

软件测试 Software Testing

13. 单元测试

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 10 月 21 日



陕西师范大学
SHAANXI NORMAL UNIVERSITY

Outline

- 什么是单元测试
- 单元测试的目标和任务
- 静态测试
- 驱动程序与桩程序
- 调试与评估
- 单元测试的管理
- 单元测试工具

单元测试

- 按阶段进行测试是一种基本的测试策略，单元测试是测试执行过程中的第一阶段
- 从单元测试的定义、目标、过程、技术与方法、评估等方面进行介绍和讨论

什么是单元测试

- 单元测试的定义
- 为何要进行单元测试

单元测试的定义

- 1 单元测试是对软件基本组成单元进行的测试，而且软件单元是在与程序的其他部分相隔离的情况下进行独立的测试。
- 2 单元测试的对象可以是软件设计的最小单元，一个具体函数或一个类的方法，也可以是一个功能模块、组件。

单元测试的主要内容

- 1 目标：确保模块被正确地编码。
- 2 依据：详细设计描述。
- 3 过程：经过设计、脚本开发、执行、调试和分析结果一个过程。
- 4 执行者：由程序开发人员和测试人员共同完成。
- 5 采用哪些测试方法：以白盒测试方法为主，辅以黑盒测试方法。
- 6 如何进行评估：通过所有单元测试用例，代码没有严重缺陷。

单元测试的基本过程

- 1 在详细设计阶段完成单元测试计划；
- 2 建立单元测试环境，完成测试设计和开发；
- 3 执行单元测试用例，并且详细记录测试结果；
- 4 判定测试用例是否通过；
- 5 提交《单元测试报告》。

为何要进行单元测试

- 1 软件测试的目的之一就是尽可能早地发现软件中存在的错误，从而降低软件质量成本，测试越早进行越好，单元测试就显得更重要，也是系统的功能测试的基础。

单元测试的目标和任务

- 目标：确保各单元模块被正确地编码是单元测试的主要目标，但是单元测试的目标不仅测试代码的功能性，还需要确保代码在结构上可靠且健全，能够在各种条件下给予正确的响应。
- 任务：为了实现上述目标，单元测试的主要任务包括逻辑、功能、数据和安全性等各方面的测试，具体地说，包括单元中所有独立执行路径、数据结构、接口、边界条件、容错性等测试。

单元测试的任务

- 1 单元中所有独立执行路径测试
- 2 单元局部数据结构测试
- 3 单元接口测试
- 4 单元边界条件测试
- 5 单元的各条错误处理通路测试
- 6 内存分析

静态测试

- 1 编码的标准和规范
- 2 代码评审 (Code Review)

编码的标准和规范

- 1 可靠性：按照某种标准或规范编写的代码比不这样做的代码更加可靠，软件缺陷更少。
- 2 可读性和维护性：符合设备标准和规范的代码易于阅读、理解和维护。
- 3 移植性：代码经常需要在不同的硬件上运行，或者使用不同的编译器编译，如果代码符合标准，迁移到另一个平台就会相对容易，甚至完全没有障碍。

代码书写规范示例

- 1 缩进 Tab
- 2 空格
- 3 对齐
- 4 空行
- 5 注释
- 6 代码长度
- 7 页宽
- 8 行数

代码评审

- 1 代码走查 (Walk Through)
- 2 正式会议审查 (Inspection)
- 3 走查与会议审查的对比
- 4 检查表

代码评审实践

- 一次检查大约 200 ~ 400 行代码，不宜超过 60 ~ 90 分钟；
- 合适的检查速度：每小时少于 300 ~ 500 行代码；
- 在审查前，代码作者应该对代码进行注释；
- 建立量化的目标并获得相关的指标数据，从而不断改进流程；
- 使用检查表 (checklist) 肯定能提高评审效果。

走查与会议审查的对比

项目	走查	审查
准备	通读设计和编码	应准备好需求描述文档、程序设计文档、程序的源代码清单、代码编码标准和代码缺陷检查表
形式	非正式会议	正式会议
参加人员	开发人员为主	项目组成员包括测试人员
主要技术方法	无	缺陷检查表
注意事项	限时、不要现场修改代码	限时、不要现场修改代码
生成文档	会议记录	静态分析错误报告
目标	代码标准规范，无逻辑错误	代码标准规范，无逻辑错误

代码评审的通用检查表

- 格式
- 程序语言的使用
- 数据引用错误
- 数据声明错误
- 计算错误
- 比较错误
- 入口和出口的连接

代码评审的通用检查表

- 存储器的使用
- 控制流程错误
- 子程序参数错误
- 输入输出错误
- 逻辑和性能
- 维护性和可靠性

驱动程序和桩程序

- 运行被测试单元，为了隔离单元，根据被测试单元的接口，开发相应的驱动程序 (Driver) 和桩程序 (Stub)。

驱动程序和桩程序

- 驱动程序 (Driver), 也称驱动模块, 用以模拟被测模块的上级模块, 能够调用被测模块。在测试过程中, 驱动模块接受测试数据, 调用被测模块并把相关的数据传送给被测模块。
- 桩程序 (Stub), 也称桩模块, 用以模拟被测模块工作过程中所调用的下层模块。桩模块由被测模块调用, 它们一般只进行很少的数据处理, 例如打印入口和返回, 以便于检查被测模块与其下级模块的接口。

驱动程序和桩程序

- 通过驱动程序和桩程序就可以隔离被测单元，而又能使测试继续下去。驱动程序作为入口，可以设置不同的数据参数，来完成各种测试用例。

调试与评估

- 1 调试与测试的对象及采用的方法在很大程度上相似，调试还用到断点控制等排错方法，但其目的却完全不同。
- 2 测试时为了找出软件中存在的缺陷，而调试是为了解决存在的缺陷，是程序员的职责。

通过单元测试的一般准则

- 1 软件单元功能与设计需求一致
- 2 软件单元接口与设计需求一致
- 3 能够正确处理输入和运行中的错误
- 4 在单元测试中发现的错误已经得到修改并且通过了测试
- 5 达到了相关的覆盖率的要求
- 6 完成软件单元测试报告

单元测试的管理

- 在单元测试的各个过程中，必须遵守一定的规则，以一些设计文档等为依据，并产生报告、分析文档等。
 - 1 计划阶段
 - 2 设计阶段
 - 3 执行阶段
 - 4 评估阶段
 - 5 提交阶段

单元测试的管理

- **计划阶段：**在软件详细设计阶段完成。制定单元测试计划的主要依据时《软件需求规格说明书》、《软件详细设计说明书》，同时要参考并符合软件的整体测试计划和集成方案。这一阶段完成时输出《单元测试计划》。
- 单元测试计划的主要内容包括测试时间表、资源分配使用表、测试的基本策略和方法。

单元测试的管理

- **设计阶段**：《单元测试计划》提交后进入设计阶段，设计阶段的主要任务是单元测试用例的设计编写、驱动程序和桩程序的设计及代码编制，单元测试用例是测试效率和质量的重要保证。
- 该阶段进行的主要依据时《单元测试计划》、《软件详细说明书》。测试用例完成后生成《单元测试用例》文档。

单元测试的管理

- **执行阶段**：单元测试执行依据需求定义、《程序详细设计说明书》完成单元测试用例的执行。对测试中发现的错误或缺陷进行记录，生成《缺陷跟踪报告》。将该报告反馈给开发人员，及时修改。
- **评估阶段**：包括测试完备性评估和代码覆盖率评估。进行评估的依据时《单元测试用例》、《缺陷跟踪报告》等。

单元测试的管理

- **提交阶段：**通过单元测试的评估，正式填写并提交《单元测试报告》。

单元测试工具

- 检测代码复杂度
 - 深度最高的块 To Deepest Block
 - 复杂度最高的方法 To Most Complex Method
- 开源的单元测试工具
 - C/C++ 语言单元测试工具
 - Java 语言单元测试工具
 - 其他语言单元测试工具

小结

- 什么是单元测试
- 单元测试的目标和任务
- 静态测试
- 驱动程序与桩程序
- 调试与评估
- 单元测试的管理
- 单元测试工具

小结

- 单元测试的对象是程序系统中的最小单元，模块或组件上，其目标不仅测试代码的功能性，还需确保代码在结构上可靠且健全。
- 单元测试时测试执行的开始阶段，而且与程序设计和实现有非常紧密地关系，所以单元测试一般由编程人员和测试人员共同完成，编程人员有时起了主要作用。
- 单元测试的主要任务是完成单元中所有的独立路径、数据结构、接口、边界条件、容错性等测试。

致谢

谢谢，欢迎提问！