

第1章 软件质量.

一. 质量观

1. 用户: 符合使用目的(重点: 需求分析和定义, 明确冻结需求).
2. 产品: 性质的总和(软件质量因素).
3. 生产者: 是否满足给定需求和规约.

二. 定义:

1. 计算机软件质量保证计划规范: 软件产品中能满足给定需求的各种特征总和.
2. IEEE: 系统、组件、过程满足指定要求的程度, 满足用户需求和期望的程度.
3. AWSI: 满足规定的和隐含的与需求能力有关的全部特征和特性.

三. 软件质量模型.

1. McCall: 软件质量分成多个质量要素, 每个质量要素涵盖多个衡量标准, 每个衡量标准由一系列度量构成。(重要方面: 产品操作, 产品修改, 产品改型).
2. Boehm: 分层结构, 除了用户期望和需求以外, 包括了硬件特性.
3. ISO/IEC 9126: 将质量模型分为内部质量模型, 外部质量模型和使用质量模型。
内/外部质量模型分为6个一级性质和21个二级性质.
4. Perry: 将关系模型反应质量要素之间的正面、反面和中立关系.

第2章 软件质量保证.

一. 软件质量管理

在软件质量方面指挥和控制组织的协调活动.

方面: 事先的预防措施, 事中的跟踪监控措施, 事后纠错措施.

二. 软件质量体系.

1. CMM: 五个成熟度等级.
2. ISO9000-3: 用于计算机软件开发、供应、安装、维护中的指南.



ISO 9000 聚焦于供应商与用户间的关系，CMM 关注软件开发过程。

三. 软件质量保证

1. 定义：
 - ① 确保软件从诞生到消亡所有阶段的质量活动。
 - ② 有计划的、系统的行动模式（提供充分信任所必须的信息）。
 - ③ 提高软件交付时的水平。
2. 目标：减少并纠正软件开发过程和软件开发结果与预期不符的情况。

第3章 ~~软件~~软件缺陷

一. 软件缺陷.

- 定义：
 - ① 产品：开发/维护过程中存在的错误等问题。
 - ② 外部：系统所需要实现的某种功能的失败或违背。

产生主要原因：规格说明书错误。

二. 软件测试中的术语.

fault：导致系统功能失效的异常状态。

error：条件与理论上正常值有差异。

failure：系统不能执行所要求的功能。



第3章 软件缺陷.

软件测试术语:

1. Verification (验证): 证明在软件存在期各个阶段, 以及阶段间的逻辑协调性、完备性、正确性.

Validation (确认): 证实软件在一个给定外部环境中的逻辑正确性.

2. 软件测试与编程调试的区别.

① 测试是为了发现软件中存在的错误, 调试是为了证明软件开发的正确性.

② 测试以已知条件开始, 使用预先定义程序, 且有预知结果; 调试以不可知内部条件开始, 结果不可预见.

③ 测试可以在不了解软件设计条件下完成; 调试必须由了解详细设计的开发人员完成.

软件测试的目的:

① 在于发现错误。一个成功的测试是发现了至今未发现的错误

② 保证软件质量.

软件测试相关工作:

1. 制定测试计划: 定义每一阶段的功能、角色、方法.

2. 组织测试人员: ① 开发人员测试 (针对性强, 但回避问题).

② 独立测试 (全面, 但学习软件成本高).

③ 用户测试 (指出需求问题, 但不全面).

3. 编写测试用例.

4. 测试实施

5. 测试结果反馈 (测试文档)

6. 回归测试: 对测试后修改的程序, 执行以前执行过的测试用例, 保证没有引入新的错误.

测试用例:

采用少量测试数据就能达到最佳测试效果.

软件测试原则:

① 穷尽测试不可行 ② 测试尽早开始 ③ 缺陷集群性 ④ 没有失效即有用是谬论.



§1. 软件测试模型.

1. V模型: 反应了测试活动与分析和设计的关系.

局限性: 需求和系统设计错误到最后才能被发现.

2. W模型: 强调测试伴随着整个软件开发周期. (开发与测试同步进行).

局限性: 把软件开发视为一系列串行活动且无法迭代开发模型.

3. H模型: 将测试活动独立出来, 形成完全独立的流程.

4. X模型: 提出了测试设计.

5. 前置模型: 将开发和测试生命周期整合在一起, 伴随项目开发周期及每个关键行为.

§2. 测试生命周期.

1. 拟定测试计划: 定义测试项目的阶段、需求、策略、资源、进度

2. 编制测试大纲 3. 设计测试用例 4. 实施测试 5. 生成测试报告.

§3. 测试计划

谁在什么时间用什么环境测什么项目用什么方法/工具.

§4. 测试设计与开发.

测试设计目的: 为每一个测试需求确定测试用例集, 并且确定执行测试用例的测试过程.

具体: 对需求确定测试用例; 对用例确定其输入及预期结果; 确定测试用例的测试环境配置、需要的驱动程序, 编写测试用例文档, 评审.

§5. 测试实施评估.

功能冻结 代码冻结.
↓ ↓
测试实施阶段: 初期期 → 细测期 → 回归测试期



第6章 静态测试

静态测试：对组件/系统进行规格或实现级别测试（不执行代码）。

包括：桌面检查、代码审查、代码走查。

代码走查：非正式的，由程序员组织讨论的，只检查代码是否有错误。

代码审查：正式的评审活动，由高级管理人员领导，检查程序和设计文档一致性。

测试内容：

① 需求定义的静态测试：

测试对用户需求描述和解释是否完整、正确。

② 设计文档的静态测试：

着重分析设计是否与需求定义一致。

③ 源代码静态测试：

着重于分析实现是否正确、完备。

第8章 面向对象测试

8.1. 面向对象单元测试。

以类为单元：① 不发生请求的类：设计驱动程序，封装在测试类中。

② 发生请求的类：设计桩程序，封装在桩中。

8.2. 面向对象集成及系统测试。

集成测试：检查构件、子系统的接口。（类间测试）。

集成策略：① 基于线程 ② 基于使用。

系统测试：测试是否满足用户需求，是否存在某些条件下诱发错误的隐患。

集中于：系统性能、强度、安全。



§1. 软件测试管理.

~~软件测试~~

1. 测试项目管理:

目标: 对成本、资源、进度、质量、风险进行分析和控制活动.

以测试项目为管理对象, 进行计划、组织、执行和控制.

2. 测试管理计划: 了解被测对象的系统方法、编程语言特点等, 确定必要测试环境.

3. 测试文档: 描述、定义、规定和报告的任何书面或图示信息.

4. 测试人员组织: 分为需求分析测试团队、设计测试团队、程序测试团队.

5. 测试过程管理: 集中在软件测试项目的启动、测试计划、测试用例执行、测试结果审查分析.

§2. 自动化~~管理~~测试.

由软件工具来代替手工进行一系列动作.

优点: 回归测试更方便, 执行手工测试困难的测试, 测试的复用性.

自动化测试原理:

①代码分析 ②录制回放 ③脚本技术 ④虚拟用户技术.

第10章 软件缺陷生命周期.

§1. 生命周期

指一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程.

发现 → 打开 → 修复 → 关闭.

§2. 软件缺陷描述.

包含: 严重性、优先级、缺陷类型、产生可能性、来源、原因.



§3. 缺陷跟踪分析.

缺陷跟踪管理: 确保每个被发现的缺陷能够及时得到处理.

分析方法: ①缺陷分析报告 ②缺陷趋势报告.

③缺陷年龄报告 ④测试结果进度报告.

§4. 软件测试评估.

目的: ①量化测试进程, 判断测试进行状态, 决定什么时候结束.

②为编写测试报告或质量分析报告提供所需数据.

基本方法: ①基于需求的测试覆盖指标

②基于代码的测试覆盖指标.

第11章 性能测试.

§1. 基本知识.

性能测试: 指在一定条件下系统行为表现符合需求规格的性能指标.

§2. 分类:

1. 负载测试: 观察响应时间, 数据吞吐量, 系统占用资源等性能指标.

2. 压力测试: 又称强度测试. 发现系统功能隐患、系统是否具有良好的容错能力和可恢复能力.

3. 可靠性测试: 在给定时间内无故障地运行的概率.

4. 安全性测试: 在外处理非授权的内/外部访问、非法入侵或故意损坏时系统的防护能力.

5. 兼容性测试: 软件与先前发布过的版本, 有依赖外部软件, 运行版本, 硬件平台兼容情况.

6. 可用性测试: 系统正常运行能力和用户接受程度.

