

软件质量

Software Quality

目录

CONTENTS

- 软件质量的5个观点
- McCall的质量因子和标准
- ISO 9126质量特性
- ISO 9000:2000基本法则

目录

CONTENTS

- 软件质量的5个观点
- McCall的质量因子和标准
- ISO 9126质量特性
- ISO 9000:2000基本法则

软件质量的5个观点

为了能更好地理解质量概念，从更广泛的角度来学习质量。Garvin（加温）分析了如何在不同领域用不同方式来理解质量，其包括5个观点。

软件质量的5个观点：

抽象的观点

用户的观点

制造的观点

产品的观点

基于价值的观点



软件质量的5个观点

● **抽象的观点**：在抽象的观点中，**质量可以通过经验来认识**，但不能用某种简单的方式来定义。质量是一种理想的状态，由于过于复杂而使其不能精确地定义。然而，拥有良好质量的对象是引人注目的，能够比较容易地识别出来。由于抽象视角的哲学性质，质量的概念不能用具体的度量标准来描述。

● **用户的观点**：用户的观点即**关于产品符合用户需求和期望的程度**。质量并不仅仅就产品能交付这一方面来说，还受到销售合同中服务提供内容的影响。在这一观点中，用户关心的是这一产品是否适合使用。这一观点实际上是非常个性化的。

软件质量的5个观点

● **制造的观点**：制造的观点起源于制造领域，在这一观点中，**质量被认为是对于需求的符合程度**。任何对于标准需求的偏离都被认为是产品质量的降低。

● **产品的观点**：产品的观点中的主要假设是：**假如制造一个产品具有好的内部特性，那么它会有好的外部质量**。产品的观点是非常引人注目的，因为它为探索产品的内部特性和外部质量之间的关系提供了可能性。在这一观点中，当前产品的质量水平表明了可衡量产品属性的存在与否。质量的产品观点可以用客观的方式来评定。一个有关软件质量的产品观点就是高度模块化，这是一种内部特性，它使得软件可测试和可维护。



软件质量的5个观点

● **基于价值的观点**：基于价值的观点是两种独立观点的结合：优秀和产品价格。质量是优秀的一种衡量方式，价值是产品价格的衡量方式。基于价值的观点的中心思想是，**客户愿意为某种质量水平花费多少**。事实上，如果产品没有经济意义，那么它的质量就是毫无意义的。本质上，基于价值的观点是在价格和质量之间的一种折中。

软件质量的5个观点-测量质量

度量方式使我们对质量概念有了**定量**上的认识。

● 用户的观点度量：

用户的观点涉及大量的质量因素，例如功能性、可靠性和易用性。通过为每个功能设计至少一个测试用例，可以轻易地测量出一个软件产品可以提供多少功能。假如功能是在不同的执行环境中实现，对于相同的功能，一个产品可能需要多个测试用例。那么，**通过的测试用例数量与为了验证功能性而设计的测试用例总数之间的比率是产品交付的功能的度量标准。**

软件质量的5个观点-测量质量

● 制造的观点度量

制造者对获得下面两种不同数量的度量有兴趣：

- **缺陷数量**：已经检测到多少缺陷？
- **返工成本**：修改已知缺陷需要花费多少代价？

缺陷数量表示到目前为止所检测到的所有缺陷的数量。缺陷数量反映出了生产工作的质量。计算出缺陷数量后，通过分析缺陷数量来改进开发过程，以使后续的项目中能减少缺陷数量。

修复缺陷需要返工，**返工的成本包括与缺陷有关的行为的所有额外成本**，例如修改文档。返工是由于在首次工作时采用不尽完美的方式而造成的额外的代价。



软件质量的5个观点-测量质量

● 制造的观点度量

返工成本可以分为：

- **开发返工成本**：这种返工代价产生在一个产品发布给客户之前。
- **运转返工成本**：这种返工代价在一个产品的运转过程中产生。

一方面，**开发返工代价**是开发效率的一种衡量标准。假如开发返工代价是零，则开发的效率是非常高的。另一方面，**运转返工代价**是产品在运转中交付的质量的衡量标准。假如开发返工代价是零，则产品在运转中交付的质量是非常高的。这是因为客户没有遇到任何缺陷，因此，开发团队不需要在修复缺陷上花费任何资源。



目录

CONTENTS

- 软件质量的5个观点
- **McCall的质量因子和标准**
- ISO 9126质量特性
- ISO 9000:2000基本法则

McCall的质量因子和标准-质量因子

质量因子代表了一个系统的一种行为特征，是一个软件系统的外部属性。表17.1列出了McCall及其他人定义的11个质量因

表 17.1 McCall 的质量因子

质量因子	定 义
正确性	程序满足它的规范并完成用户的任务目标的程度
可靠性	程序在要求的精度下能按预期来执行它的目标功能的程度
有效性	程序执行一个功能所需要的计算机资源和代码量
完整性	能够控制未经授权的人访问软件或者数据的程度
易用性	在程序的学习、操作、准备输入和解释输出方面所需要的努力
可维护性	在运行的程序中定位和修复缺陷所需要的努力
易测性	测试程序以确定其执行目标功能所需要的努力
适应性	修改运行的程序所需要的努力
可移植性	把程序从一个硬件或者软件环境转移到另一个环境所需要的努力
可重用性	软件系统的一部分可以重用到另一个应用中的程度
互操作性	把一个系统和另一个系统联合起来所需要的努力



McCall的质量因子和标准-质量因子

在表17.1中定义的11种**质量因子可分成三大类**，见表17.2。

表 17.2 McCall 质量因子的分类

质量分类	质量因子	大的目标
产品操作	正确性	实现客户的需求了吗？
	可靠性	一直保持正确性吗？
	有效性	快速地解决了期望的问题吗？
	完整性	安全吗？
	易用性	可以运行吗？
产品修复	可维护性	可以修复吗？
	易测性	可以测试吗？
	适应性	可以改变吗？
产品转变	可移植性	可以在另一个机器上使用吗？
	可重用性	部件可以重用吗？
	互操作性	能和其他系统进行交互吗？



McCall的质量因子和标准-质量标准

质量标准
是与软件开发
有关的质量因
子的一个属性。
(23种)。

表 17.3 McCall 的质量标准

质量标准	定 义
访问审计	很容易进行对软件和数据是否符合标准与需求的检查
访问控制	提供对软件和数据的控制与保护
准确性	计算和输出的准确度
通信公共	使用标准协议和接口的程度
完全性	已经达到的所需功能完全实现的程度
通信性	容易对输入/输出进行分析
简明性	源代码紧凑, 依据代码行数判断
一致性	在一个项目过程中使用统一的设计、实现技术和标记
数据公共	标准数据表示的使用
容错性	在不利条件下确保持续工作的程度
执行效率	软件的运行时间效率
扩展性	储存需求或者软件功能可以扩展的程度
通用性	软件组件的潜在应用范围
硬件独立性	软件依赖基础硬件的程度
度量性	软件提供使用度量或者识别错误的程度
模块性	高独立模块的提供
可操作性	很容易操作软件
自我描述性	提供在线文档, 解释组件实现
简单性	能够很容易理解软件
软件系统独立性	软件独立于它的运行环境的程度, 例如非标准的语言结构、操作系统、库函数、数据库管理系统等
软件效率	软件的运行时内存需求
可追溯性	连接软件组件到需求的能力
可培训性	新用户可以很容易地使用系统



质量因子和质量标准的关系

质量因子和质量标准之间的关系如图17.1所示。

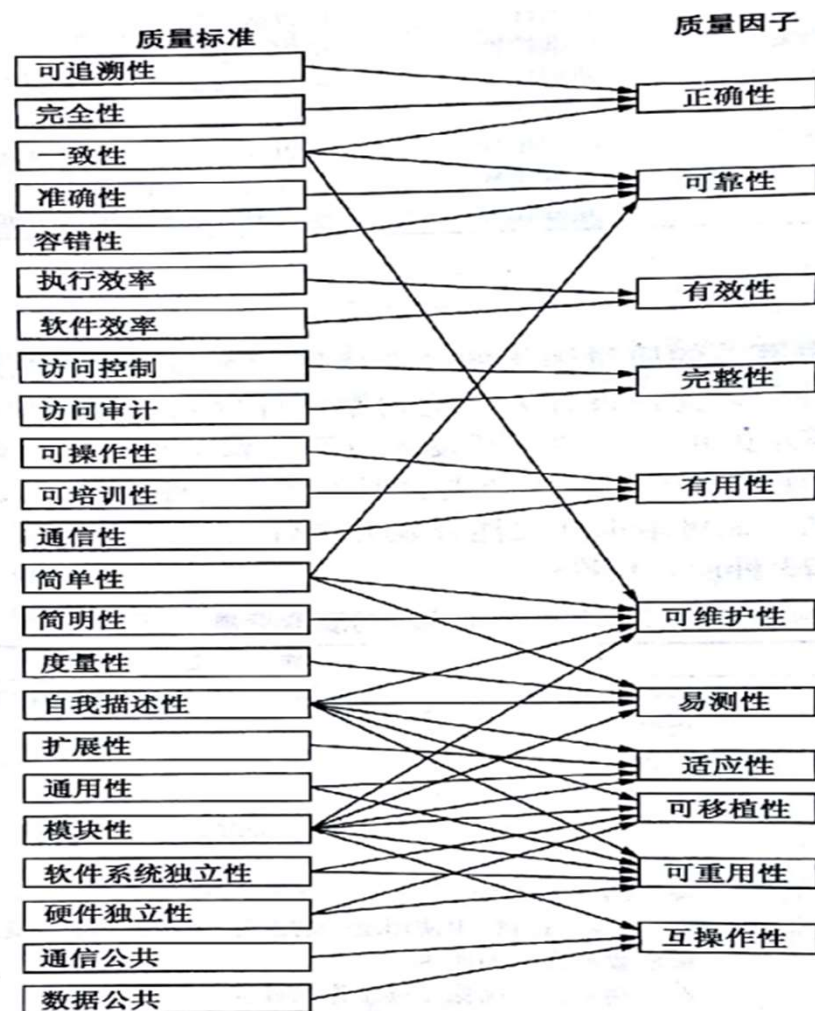


图 17.1 质量因子与质量标准之间的关系^[6]

McCall的质量因子和标准-质量度量

由于质量因子并不是完全独立的，因此我们不能期望改善所有的质量因子。

我们注意到如下关系的两种特性：

- **如果努力去改善一种质量因子，可能另一种质量因子就会下降。** 努力使软件可测试，则软件效率可能下降。
- **一些质量因子会正面地影响其他因子。** 例如，增强系统正确性的努力会增加其可靠性。作为另一个例子，增强系统易测性的努力会改善其可维护性。

McCall的质量因子和标准-质量度量

质量度量是一种衡量，它关注一个质量标准的某些方面。每个标准都对应一到多个质量度量。

度量可以从下面得到：

- 列出有关**质量标准**的相关问题集，并为每个问题找到“是”或者“否”的答案。
- 把答案为“是”的数目除以问题总数，得到一个介于0~1之间的值。结果数字代表目标质量度量。

目录

CONTENTS

- 软件质量的5个观点
- McCall的质量因子和标准
- ISO 9126质量特性
- ISO 9000:2000基本法则

ISO 9126质量特性

ISO 9126 定义了6个如下所示的质量特性的分类：

功能性：一组属性集，它对一组功能及它们指定特性的存在施加影响。这些功能将满足指定或者隐含的需求。

可靠性：一组属性集，它对软件在一段指定时间内在指定条件下维持其性能的能力施加影响。

易用性：一组属性集，它对使用产品所需要的努力和通过指定或者暗示用户集对使用效果进行评价施加影响。

ISO 9126质量特性

ISO 9126 定义了6个如下所示的质量特性的分类：

有效性：一组属性集，它对在指定条件下软件性能和资源使用数量之间的关系施加影响。

可维护性：一组属性集，它对进行特定修改所需要的努力施加影响（这些修改可能包括修复、改善，或者软件对于环境变化的适应和在需求和功能规范中的变更）。

可移植性：一组属性集，它对软件从一个环境转移到另一个环境（这包括组织的、硬件的或者软件的环境）的能力施加影响。

ISO 9126质量特性

ISO 9126质量模型实例

如图17.2所示，它将质量特性分解为更为具体的子特性。

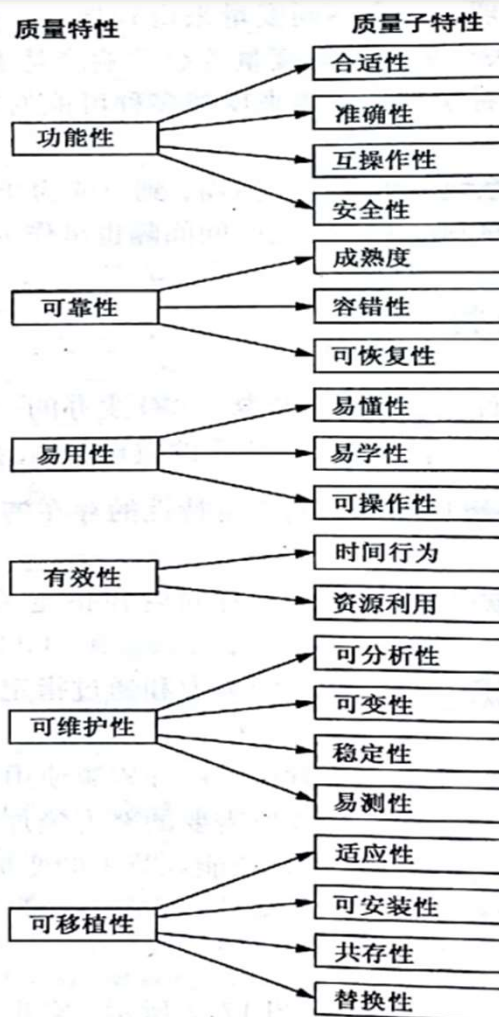


图 17.2 ISO 9126 样本质量模型把标准特性细化成子特性(引自参考文献[4], © 1996 IEEE)

ISO 9126质量特性

组织必须在充分理解需求之后，定义自己的质量特性和子特性，在软件开发环境中识别他们需要满足的不同质量特性的水平。因此理解在向最高水平前进的过程中，进行到下个可实现的步骤的需求是非常重要的。

在这一点上，比较 McCall 的质量模型和ISO9126模型是非常有用的。

ISO 9126质量特性

比较McCall 的质量模型和ISO9126模型

相似点:

- 两个模型都把注意力放在相同的抽象实体上（即软件质量）。
- McCall 模型中的质量因子在 ISO 9126模型中称为质量特性。
- 可靠性、易用性、有效性、可维护性和可移植性等高层质量因子 / 特性可以在两个模型中找到。

ISO 9126质量特性

比较McCall 的质量模型和ISO9126模型

不同点:

- ISO 9126强调对于用户可见的特性，而 McCall模型考虑内部质量。
- 在 McCall 模型中，一个质量标准能影响到多个质量因子，而在ISO9126中，一个子特性只能影响一个质量特性。
- McCall 模型中的一个高层质量因子（例如易测性），在ISO 9126中是可维护性的一个低层子特性。

目录

CONTENTS

- 软件质量的5个观点
- McCall的质量因子和标准
- ISO 9126质量特性
- **ISO 9000:2000软件质量标准**

ISO 9000:2000基本法则

ISO 9000:2000标准有三个部分，即**ISO 9000**、**ISO 9001**和**ISO 9004**。

ISO 9000文档是**有关基本功能和词汇表**的，**ISO 9001**文档是**有关需求**的，而**ISO 9004**文档**包含性能改进的指南**。

ISO 9000:2000基本法则

ISO 9000文档基于8个原理：**客户焦点、领导力、人员参与、过程方法、系统管理方法、持续的改善、基于事实的决策方法及互利的供应商关系。**

ISO 9001文档处理如下的5种需求：**系统需求、管理需求，资源需求、实现需求和补救需求。**

总结

软件质量的5个观点：抽象的观点，用户的观点，制造的观点，产品的观点，基于价值的观点。

另一个软件质量的概念，通常称为质量因子和质量标准。质量因子表示了一个系统的行为特性，**11种质量因子：**正确性、可靠性、有效性、完整性、易用性、可维护性、易测性、适应性、可移植性、可重用性和互操作性。

总结

这**11种质量因子**分成了**3类**：产品操作、产品修复和产品转变。

①产品操作关注正确性、可靠性、有效性、完整性和易用性。

②产品修复关注可维护性、易测性和适应性。

③产品转变关注可移植性、可重用性和互操作性。

表 17.2 McCall 质量因子的分类

质量分类	质量因子	大的目标
产品操作	正确性	实现客户的需求了吗？
	可靠性	一直保持正确性吗？
	有效性	快速地解决了期望的问题吗？
	完整性	安全吗？
	易用性	可以运行吗？
产品修复	可维护性	可以修复吗？
	易测性	可以测试吗？
	适应性	可以改变吗？
产品转变	可移植性	可以在另一个机器上使用吗？
	可重用性	部件可以重用吗？
	互操作性	能和其他系统进行交互吗？



总结

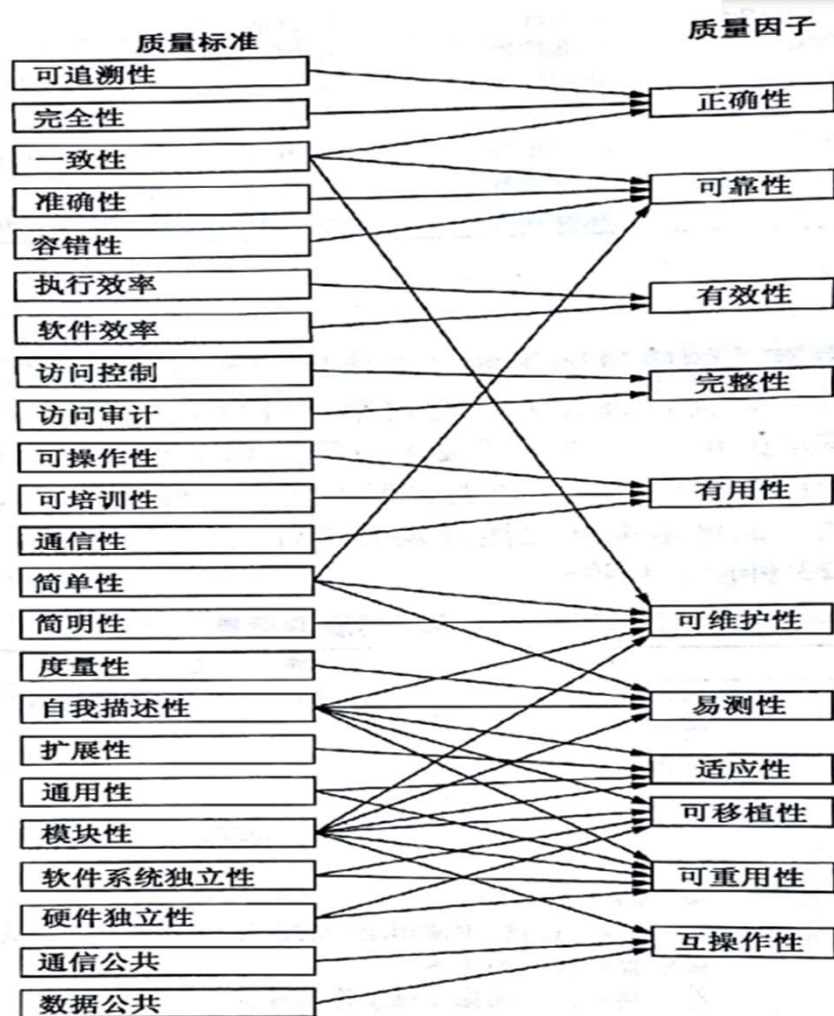
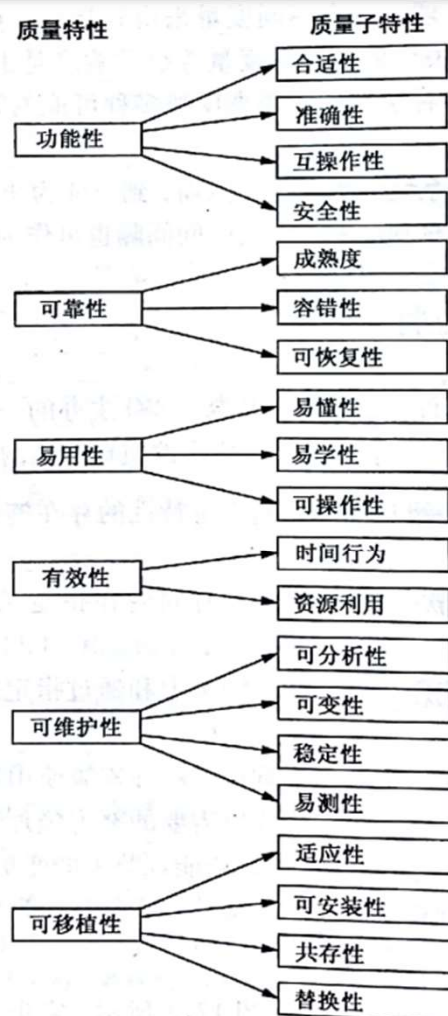


图 17.1 质量因子与质量标准之间的关系^[6]



ISO 9126 样本质量模型把标准特性细化成子特性(引自参考文献[4], © 1996 IEEE)

总结

ISO 9000 : 2000标准有三个部分, 即 ISO 9000 , ISO 9001和 ISO 9004。**ISO 9000文档**是有关基本功能和词汇表的, **ISO 9001文档**是有关需求的, 而**ISO 9004文档**包含性能改进的指南。

软件质量

End
