

---

Chapter 1

# Overview of Software Engineering

---

School of Data & Computer Science  
Sun Yat-sen University

*Approaches & Technologies*





## OUTLINE



- 1.1 软件与软件危机
- 1.2 软件开发与软件工程
- 1.3 软件生命周期模型
- **1.4 软件质量标准**
- 1.5 敏捷开发
- 1.6 软件生命周期过程





### ■ 软件质量问题

- 软件质量问题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。





### ■ 软件质量的概念

#### ■ 现代质量管理

- 质量是“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。





### ■ 软件质量的概念

#### ■ 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.





### ■ 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。





### ■ 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% 的用户总是有效的。
  - 例如：性能测试和恢复测试

#### ■ Accessibility (易用性/可用性)

- 产品 (或服务) 容易被用户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





### ■ 高质量软件标准体系

#### ■ 产品质量

- 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活动进行改进。
- 软件产品质量模型： *McCall* 模型， *Boehm* 模型， ISO/IEC 9126 模型
- GB/T 16260.1-2006 ISO/IEC 9126-1:2001 质量模型
  - GB/T 16260.2-2006： 外部度量
  - GB/T 16260.3-2006： 内部度量
  - GB/T 16260.4-2006： 使用质量的度量







### ■ 高质量软件标准体系

#### ■ 过程质量控制

- 软件能力成熟度模型 CMM (Capability Maturity Model, SEI)
- 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, ISO/IEC/JTC1)
  - ISO/IEC 12207-2004
  - ISO/IEC 15504-2003-2013
  - ISO/IEC 33001-2015





### ■ 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付出的努力的程度 (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。





### ■ 软件产品质量标准

#### ■ 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

- 对于一个运行阶段的软件，当环境改变或软件发生错误时，进行相应修改所付出的努力的程度。

#### ■ 兼容性 Compatibility

#### ■ 可扩展性 Extensibility





### ■ 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





### ■ 软件质量的三种不同视面

#### ■ 管理者角度

- 关注质量的总体，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。

- 软件质量特性是从用户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。





### ■ 软件质量特性

#### ■ 软件质量特性的定义

##### (1) 功能性 Functionality

- 软件实现的功能达到其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破坏，软件应具备的自身保护能力。





## ■ 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渡所需要付出的努力的程度。

#### (9) 可重用性 Reusability

- 软件整体或其中一部分可以被再次利用的程度。





### ■ 软件质量层次模型

■ *Barry W. Boehm* 在1976年首次提出了软件质量层次模型并逐渐完善。

■ 软件产品的质量包括高层属性、中层属性和原始属性。

■ 高层属性关注3个问题

- As-is Utility 软件的可用性； Maintainability 软件的可维护性； Portability 软件的可移植性。

■ 中层属性包含了 7 个质量要素：

- Portability (General utility characteristics)
- Reliability (As-is utility characteristics)
- Efficiency (As-is utility characteristics)
- Usability (As-is utility characteristics, Human Engineering)
- Testability (Maintainability characteristics)
- Understandability (Maintainability characteristics)
- Flexibility (Maintainability characteristics, Modifiability)

■ 原始属性处于最低层，需要引入量化指标，以便得到软件质量的整体评价。







### ■ 软件质量层次模型

#### ■ Boehm 软件质量层次模型

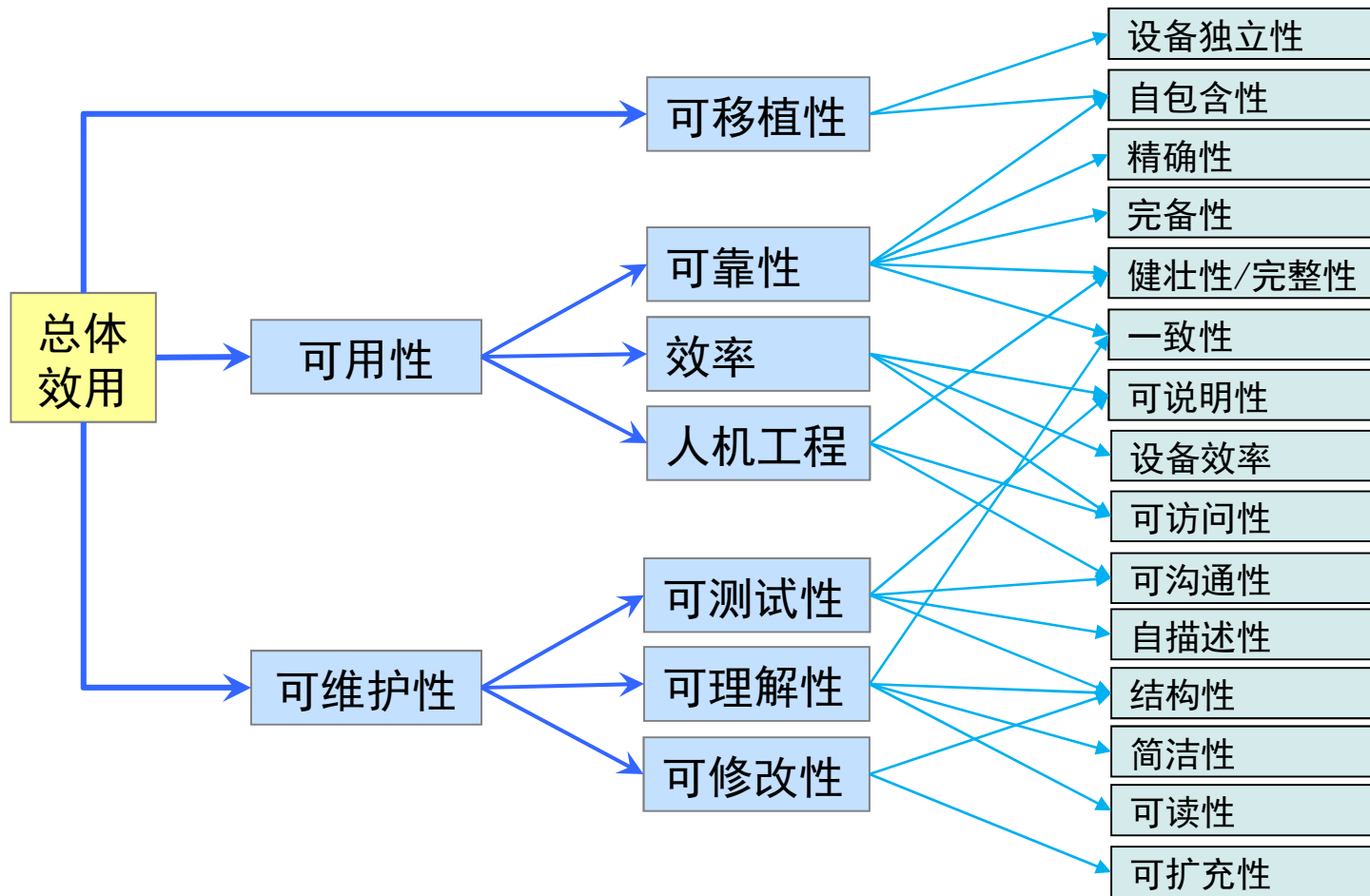
##### ■ 软件质量可以分为23个质量特性，它们是：

- 可访问性、可说明性、精确性、可扩充性、可沟通性、完备性、简洁性、一致性、设备独立性、设备效率、人机工程、可读性、可维护性、可修改性、可移植性、可靠性、健壮性 (完整性)、自包含性、自描述性、结构性、可测试性、可理解性、可用性。



## ■ 软件质量层次模型

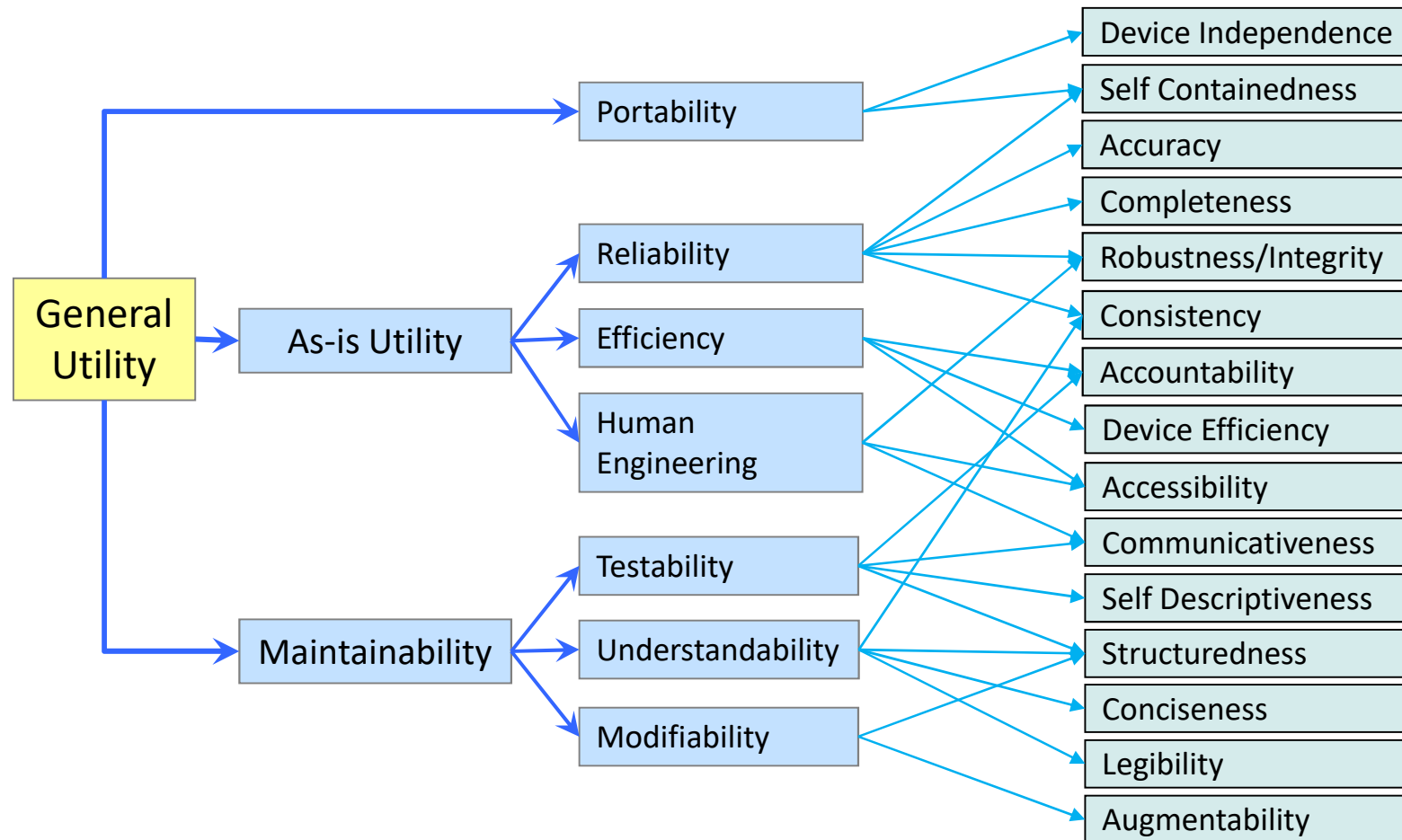
### ■ Boehm 软件质量层次模型 (续)





## ■ 软件质量层次模型

### ■ Boehm 软件质量层次模型 (续)





### ■ 软件质量层次模型

#### ■ McCall 软件质量模型

- McCall 软件质量模型也称为 GE 模型 (General Electrics Model, *Jim McCall*, 1977), 它将软件质量分解为 FCM 3层模型并进行度量。

#### ■ FCM 3层模型结构

- 质量要素 (Factor)
  - 11 Factors (To specify): 描述软件的外部视角, 即用户或使用者的视角。
- 衡量标准 (Criteria)
  - 23 Criteria (To build): 描述软件的内部视角, 即开发人员的视角。
- 量度标准 (Metrics)
  - Metrics (To control): 定义衡量指标和方法。





## ■ 软件质量层次模型

### ■ McCall 软件质量模型 (续)

层级	概 念	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、效率特性、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	衡量标准的一定组合可以反映某一项质量要素	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	可由各使用单位自行定义	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。





### ■ 软件质量层次模型

#### ■ McCall 软件质量模型 (续)

##### ■ FCM 模型的11个质量要素也可以分为三大类。

- 产品操作 (Product Operation, basic operational characteristics) 包括:
  - Correctness, Reliability, Efficiency, Integrity, Usability.
- 产品修正 (Product Revision, ability to change) 包括:
  - Maintainability, Testability, Flexibility.
- 产品转移 (Product Transition, adaptability to new environments) 包括:
  - Portability, Reusability, Interoperability.

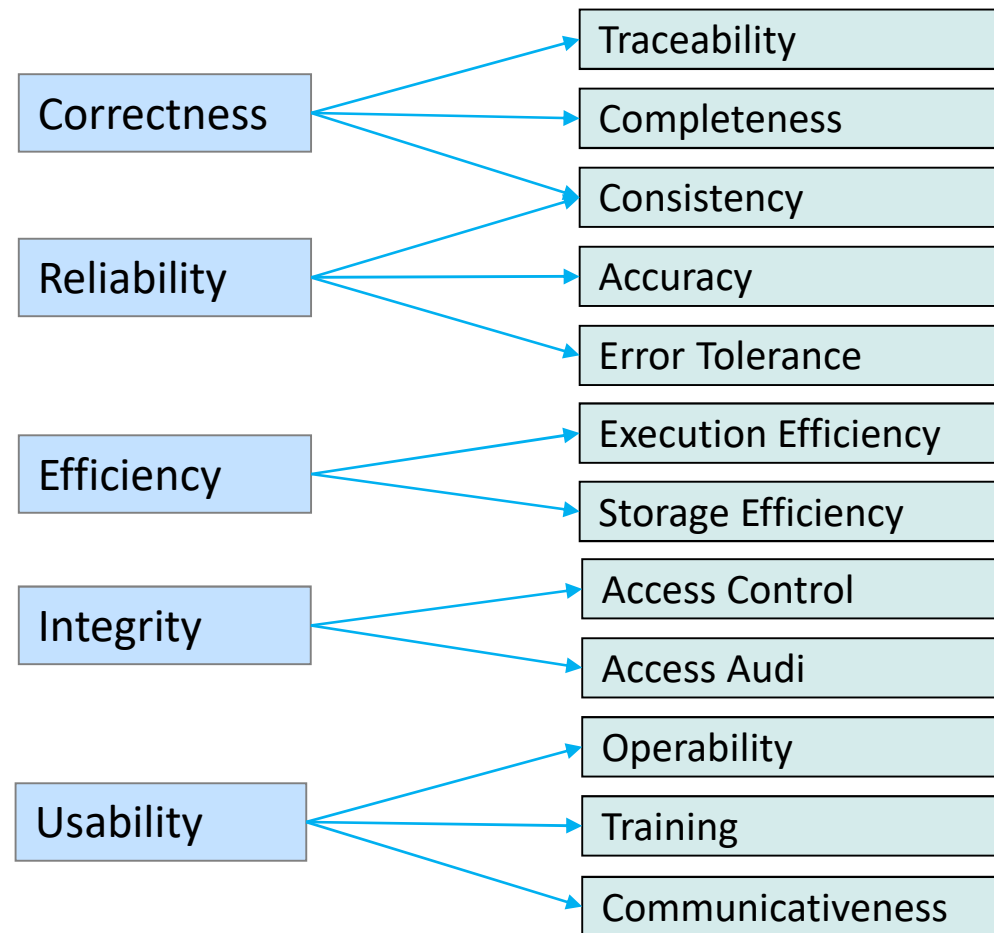




## ■ 软件质量层次模型

### ■ McCall 软件质量模型 (续)

■ McCall 模型的11个质量要素和23个衡量标准的关联关系：

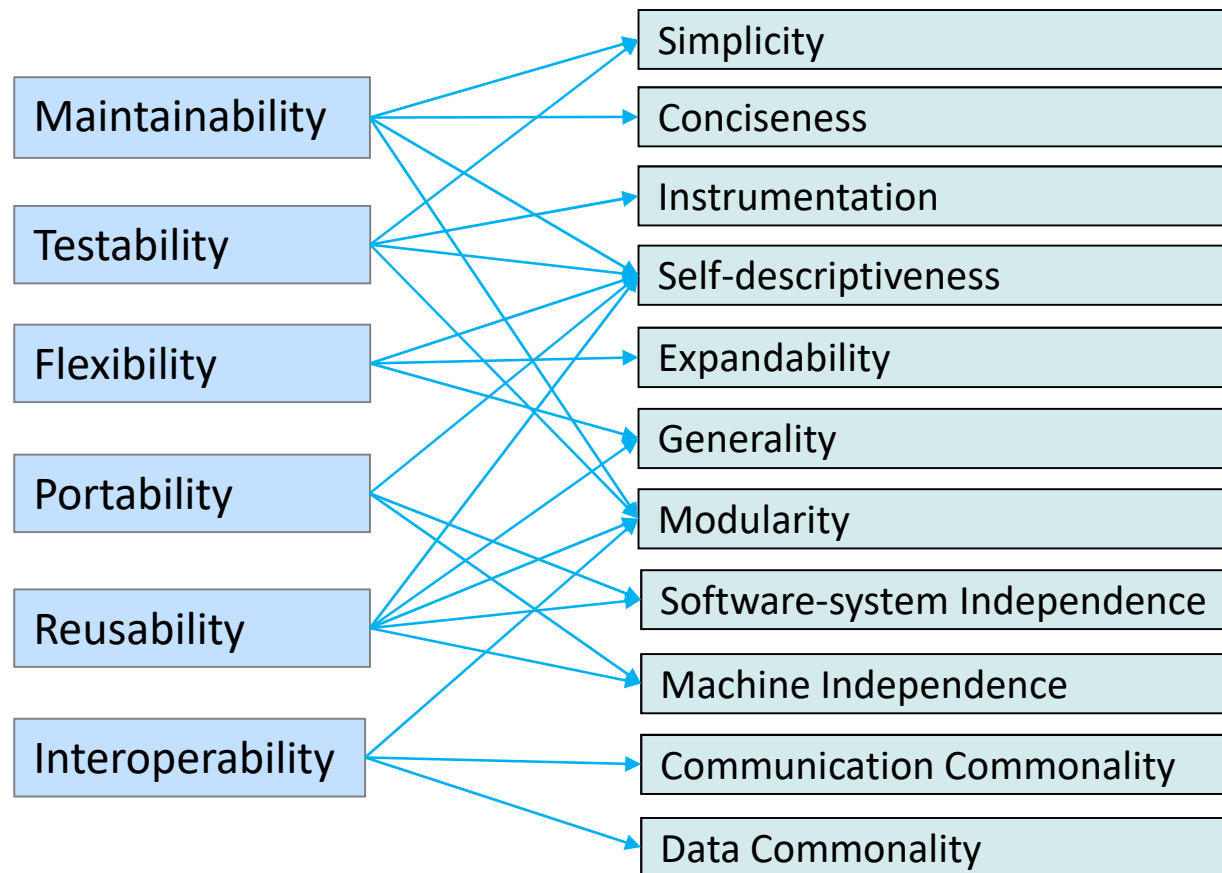




## ■ 软件质量层次模型

### ■ McCall 软件质量模型 (续)

#### ■ McCall 模型的11个质量要素和23个衡量标准的关联关系：(续)







### ■ 软件质量层次模型

#### ■ ISO/IEC 9126-1:2001 软件质量模型

- ISO/IEC 9126:2001 is revised by **ISO/IEC 25010:2011** : Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.
- ISO/IEC 25010 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, and is a part of the SQuaRE series of International Standards, which consists of the following divisions:
  - Quality Management Division (ISO/IEC 2500n),
  - Quality Model Division (ISO/IEC 2501n),
  - Quality Measurement Division (ISO/IEC 2502n),
  - Quality Requirements Division (ISO/IEC 2503n),
  - Quality Evaluation Division (ISO/IEC 2504n),
  - SQuaRE Extension Division (ISO/IEC 25050 – ISO/IEC 25099).





### ■ 软件质量层次模型

#### ■ ISO/IEC 9126-1:2001 软件质量模型 (续)

##### ■ ISO/IEC 9126-1:2001 软件质量模型的3层框架结构:

- 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)

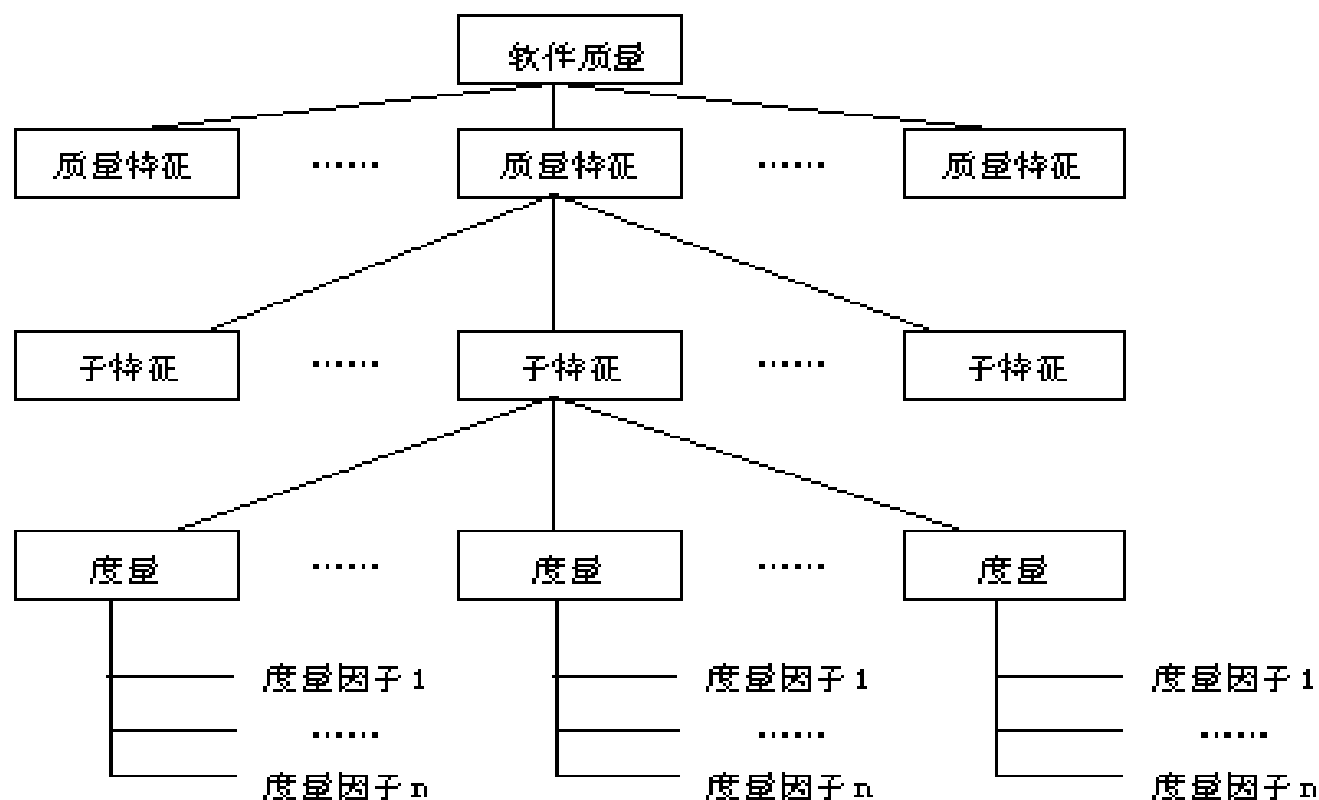




## ■ 软件质量层次模型

### ■ ISO/IEC 9126-1:2001 软件质量模型 (续)

#### ■ ISO/IEC 9126-1:2001 软件质量模型结构





### ■ ISO/IEC 9126-1:2001 软件质量模型

- ISO/IEC 9126-1:2001 将软件质量总结为6个质量特性和21个质量子特性，每个质量特性包括一系列的质量子特性。

- 6个质量特性是：

- 功能性、可靠性、可用性、效率特性、可维护性、可移植性

- 21个质量子特性是：

- 适合性、准确性、互操作性(互用性)、依从性、安全性、成熟性、容错性、可恢复性、可理解性、易学性、可操作性、时间特性、资源特性、可分析性、可改变性、稳定性、可测试性、适应性、可安装性、遵循性(一致性/符合性)、可替换性(易替换性)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义

##### ■ 功能性 (Functionality)

- 与软件所具有的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。
- 功能性包括：
  - 适合性 (Suitability)
  - 准确性 (Accuracy)
  - 互操作性 (互用性, Interoperability)
  - 依从性 (Compliance)
  - 安全性 (Security)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义 (续)

##### ■ 可靠性 (Reliability)

- 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。
- 可靠性包括：
  - 成熟性 (Maturity)
  - 容错性 (Fault Tolerance)
  - 可恢复性 (Recoverability)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义 (续)

##### ■ 可用性 (Usability)

- 根据规定或潜在的用户的评估所得出的关于与使用软件所需要的努力程度有关的一组属性。
- 可用性包括：
  - 可理解性 (Understandability)
  - 易学性 (Learnability)
  - 可操作性 (Operability)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义 (续)

##### ■ 效率特性 (Efficiency)

- 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。
- 效率特性包括：
  - 时间特性 (Time Behavior)
  - 资源特性 (Resource Behavior)







### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义 (续)

##### ■ 可维护性 (Maintainability)

- 与对软件进行修改的难易程度有关的一组属性。
- 可维护性包括：
  - 可分析性 (Analysability)
  - 可改变性 (Changeability)
  - 稳定性 (Stability)
  - 可测试性 (Testability)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的6个质量特性的定义 (续)

##### ■ 可移植性 (Portability)

- 与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。
- 可移植性包括：
  - 适应性 (Adaptability)
  - 可安装性 (Installability)
  - 遵循性 (一致性/符合性, Conformance)
  - 可替换性 (易替换性, Replaceability)





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义

##### ■ 与功能性相关：

- 适合性：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：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)：与同其他指定系统进行交互的能力有关的软件属性。为避免可能与可替换性的含义相混淆，此处用互操作性而不用兼容性。
- 依从性：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：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。





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义 (续)

##### ■ 与可靠性相关：

- 成熟性：与由软件故障引起失效的频度有关的软件属性。
- 容错性：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：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。





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义 (续)

##### ■ 与可用性相关:

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





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义 (续)

##### ■ 与效率特性相关

- 时间特性：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：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。





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义 (续)

##### ■ 与可维护性特性相关：

- 可分析性：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：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：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：与确认已修改软件所需的努力有关的软件属性。此子特性的含义可能会随着相关研究的进展而修订。





### ■ ISO/IEC 9126-1:2001 软件质量模型

#### ■ ISO/IEC 9126-1 质量模型的21个质量子特性的定义 (续)

##### ■ 与可移植性特性相关：

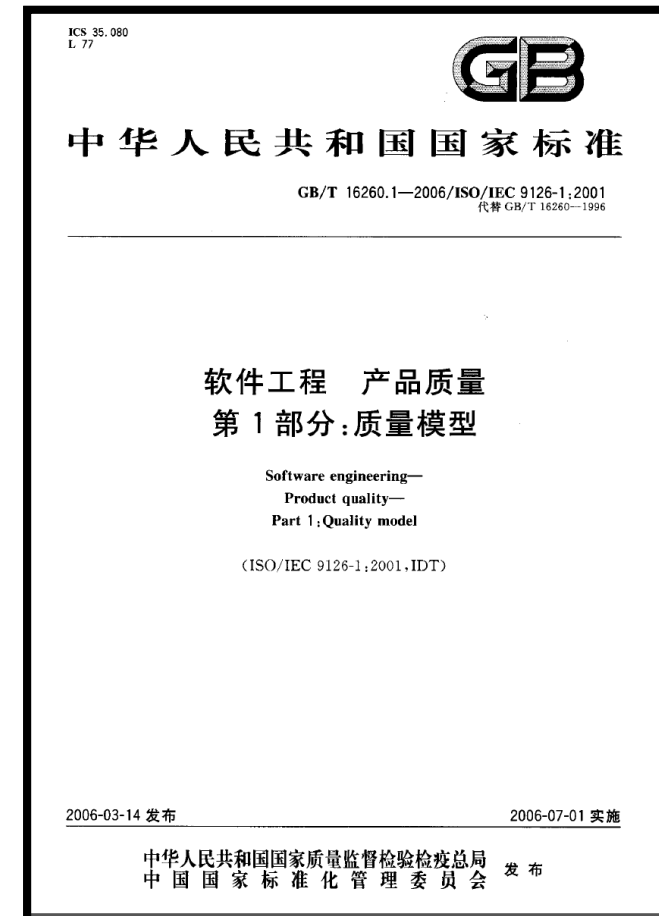
- 适应性：与软件无需采用有别于为该软件准备的活动或手段就可能适应不同的规定环境有关的软件属性。
- 可安装性：与在指定环境下安装软件所需努力有关的软件属性。
- 遵循性：使软件遵循与可移植性有关的标准或约定的软件属性。
- 可替换性：与软件在该软件环境中用来替代指定的其他软件的机会和努力有关的属性。为避免与互操作性的含义相混淆，此处用可替换性而不用兼容性。特定软件的可替换性并不隐含此软件可由所考虑的软件所替代。可替换性可能包含易安装性和适应性这两个属性。由于此概念的重要性，它已被采用作为一个独立的子特性。







### ■ GB/T 16260.1-2006 软件质量模型





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 软件质量特性

##### ■ 功能性

- 在指定条件下使用时，软件产品提供满足明确和隐含要求的功能的能力。
- 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

- 在指定条件下使用时，软件产品维持规定的性能级别的能力。
- 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，还反映了在故障发生时能继续运行的程度。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 软件质量特性 (续)

##### ■ 易用性

- 在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力。
- 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

- 在规定条件下，相对于所用资源的数量，软件产品可提供适当性能的能力。
- 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。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 软件质量特性 (续)

##### ■ 维护性

- 软件产品可被修改的能力。修改可能包括纠正、改进或软件对环境、需求和功能规格说明变化的适应。
- 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，它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

- 软件产品从一种环境迁移到另外一种环境的能力。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性

##### ■ 适合性 (Suitability)

- 软件产品为指定的任务和用户目标提供一组合适的功能的能力。

##### ■ 准确性 (Accurateness)

- 软件产品提供具有所需精度的正确或相符的结果或效果的能力。

##### ■ 互操作性 (Interoperability)

- 软件产品与一个或更多的规定系统进行交互的能力。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 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)

- 软件产品保护信息和数据的能力，以使未授权的人员或系统不能阅读或修改这些信息和数据，而不拒绝授权的人员或系统对它们的访问。

##### ■ 依从性 (Compliance)

- 软件产品遵循相关的标准、约定或法规以及类似规定的能力。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性 (续)

##### ■ 成熟性 (Maturity)

- 软件产品为避免由软件中故障而导致失效的能力。
- 成熟性体现了软件产品对失效频度的控制能力。

##### ■ 容错性 (Fault Tolerance)

- 在软件出现故障或违反其指定接口的情况下，软件产品维持规定的性能级别的能力。

##### ■ 易恢复性 (Recoverability)

- 在失效发生的情况下，软件产品重建规定的性能级别并恢复受直接影响的数据的能力。
- 成熟性、容错性和易恢复性合并形成有效性 (Availability)。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性 (续)

##### ■ 易理解性 (Understandability)

- 软件产品使用户能够理解软件是否合适以及如何能将软件用于特定的任务和使用条件的能力。
- 易理解性与用户为识别逻辑概念及其应用所作的努力有关。

##### ■ 易学性 (Learnability)

- 软件产品使用户能学习其应用的能力。

##### ■ 易操作性 (Operability)

- 软件产品使用户能操作和控制它的能力。

##### ■ 吸引性 (Attractability)

- 软件产品吸引用户的能力。







### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性 (续)

##### ■ 时间特性 (Time Behavior)

- 在规定条件下，软件产品执行其功能时，提供适当的响应和处理时间以及吞吐率的能力。

##### ■ 资源利用性 (Resource Behavior)

- 在规定条件下，软件产品执行其功能时，使用合适数量和类别的资源的能力。

##### ■ 易分析性 (Analyzability)

- 软件产品诊断软件中的缺陷或失效原因或识别待修改部分的能力。

##### ■ 易改变性 (Changeability)

- 软件产品使指定的修改可以被实现的能力。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性 (续)

##### ■ 稳定性 (Stability)

- 软件产品避免由于软件修改而造成意外结果的能力。

##### ■ 易测试性 (Testability)

- 软件产品使已修改软件能被确认的能力。

##### ■ 适应性 (Adaptability)

- 软件产品毋须采用额外的活动或手段就可以适应不同指定环境的能力。

##### ■ 易安装性 (Installability)

- 软件产品在指定环境中被安装的能力。





### ■ GB/T 16260.1-2006 软件质量模型

#### ■ 质量子特性 (续)

##### ■ 共存性 (Compossibility)

- 软件产品在公共环境中同与其分享公共资源的其他独立软件共存的能力。

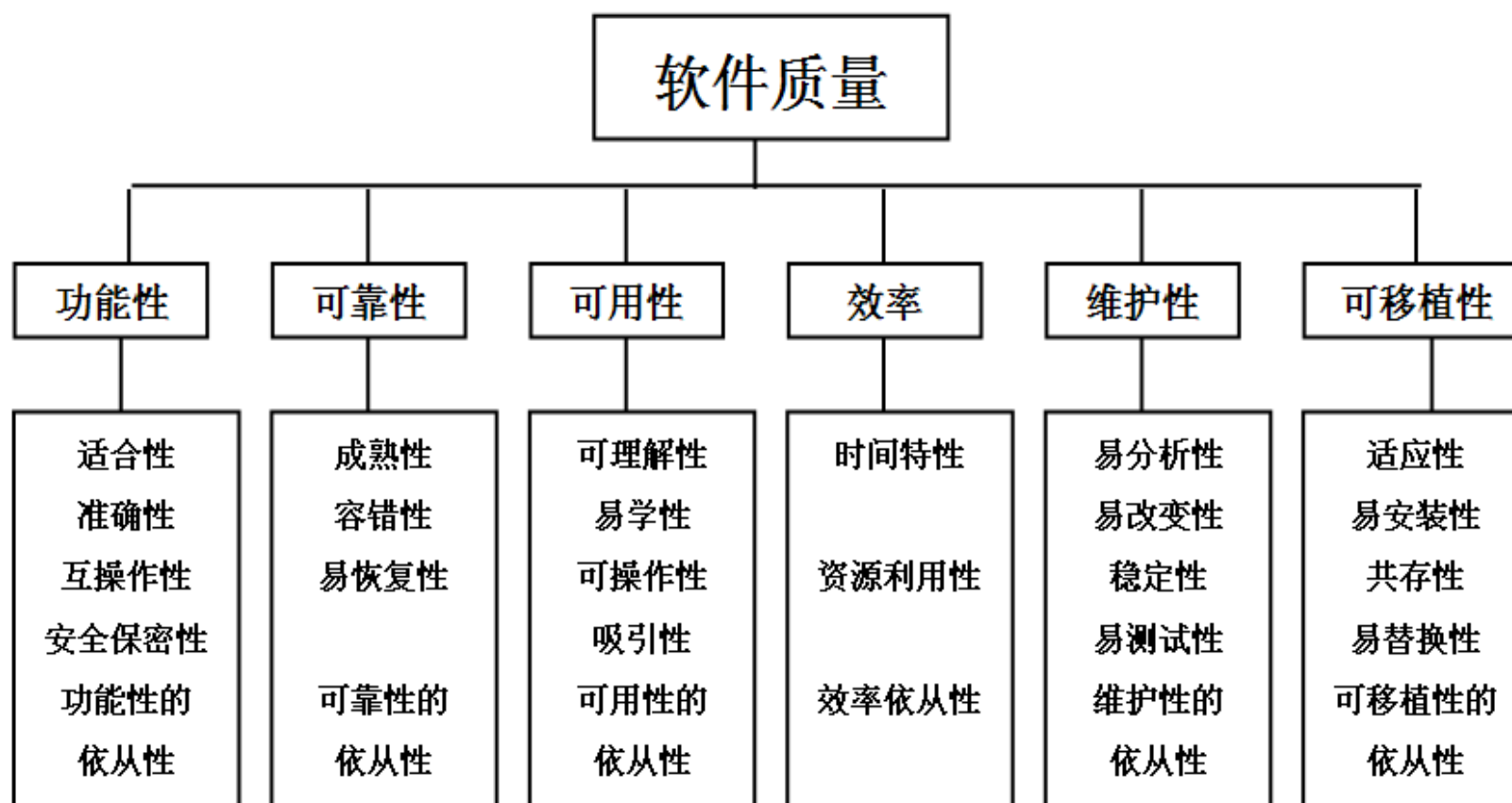
##### ■ 易替换性 (Replaceability)

- 软件产品在同样环境下，替代另一个相同用途的指定软件产品的能力。





## ■ GB/T 16260.1-2006 软件质量模型



# Thank you!

