

软件质量保障

软件质量保障背景

- 在软件开发过程中，即使采用了多种质量保证方法，软件产品中仍可能存在错误和缺陷，特别是对于规模大、复杂性高的软件。
- 软件测试是保证软件质量的关键步骤，是对规约、设计和编码的最后复审。

软件质量保障定义

- **质量保证**：旨在衡量和提高产品质量的所有活动，包括整个开发过程、培训和团队准备。
- **质量控制**：旨在验证产品质量、检测故障或缺陷并确保在发布前修复缺陷的活动。

软件质量控制技术

- **测试**：设计测试用例、在受控环境中执行程序并验证输出是否正确过程。
- **检查和评审**：可应用于程序或中间软件工件，需要多人参与，是发现错误的一种有效方法。
- **形式化方法**：用来“证明”程序正确性的数学方法，很少应用于商业软件。
- **静态分析**：对程序或中间软件产品的静态结构进行分析的过程，通常是自动进行的。

质量的定义

- 质量可以定义为**符合规约或达到目的**，这两个定义并不等价。

软件的确认和验证

- **验证(Verification)**：检查软件生存期各个阶段过程活动的结果是否满足规约的描述。
- **确认(Validation)**：证实在一个给定的外部环境中软件的逻辑正确性，即是否满足用户的要求。

软件失败的术语

- 包括缺点、偏差、谬误、失效、问题、矛盾、错误、特殊、特征、毛病、缺陷、异常和漏洞等。

显式和隐式的需求和规约

- 显式需求需要在需求文档中明确，而隐式规约虽不具有权威性，但却是很好的参考，应尽可能遵循。

区分缺陷的严重性和优先级

- **严重性**：衡量缺陷可能对用户或组织造成的影响或后果。
- **优先级**：衡量缺陷在开发组织眼中的重要性。

软件测试

- 测试是通过发现缺陷和问题来评估产品质量和改进产品质量的活动。

测试的目的

- 发现软件中的缺陷，以便纠正或减少这些缺陷。
- 对质量进行总体评估，包括在一定程度上保证产品在许多考虑过的情况下都能正常运行，并对遗留的缺陷进行估计。

传统的测试技术

- 白盒测试：关注软件的内部逻辑和代码结构。
- 黑盒测试：从用户的角度出发，测试软件的功能是否符合需求。

最新的测试技术

- 模糊测试、差分测试、蜕变测试。

静态分析

- 基于静态分析的方法通过语法和浅层次语义分析预测程序的行为，寻找缺陷存在的证据。

形式化方法

- 用数学技术来证明程序是绝对正确的方法。

检查和审查

- 由软件开发人员团队对代码或中间文档进行审查是最经济有效的错误检测技术之一。

桌面检查

- 由程序员自己检查自己编写的程序，对照缺陷列表进行检查，对程序推演测试数据，并补充相关的文档。

代码检查

- 审查小组最好由四人组成：组长、程序的设计者、程序的编写者和程序的测试者。

走查

- 文件作者向一小组人员解释软件产品，参与者对可能发生的错误、违背开发标准的行为以及其他问题进行提问与批注。