

软件工程中的测试技术

第二部分

目录

- (6) 单元测试和集成测试
- (7) 系统测试
- (8) 软件性能测试和可靠性测试
- (9) 其他测试
- (10) 软件自动化测试

单元测试和集成测试

- 6.1 单元测试的基本概念

- 定义和目标

- 单元是可以编译和执行的最小软件构件。
 - 单元是决不会指派给多个设计人员开发的软件构件。
 - 单元测试是在软件开发过程中要进行的最低级别的测试活动，或者说是针对软件设计的最小单位——程序模块，进行正确性检验的测试工作。

单元测试和集成测试

- 6.1 单元测试的基本概念

- 定义和目标

- 目标：

- 验证代码是与设计相符合的。

- 跟踪需求和设计的实现。

- 发现设计和需求中存在的缺陷。

- 发现编码过程中引入的错误。

单元测试和集成测试

- 6.1 单元测试的基本概念
 - 区别
 - 单元测试和集成测试的区别
 - 单元测试与系统测试的区别

单元测试和集成测试

- 6.2 单元测试策略

- 自顶向下的单元测试策略

- 从最顶层开始，把顶层调用的单元用桩模块代替，对顶层模块做单元测试。
 - 对第二层测试时，使用上面已测试的单元作驱动模块，并为测试模块编写新的桩模块。
 - 依次类推，直到全部单元测试结束。

单元测试和集成测试

- 6.2 单元测试策略

- 自顶向下的单元测试策略

- 优点：可以在集成测试之间为系统提供早期的集成途径。
 - 缺点：单元测试被桩模块控制，随着单元测试的不断进行，测试过程也会变得越来越复杂，测试难度、开发和维护的成本不断增加，要求的低层次结构覆盖率也难以得到保证。

单元测试和集成测试

- 6.2 单元测试策略

- 自底向上的单元测试策略

- 先对模块调用图上的最底层模块进行测试，使用驱动模块来代替调用它的上层模块。
 - 对上一层模块进行单元测试时，用已经被测试过的模块作桩模块，并为被测模块编写新的驱动模块。
 - 依次类推，直到全部测试结束。

单元测试和集成测试

- 6.2 单元测试策略

- 自底向上的单元测试策略

- 优点：不需要单独设计桩模块；无需依赖结构设计，可以直接从功能设计中获取测试用例；可以为系统提供早期的集成途径；在详细设计文档中缺少结构细节时可以使用该测试策略。
 - 缺点：随着单元测试的不断进行，测试过程会变得越来越复杂，测试周期延长，测试和维护的成本增加。

单元测试和集成测试

- 6.2 单元测试策略

- 孤立测试

- 孤立测试不需要考虑每个模块与其他模块之间的关系，分别为每个模块单独设计桩模块和驱动模块，逐一完成所有单元模块的测试。
 - 优点：方法简单、容易操作，因此所需测试时间短，能够达到高覆盖率。
 - 缺点：不能为集成测试提供早期的集成途径。依赖结构设计信息，需要设计多个桩模块和驱动模块，增加了额外的测试成本。

单元测试和集成测试

- 6.2 单元测试策略

- 综合测试

- 在单元测试过程中，编写桩模块的工作量是相当大的。为了有效地减少开发桩模块的工作量，可以考虑综合自底向上测试策略和孤立测试策略。

单元测试和集成测试

- 6.3 单元测试分析
 - 模块接口
 - 局部数据结构
 - 独立路径
 - 出错处理
 - 边界条件

单元测试和集成测试

- 6.4 集成测试的基本概念

- 定义

- 根据实际情况对程序模块采取适当的集成测试策略组装起来，对系统的接口以及集成后的功能进行正确性检验的测试。

- 集成测试与系统测试的区别

- 测试对象
 - 测试时间
 - 测试方法
 - 测试内容

单元测试和集成测试

- 6.4 集成测试的基本概念
 - 集成测试与系统测试的区别
 - 测试目的
 - 测试角度
 - 集成测试与开发的关系
 - 集成测试重点
 - 集成测试层次
 - 内集成测试、子系统内集成测试、子系统间集成测试

单元测试和集成测试

- 6.4 集成测试的基本概念
 - 集成测试环境
 - 硬件环境
 - 操作系统环境
 - 数据库环境
 - 网络环境
 - 测试工具运行环境
 - 其他环境

目录

- (6) 单元测试和集成测试
- (7) 系统测试
- (8) 软件性能测试和可靠性测试
- (9) 其他测试
- (10) 软件自动化测试

系统测试

- 7.1 系统测试概念

- 什么是系统测试

- 系统测试就是将已经集成好的软件系统，作为整个计算机系统的一个元素，与计算机硬件、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行（使用）环境下，对计算机系统进行一系列的集成测试和确认测试。
 - 系统测试属于黑盒测试范畴，不再对软件的源代码进行分析和测试。

系统测试

- 7.1 系统测试概念
 - 系统测试与单元 / 集成测试的区别
 - 测试方法不同。
 - 考察范围不同。
 - 评估基准不同。

系统测试

- 7.2 系统测试的方法
 - 功能测试
 - 协议一致性测试
 - 性能测试
 - 压力测试
 - 容量测试
 - 安全性测试
 - 失效恢复测试
 - 备份测试

系统测试

- 7.2 系统测试的方法

- GUI 测试
- 健壮性测试
- 兼容性测试
- 易用性测试
- 安装测试
- 文档测试
- 在线帮助测试
- 数据转换测试

系统测试

- 7.2 系统测试的实施

- 确认测试

- 确认测试 (Validation Testing) 又称有效性测试，其任务是确认软件的有效性，即确认软件的功能和性能及其他特性是否与用户的要求一致。

- α 测试和 β 测试

- α 测试是用户在开发环境下进行的测试，也可以是产品供应商内部的用户在模拟实际操作的环境下进行的测试。
 - β 测试是由软件的多个用户在一个或多个用户的实际使用环境下进行的测试。

系统测试

- 7.2 系统测试的实施

- 验收测试

- 验收测试是以用户为主的测试。由用户参加设计测试用例，使用用户界面输入测试数据，并分析测试的输出结果。

- 回归测试

- 回归测试是在软件变更之后，对软件重新进行的测试。回归测试的目的是检验对软件进行的修改是否正确，保证（由于测试或者其他原因的）改动不会带来不可预料的行为或者另外的错误。

目录

- (6) 单元测试和集成测试
- (7) 系统测试
- (8) 软件性能测试和可靠性测试
- (9) 其他测试
- (10) 软件自动化测试

软件性能测试和可靠性测试

- 8.1 软件性能测试的基本概念

- 什么是软件性能

- 软件性能是软件的一种非功能特性，它关注的不是软件是否能够完成特定的功能，而是在完成该功能时展示出来的及时性。

- 软件性能的指标

- 响应时间
 - 系统响应时间和应用延迟时间
 - 吞吐量
 - 并发用户数
 - 资源利用率

软件性能测试和可靠性测试

- 8.1 软件性能测试的基本概念
 - 软件性能的测试
 - 性能测试 (Performance Testing)
 - 并发测试 (Concurrency Testing)
 - 压力测试 (Stress Testing)
 - 可靠性测试 (Reliability Testing)
 - 负载测试 (Load Testing)
 - 配置测试 (Configuration Testing)
 - 失效恢复测试 (Recovery Testing)

软件性能测试和可靠性测试

- 8.2 软件性能测试的执行
 - 性能测试的过程与组织
 - PTGM 测试过程模型
 - 测试前期准备
 - 引入测试工具
 - 制定测试计划
 - 测试设计与开发
 - 测试执行与管理
 - 测试结果分析

软件性能测试和可靠性测试

- 8.2 软件性能测试的执行
 - 性能分析
 - 性能下降曲线的分析
 - 性能平坦区
 - 性能轻微下降区
 - 性能急剧下降区
 - 性能计数器的分析
 - 内存分析
 - 处理器分析
 - 磁盘 I/O 分析
 - 进程分析

软件性能测试和可靠性测试

- 8.2 软件性能测试的执行
 - 性能测试的自动化
 - 压力的产生的调度
 - 性能监控
 - 结果分析
 - 性能测试工具

软件性能测试和可靠性测试

- 8.3 软件可靠性的概念

- 错误、缺陷、故障与失效

- 软件错误是指软件生存周期的所有阶段软件的状态或行为与人们预期的软件状态或行为的偏差。
 - 软件缺陷泛指软件中一切不好的东西，因此可以说，软件缺陷的概念涵盖了软件错误，比软件错误更广泛。
 - 软件故障是指软件代码中的错误。
 - 软件失效是指由软件故障引起的在软件运行期间出现的错误。

软件性能测试和可靠性测试

- 8.3 软件可靠性的概念

- 软件可靠性定义

- 在规定的条件下，在规定的时间内，软件不引起系统失效的概率；
 - 在规定的周期内，在所述条件下程序执行所要求的功能的能力。

- 软件可靠性指标

- 通常用平均无失效时间 (Mean Time to Failure, MTTF) 来直观地表示软件的可靠性。
 - 所谓平均无失效时间，就是指软件运行后，到下一次发生失效的平均时间。

目录

- (6) 单元测试和集成测试
- (7) 系统测试
- (8) 软件性能测试和可靠性测试
- (9) 其他测试
- (10) 软件自动化测试

其他测试

- 9.1 兼容性测试

- 硬件兼容性测试

- 不同的硬件配置可能影响软件的性能，因此需要有针对性地进行测试。
 - 如果软件使用了某些硬件的特定功能，一般需要对此进行兼容性测试。
 - 硬件兼容性测试的具体内容
 - 与整机的兼容性
 - 与板卡及外部设备的兼容性

其他测试

- 9.1 兼容性测试

- 软件兼容性测试

- 通常需要针对与该软件可能发生交互的其他软件的兼容性进行测试。
 - 主要考虑以下方面
 - 与操作系统的兼容性
 - 与数据库的兼容性
 - 与浏览器的兼容性
 - 与中间件的兼容性
 - 与其他软件的兼容性
 - 平台软件的兼容性测试

其他测试

- 9.2 易用性测试
 - 易安装性测试
 - 安装手册的易用性
 - 安装的自动化程度
 - 安装的灵活性
 - 安装中断的处理
 - 修复安装和卸载
 - 多环境安装的支持

其他测试

- 9.2 易用性测试
 - 功能易用性测试
 - 业务符合度
 - 功能定制性
 - 功能的关联度
 - 数据的共享度
 - 用户约束的合理性
 - 反馈的及时性

其他测试

- 9.2 易用性测试

- 用户界面测试

- 界面整体测试

- 合理性测试
 - 一致性测试
 - 规范性测试

- 界面元素测试

- 窗口测试
 - 菜单测试
 - 文字测试
 - 图标测试

- 输入测试

其他测试

- 9.3 构件测试

- 与测试相关的构件特性

- 软件构件最受关注的特性是可复用性，人们希望只写一次构件，就可以无缝地将构件库中的构件集成到一个新的环境中。为了使用这样的构件来设计系统：
 - 构件本身具有高可靠性。
 - 构件可以安全地与系统其他部分进行交互。
 - 与测试相关的构件特性包括以下几个方面：
 - 可观察性，可追踪性，可控制性，可理解性。

其他测试

- 9.4 极限测试

- 极限编程基础

- 极限编程具有以下几主要特征：

- 简单的分析设计
 - 频繁的客户交流
 - 增量式开发
 - 连续的测试

其他测试

- 9.4 极限测试

- 极限测试

- 极限测试的过程

- 单元测试

- 验收测试

- 极限测试的实施

- 单元测试用例的生成

- 单元测试工具的使用

- 模拟对象的使用

其他测试

- 9.5 文档测试
 - 文档测试的范围
 - 开发文档
 - 可行性研究报告
 - 软件需求说明书
 - 数据库设计说明书
 - 数据要求说明书
 - 概要设计说明书
 - 详细设计说明书

其他测试

- 9.5 文档测试
 - 文档测试的范围
 - 管理文档
 - 项目开发计划
 - 测试计划
 - 测试报告
 - 开发进度报告
 - 开发总结报告

其他测试

- 9.5 文档测试
 - 文档测试的范围
 - 用户文档
 - 宣传材料
 - 包装材料
 - 用户许可说明书
 - 安装手册
 - 用户手册
 - 概念手册
 - 在线帮助
 - 示例和模板

目录

- (6) 单元测试和集成测试
- (7) 系统测试
- (8) 软件性能测试和可靠性测试
- (9) 其他测试
- (10) 软件自动化测试

软件自动化测试

- 10.1 自动化测试的原理与方法
 - 自动化测试的基本结构由六个部分组成
 - 构建存放程序软件包和测试软件包的服务器。
 - 存储测试用例和测试结果的数据库服务器。
 - 执行测试的运行环境。
 - 控制服务器。
 - Web 服务器。
 - 客户服务器。

软件自动化测试

- 10.1 自动化测试的原理与方法
 - 自动化测试的实现方法
 - 代码分析
 - 捕获回放
 - 脚本技术
 - 自动比较
 - 测试管理

软件自动化测试

- 10.2 自动化测试的限制
 - 不能取代手工测试
 - 手工测试比自动化测试发现的缺陷更多
 - 对测试质量的依赖性极大
 - 自动化测试不能提高有效性
 - 自动化测试可能会制约软件开发
 - 工具本身不具备想象力

软件自动化测试

- 10.3 自动化测试用例的生成
 - 从目前的研究结果来看，能够实用化的测试设计或测试输入生成的自动化技术还不成熟。
 - 当前流行的自动化测试工具中主要使用脚本技术来生成自动化测试用例。

软件自动化测试

- 10.4 测试执行自动化

- 前处理

- 在大多数测试用例中，开始测试之间要具备一些适当的先决条件。这些先决条件应该被定义为每个测试用例的一部分并在测试之间实现。
 - 所有与建立和恢复这些测试先决条件相关的工作就是前处理，因为这是在测试工作开始之前必须进行的处理。

软件自动化测试

- 10.4 测试执行自动化

- 后处理

- 测试用例一旦执行将立即产生测试结果，其中包括测试的直接产物（当前结果）和副产品（如工具日志文件），其所涵盖范围可能很广，因此需要对这些产物进行处理，以评估测试的成败。
 - 有些测试结果可以清除，而有一些需要保留，要保留的结果应该存放到一个公共的位置以便对其进行分析或只是为了防止其被以后的测试所改变或破坏这就是所谓的后处理。

软件自动化测试

- 10.4 测试执行自动化

- 自动化测试过程

- 自动化测试过程的核心内容是执行测试用例，其本质和执行一段程序没有实质区别。
 - 由于自动化测试是执行批量的测试用例，保证自动化测试的连续性和准确性很重要，因此要进行一系列的前期准备和后期清理工作。

软件自动化测试

- 10.5 测试结果比较自动化
 - 比较是软件测试中自动化程度最高的任务，常常是从自动化中受益最多的任务。
 - 如果仅执行自动测试，而不进行自动比较，就不算是自动化测试。
 - 自动比较的内容可能是多种类型的：文本信息、专用格式化数据、屏幕输出内容、电子邮件信息、发送到硬件设备的数据或信号、通过网络发送到其他机器和进程的信息、数据库内容等。

重视软件工程 重视方法论

The End