1.4.3 软件质量分层模型

- *McCall* 软件质量度量 FCM 模型

层级	名称	内容
第1层	质量要素： 描述和评价软件质量的一组属性	功能性、可靠性、易用性、效率性、可维护性、可移植性等质量特性以及将质量特性细化产生的副特性
第2层	衡量标准： 衡量标准的组合反映某一软件质量要素	精确性、稳健性、安全性、通信有效性、处理有效性、设备有效性、可操作性、培训性、完备性、一致性、可追踪性、可见性、硬件系统无关性、软件系统无关性、可扩充性、公用性、模块性、清晰性、自描述性、简单性、结构性、文件完备性等
第3层	量度标准： 可由各使用单位自定义	根据软件的需求分析、概要设计、详细设计、编码、测试、确认、维护与使用等阶段，针对每一个阶段制定问卷表，以此实现软件开发过程的质量度量

1.4 软件质量模型

McCall 软件质量层次模型



1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 将软件质量总结为6个质量特性，每个特性又包括一系列子特性。6个质量特性是：
 - ✧ 功能性：与一组功能及其指定性质有关的一组属性，这里的功能是满足明确或隐含的需求的那些功能。
 - ✧ 可靠性：在规定的一段时间和条件下，与软件维持其性能水平的能力有关的一组属性。
 - ✧ 可用性：由一组规定或潜在的用户为使用软件所需作的努力和所作的评价有关的一组属性。
 - ✧ 效率：与在规定条件下软件的性能水平与所使用资源量之间关系有关的一组属性。
 - ✧ 可维护性：与进行指定的修改所需的努力有关的一组属性。
 - ✧ 可移植性：与软件从一个环境转移到另一个环境的能力有关的一组属性。

1.4 软件质量模型

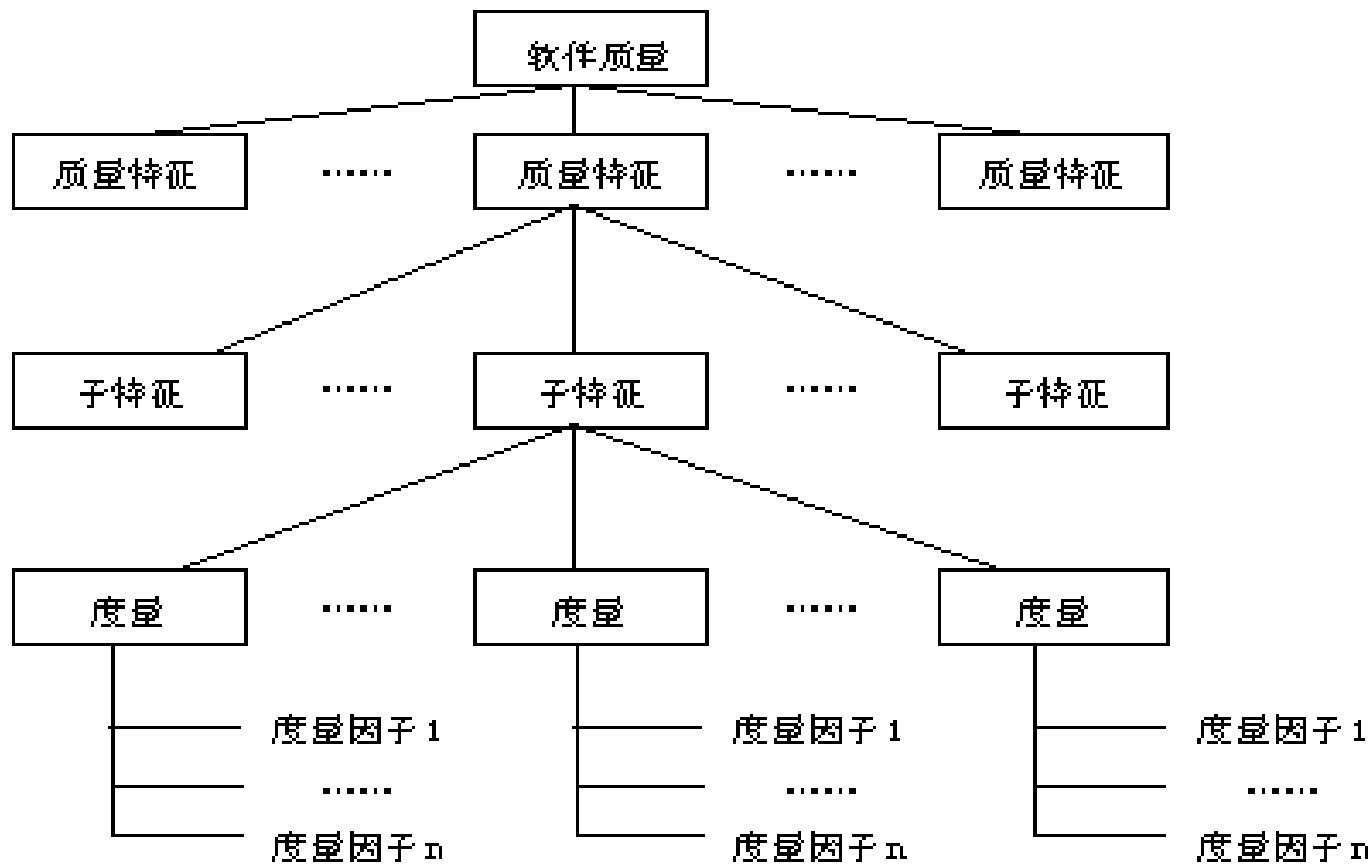
1.4.3 软件质量分层模型

- ISO/IEC 9126 软件质量模型包括3层：
 - 高层
 - ✧ 软件质量需求评价准则 (SQRC)
 - 中层
 - ✧ 软件质量设计评价准则 (SQDC)
 - 低层
 - ✧ 软件质量度量评价准则 (SQMC)

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 软件质量框架模型



1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型的6个质量特性和21个质量子特性

(1) 功能性 (Functionality)

◇ 功能性是指与软件所具有的各项功能及其规定性质有关的一组属性，包括：

- 适合性 (Suitability)：与规定任务能否提供一组功能以及这组功能的适合程度有关的软件属性。适合程度的例子是面向任务系统中由子功能构成功能是否合适、表容量是否合适等。
- 准确性 (Accuracy)：与能否得到正确或相符的结果或效果有关的软件属性。此属性包括计算值所需的准确程度。
- 互操作性 (互用性, Interoperability)：与同其他指定系统进行交互的能力有关的软件属性。为避免可能与易替换性的含义相混淆，此处用互操作性 (互用性) 而不用兼容性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (1) 功能性 (Functionality)

- ◇ 功能性是指与软件所具有的各项功能及其规定性质有关的一组属性，包括：(续)

- 依从性 (Compliance): 使软件遵循有关的标准、约定、法规及类似规定的软件属性。
 - 安全性 (Security): 与防止对程序及数据的非授权的故意或意外访问的能力有关的软件属性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

(2) 可靠性 (Reliability)

- ◇ 可靠性是指在规定运行条件下和规定时间周期内，与软件维护其性能级别的能力有关的一组属性。可靠性反映的是软件中存在的需求错误、设计错误和实现错误而造成的失效情况。包括
 - 成熟性 (Maturity): 与由软件故障引起失效的频度有关的软件属性
 - 容错性 (Fault tolerance): 与在软件故障或违反指定接口的情况下，维持规定的性能水平的能力有关的软件属性。指定的性能水平包括失效防护能力。
 - 可恢复性 (Recoverability): 与在失效发生后，重建其性能水平并恢复直接受影响数据的能力以及为达此目的所需的时间和努力有关的软件属性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (3) 可用性 (Usability)

- ◇ 可用性是指根据规定用户或隐含用户的评估所作出的关于与使用软件所需要的努力程度有关的一组属性。包括：

- 可理解性 (Understandability)：与用户为认识逻辑概念及其应用范围所花的努力有关的软件属性。
 - 易学性 (Learnability)：与用户为学习软件应用 (例如运行控制、输入、输出) 所花的努力有关的软件属性。
 - 可操作性 (Operability)：与用户为操作和运行控制所花努力有关的软件属性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (4) 效率 (Efficiency)

- ◇ 效率是指在规定条件下，与软件性能级别和所使用资源总量之间的关系有关的一组属性。包括：

- 时间特性 (Time behavior)：与软件执行其功能时响应和处理时间以及吞吐量有关的软件属性。
 - 资源特性 (Resource behavior)：与在软件执行其功能时所使用的资源数量及其使用时间有关的软件属性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (5) 可维护性 (Maintainability)

- ◇ 可维护性是指与对软件进行修改的难易程度有关的一组属性。
包括：

- 可分析性 (Analysability)：与为诊断缺陷或失效原因及为判定待修改的部分所需努力有关的软件属性。
 - 可改变性 (Changeability)：与进行修改、排除错误或适应环境变化所需努力有关的软件属性。
 - 稳定性 (Stability)：与修改所造成的未预料结果的风险有关的软件属性。
 - 可测试性 (Testability)：与确认已修改软件所需的努力有关的软件属性。此子特性的涵义可能会被研究中的修改加以改变。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (6) 可移植性 (Portability)

- ◇ 可移植性是指与一个软件从一个环境转移到另一个环境运行的能力有关的一组属性。包括：

- 适应性 (Adaptability)：与软件无需采用有别于为该软件准备的活动或手段就可能适应不同的规定环境有关的软件属性。
 - 可安装性 (Installability)：与在指定环境下安装软件所需努力有关的软件属性。
 - 遵循性 (一致性, Conformance)：使软件遵循与可移植性有关的标准或约定的软件属性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- ISO/IEC 9126 质量模型 (续)

- (6) 可移植性 (Portability)

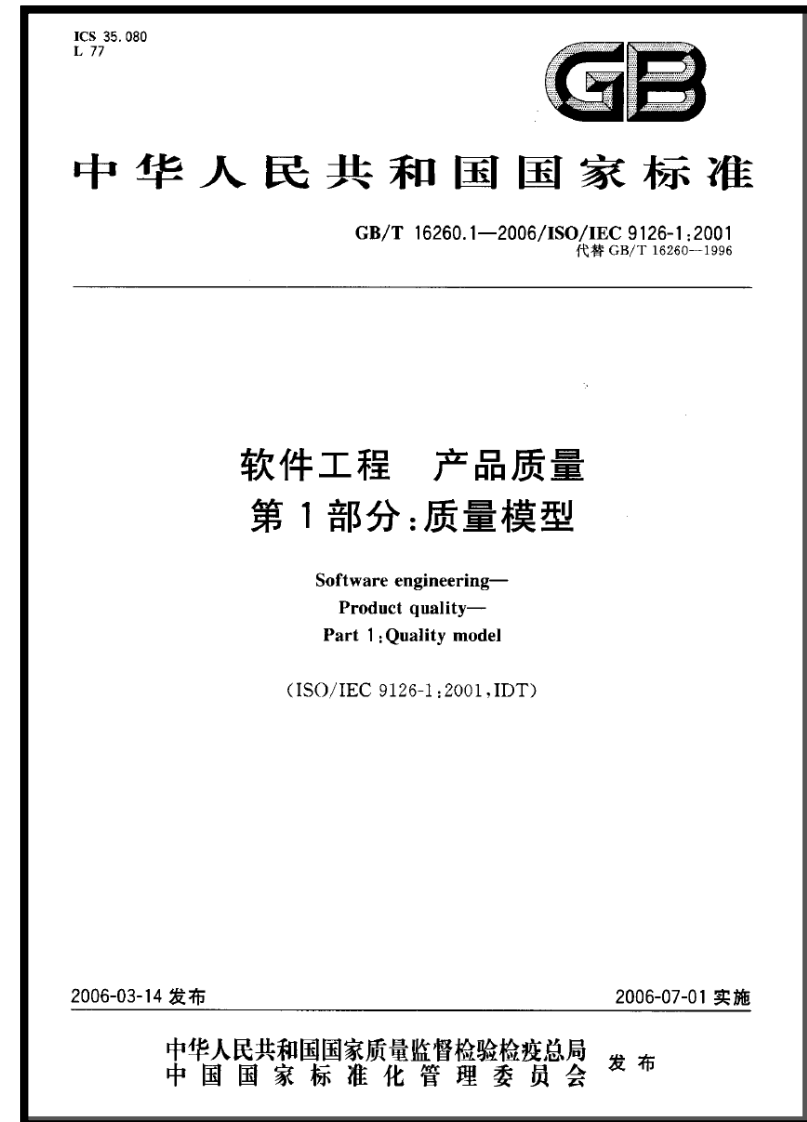
- ◇ 可移植性是指与一个软件从一个环境转移到另一个环境运行的能力有关的一组属性。包括：(续)

- 可替换性 (Replaceability)：与软件在该软件环境中用来替代指定的其他软件的机会和努力有关的软件属性。为避免可能与互操作性 (互用性) 的含义相混淆，此处用易替换性而不用兼容性。特定软件的易替换性并不隐含此软件可由所考虑的软件所替代。易替换性可能包含易安装性和适应性这两个属性。由于此概念的重要性，它已被采用作为一个独立的子特性。

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型



1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型
 - 软件质量特性
 - ◇ 功能性
 - 软件所实现的功能满足用户需求的程度
 - 功能性反映了所开发的软件满足用户表述或隐含需求的程度，即用户要求的功能是否全部实现
 - ◇ 可靠性
 - 在规定时间和条件下，软件所能维持其性能水平的程度
 - 可靠性对某些软件是重要的质量要求，它除了反映软件满足用户需求正常运行的程度，且反映了在故障发生时能继续运行的程度

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 软件质量特性 (续)
 - ◇ 易用性
 - 对于一个软件，用户学习、操作、准备输入和理解输出时，所做努力的程度
 - 易使用性反映了与用户的友善性，即用户在使用本软件时是否方便
 - ◇ 效率
 - 在指定的条件下，用软件实现某种功能所需的计算机资源 (包括时间) 的有效程度
 - 效率反映了在完成功能要求时，有没有浪费资源。资源这个术语有比较广泛的含义，它包括了内存、外存的使用，通道能力及处理时间等

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 软件质量特性 (续)
 - ◇ 维护性
 - 在一个可运行软件中, 为了满足用户需求、环境改变或软件错误发生时, 进行相应修改所做的努力程度
 - 可维修性反映了在用户需求改变或软件环境发生变更时, 对软件系统进行相应修改的容易程度
 - 一个易于维护的软件系统也是一个易理解、易测试和易修改的软件能够纠正或增加新的功能, 或允许在不同软件环境中进行操作
 - ◇ 可移植性
 - 从一个计算机系统/环境转移到另一个计算机系统/环境的容易程度

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)

- 质量子特性

- (每一个软件质量特性都分别与若干质量子特性相对应)

- ✧ 适用性/适合性 (Suitability)

- 与能否提供一组功能以及这组功能是否适合相关的软件属性

- ✧ 准确性 (Accurateness)

- 与能否得到正确或相符的结果或效果有关的软件属性

- ✧ 互操作性 (Interoperability)

- 与其它指定系统交互使用的能力有关的软件属性

- ✧ 符合性/依从性 (Compliance)

- 与遵循和应用有关的标准、约定或类似规定有关的软件属性

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 质量子特性 (续)
 - ✧ 可靠安全性 (Safety)
 - 与避免可能引起死亡、伤害或职业病以及避免对设备或财产造成损坏或损失能力有关的软件属性
 - ✧ 保密安全性 (Security)
 - 与防止对程序和数据进行偶然或故意的非法存取的能力有关的软件属性
 - ✧ 成熟性 (Maturity)
 - 与软件故障引起的失效频度有关的软件属性
 - ✧ 容错性 (Fault Tolerance)
 - 与在产生软件故障或违反规定接口的情况下，保持软件与规定的性能级别的能力有关的软件属性



1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 质量子特性 (续)
 - ✧ 可恢复性 (Recoverability)
 - 与在软件运行失效且需使其重新工作时，重建其性能级别和恢复直接受影响数据的能力有关的软件属性
 - ✧ 可理解性 (Understandability)
 - 与用户为识别逻辑概念及其应用所作的努力有关的软件属性
 - ✧ 易学性 (Learnability)
 - 与用户学习软件的应用所作的努力有关的软件属性
 - ✧ 易操作性 (Operability)
 - 与用户为掌握操作和操作控制所作努力有关的软件属性

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 质量子特性 (续)
 - ✧ 时间行为 (Time Behavior)
 - 与响应和处理时间以及软件执行其功能的吞吐率有关的软件属性
 - ✧ 资源行为 (Resource Behavior)
 - 与软件执行其功能时，所用资源的数量以及这些资源所占时间有关的软件属性
 - ✧ 易分析性 (Analyzability)
 - 与为诊断缺陷或失效原因、或为标识需要修改部分所需努力有关的软件属性
 - ✧ 易改变性 (Changeability)
 - 与进行修改、排错或适应环境的难度有关的软件属性

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 质量子特性 (续)
 - ◇ 稳定性 (Stability)
 - 与修改造成未预料效果所导致的风险有关的软件属性
 - ◇ 可测试性 (Testability)
 - 与确认被修改软件所需努力有关的软件属性
 - ◇ 适应性 (Adaptability)
 - 与软件无需采用有别于为该软件准备的处理或手段就可以适应于不同的规定环境有关的软件属性
 - ◇ 易安装性 (Installability)
 - 与将软件安装到指定环境所需努力有关软件属性

1.4 软件质量模型

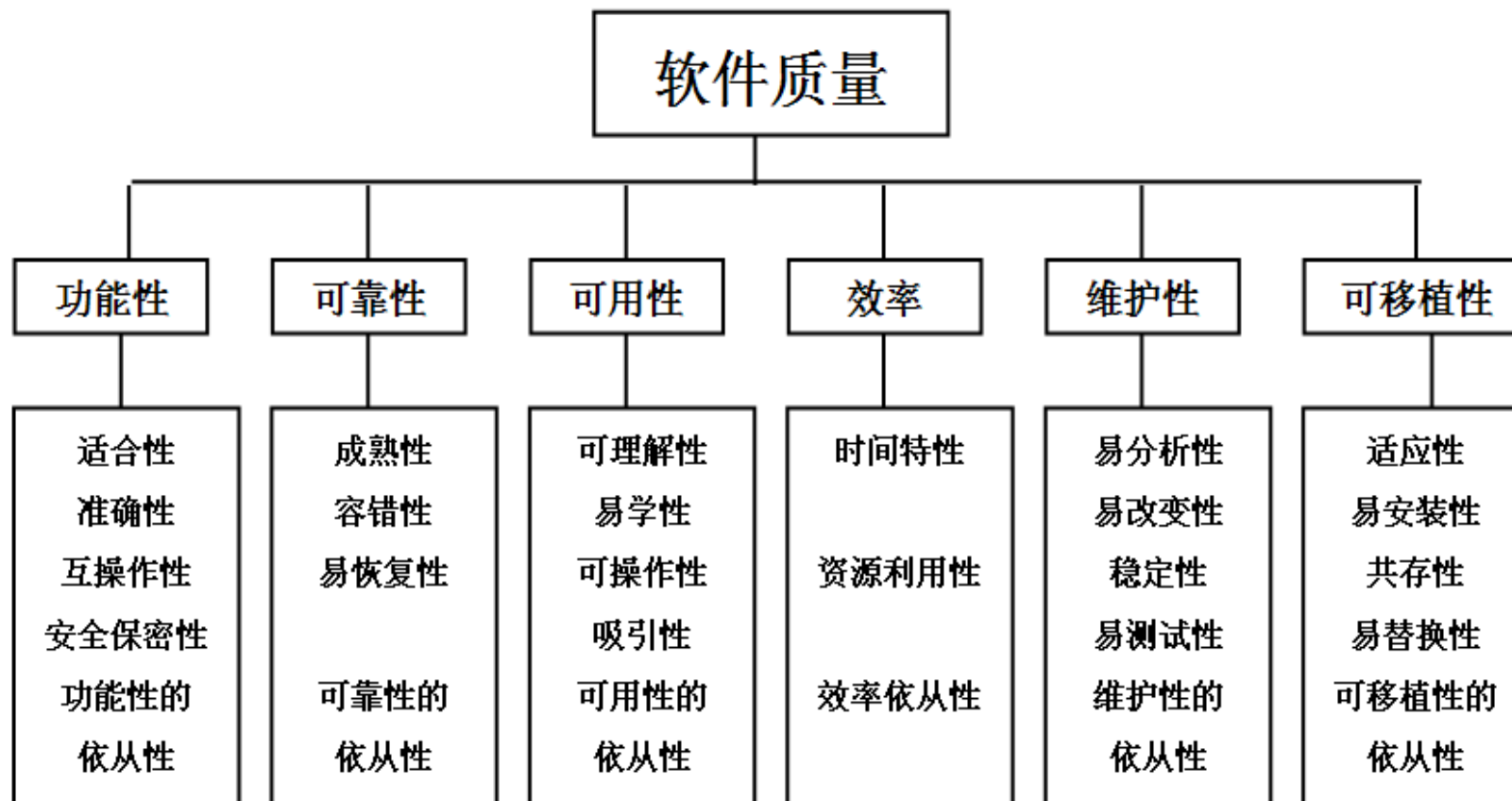
1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型 (续)
 - 质量子特性 (续)
 - ✧ 符合性 (Conformance)
 - 与使软件遵循移植有关标准或约定有关的软件属性
 - ✧ 可替换性 (Replaceability)
 - 与在规定应用其他软件の場合而应用此软件的可能性以及所需努力有关的软件属性

1.4 软件质量模型

1.4.3 软件质量分层模型

- GB/T 16260-2006 软件质量模型



Thank you!

