

提高软件质量的方法

软件学院 14126172 于浩川

软件产业已进入高速发展时期，软件产品质量越来越受到人们的重视，软件质量管理体系是否健全也渐渐地为人们所重视，软件质量贯彻着整个软件工程领域，直接影响着从开发到测试再到管理的各个层面。随着科技的茂展，软件质量广泛的被关系国计民生的各行各业所接受，越发显得软件质量的重要性。软件质量管理需求的考虑是方方面面的，有市场因素，竞争力因素，成本因素和实效性。总体说来，软件能做到易改、易测、易用、易护，质量才有保障。

● 何为软件质量。

国际化标准组织ISO在ISO / IEC 9126中将软件质量定义为：反映软件产品满足规定需求和潜在需求能力的特征和特征的总和。M. J. Fissler将软件质量定义为：所有描述计算机软件优秀程度的特性的组合，也就是说，为了满足软件的各项精确定义的功能、性能要求，符合文档化的开发标准，需要相应的给出或设计一些质量特性及其组合。要得到高质量的软件产品，就必须使这些质量特性得到满足。目前对软件质量特性有多种提法，但实际上是大同小异。ISO / IEC 9126国际标准中定义的软件质量特性为以下六项：功能性(functionality)；可靠性(reliability)；易使用性(usability)；效率(efficiency)；可维护性(maintainability)；可移植性(portability)。

软件质量的最终定义尚未出现，但是有关提高软件质量的研究已经全面开展。目前，软件质量主要从两方面展开研究：(1)软件开发过程的质量保证，以过程文档化和科学化管理为内容；(2)软件过程和产品的质量评估，包括中间产品和最终产品，采用软件度量技术作为软件质量特性量化的主要技术。那么，这样看来，软件质量应包括两方面的内容，即软件过程质量和软件产品质量。目前流行的标准和模型充分体现了这一点，其中最具代表性就是ISO系列标准和CMM。

● 软件质量保证的概念与目标。

软件质量保证就是确保软件产品从诞生到结束为止的所有阶段的质量的活动，即确定、达到和维护需要的软件质量而进行的有计划、有系统的管理活动，向用户及社会提供满意的高质量产品，并向管理层保证拟定出的标准、步骤、实践和方法能够正确地被所有项目所采用。

软件质量保证(Software Quality Assurance, 简称SQA)的目标是以独立审查方式，从第三方的角度监控软件开发任务的执行，就软件项目是否正确遵循已制定的计划、标准和规程给开发人员和管理层提供反映产品和过程质量的信息和数据，提高项目透明度，同时辅助软件工程组取得高质量的软件产品。简单说来，也即验证在软件开发过程中是否遵循了合适的过程 and 标准，主要包括以下四个方面：

1. 通过监控软件开发过程来保证产品质量；

2. 保证开发出来的软件和软件开发过程符合相应标准与规程；
3. 保证软件产品、软件过程中存在的不符合问题得到处理，必要时将问题反映给高级管理者；
4. 确保项目组制定的计划、标准和规程适合项目组需要，同时满足评审和审计需

● 影响软件质量的因素

软件质量问题主要来源于软件的开发过程，而影响开发过程质量的主要特性包括：开发所采用的技术、开发者个人的业务经历水平及开发所使用的工具，常见影响软件质量的因素归纳总结有以下几个方面。

1. 需求分析因素。对用户的提出的需求沟通不到位，转换为软件需求规格说明不完整；用户需求变更管理不到位，用户需求可能存在变更，而软件开发者没有及时更改软件或软件更改引入新的问题等。
2. 软件设计方法因素。软件开发者往往是先编码后写设计方案，或者按照自己的编程习惯开发软件，导致软件不符合用户需求，或未考虑用户对特殊情况处理和错误处理情况等。
3. 编码过程因素。主要表现在软件编码不规范，软件容错能力不够等。
4. 内部测试过程因素。如内部测试未开展或内部测试不充分等。
5. 软件文档因素。如文档描述存在二义性、文文之间描述不一致，文档版本不一致等。

● 软件质量保证方法

1. 建立以软件质量为核心的质量体系

建立树立为软件质量为核心的软件质量管理体系，如依据软件能力成熟度模型集成(CMMI)建立管理体系，明确在软件开发的各个阶段必须控制的质量指标。软件项目组严格按照质量体系开发软件，质量保证人员必须全程跟踪监督。软件质量体系度量包括以下方面。

(1) 需求分析质量度量。软件需求分析是否充分、需求是否完整、需求定义是否准确，是否存在需求理解不同的情况，软件需求规格说明文档是否覆盖软件任务书或合同中所有要求。

(2) 软件设计质量度量。在软件设计过程，是否进行了概要设计和详细设计，软件设计是否满足软件需求，软件设计文档是否详细等。

(3) 测试结果质量度量。内部测试如单元测试、单元集成和测试、配置项测试、系统合格性测试是否都逐层开展，发现的问题是否都进行了处理，测试是否依据了软件需求文档等。

(4) 验收结果质量度量。是否组织开展软件验收工作，对软件完成的功能数量，各项性能指标等是否进行了评审，各项功能性能质量是否满足用户需求等。

2. 选择最合适的开发方法

软件开发方在需求确定以后，应根据软件特性选择最合适的开发方法。目前软件开发方法主要有Parnas法、Jackson法、Yourdon法等，面向对象方法、原型化方法、可视化方法等。其中可视化方法适用于图形类应用软件开发；面向数据结构的方法及原型化方法适合于中小型系统的开发。优先推荐使用Parnas法，该方法是1972年Parnas提出来的，基本思想是在设计时提前预测软件可能存在的

需求变更，通过对可能变化的信息集中在某些模块内，使其与其他模块无关，提高了软件的可维护性，避免了错误的蔓延，也就提高了软件的可靠性。

3. 采用软件重用技术

软件重用是指在开发新软件的过程中重复使用已有的软件成分，该软件成分可能是已存在的软件，也可能是专门设计的可重用的软件构件。广义的软件重用还包括软件的开发思想方法、文档、环境、数据等。最大限度地采用软件重用技术，不仅能缩短开发周期，提高开发效率，也能提高软件的可维护性和可靠性。

4. 加强软件容错设计

容错是指系统或软件出现有限数目的硬件或软件故障的情况下，系统仍具有连续正确执行任务的固有能力和能力。其主要目的是提供足够的冗余信息和合适的算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高软件可靠性。

软件容错设计有冗余设计与非冗余设计两类方法。冗余设计有软件冗余、硬件冗余、时间冗余和信息冗余等。其中软件冗余又有表决器、N 版本和恢复块等方法在软件应用较少，不作为本文论述的重点。非冗余设计有保护处理、异常处理、容差处理看门狗等技术等。

提高软件质量的方法有多种途径，软件开发方应根据具有软件特点采取不同的质量方法，选择最合理的开发方法，采用最可靠、最成熟的技术和，加强软件内部测试和开发过程的质量管理工作，才能有效保证软件最终质量。