

软件测试 Software Testing

15. 系统测试

程适

cheng@snnu.edu.cn

计算机科学学院

2016 年 11 月 4 日



陕西师范大学
SHAANXI NORMAL UNIVERSITY

Outline

- 功能测试
- 非功能测试

系统测试

- 系统测试一般由若干个不同测试组成，目的时充分运行系统，验证整个系统是否满足非功能性的质量要求，如：
 - 是否都能正常工作并完成所赋予的任务？
 - 在大量用户使用的情况下，能够经得住考验？
 - 系统出错了，能否很快恢复过来或者将故障转移出去？
 - 能否长期地、稳定地运行下去？

理解系统测试

- 系统测试应该是客户 (或用户) 能够理解的活动，常常与客户验收测试结合在一起进行。
- 一般来说，系统测试是功能性测试，而不是结构性测试，主要是因为没有高层结构表示法。

功能测试

- 功能测试可以发生在单元测试中，也可以在集成测试、系统测试中进行，软件功能是最基本的，需要在各个层次保证功能执行的正确性。
- 在单元功能测试中，其目的是保证所测试的每个独立模块的功能是正确的，主要是从输入条件和输出结果来判断是否满足程序的设计要求。

功能测试

- 在系统集成过程中和系统集成之后所进行的系统功能测试，不仅要考虑模块之间的相互作用，而且要考虑系统的应用环境，其衡量标准是实现产品规格说明书上所要求的功能，特别需要模拟用户完成从头到尾 (end-to-end, 端到端) 的测试，确保系统可以完成实现设计的功能，满足用户的真正需求。
- 功能测试主要是根据产品规格说明书，来检验被测试的系统是否满足各方面功能的使用要求。
- 对于功能测试，针对不同的应用系统，其测试内容的差异很大，但都可以归为界面、数据、操作、逻辑、接口等几个方面。

功能测试

- 程序安装、启动正常，有相应的提示框、错误提示等；
- 每项功能符合实际要求；
- 系统的界面清晰、美观；
- 菜单、按钮操作正常、灵活，能处理一些异常操作；
- 能接受正确的数据输入，对异常数据的输入可以进行提示、容错处理等；
- 数据的输出结果准确，格式清晰，可以保存和读取；

功能测试

- 功能逻辑清楚，符合使用者习惯；
- 系统的各种状态按照业务流程而变化，并保持稳定；
- 支持各种应用的环境；
- 能配合多种硬件周边设备；
- 软件升级后，能继续支持旧版本的数据；
- 与外部应用系统的接口有效。

非功能性测试

- 常常将非功能性测试（如安全性、可靠性、性能）看作是系统测试，以区别于功能测试。

非功能性测试

- 1 性能测试
- 2 压力测试 (Stress test)
- 3 容量测试
- 4 安全性测试
- 5 可靠性测试 (Reliability)
- 6 容错性测试 (Recovery test)

性能测试

- 1 系统负载
- 2 系统性能指标
- 3 性能测试的基本过程

系统负载

- 在线用户：通过浏览器访问登录 Web 应用系统后并且还没有退出该应用系统的用户。
- 虚拟用户：模拟浏览器向 Web 服务器发送请求并接收响应的一个进程或线程；
- 并发用户

系统负载

- 用户并发数量
- 思考时间
- 负载模式：就是加载的方式。如一次性建立连接、逐渐增加连接数、随机加载、峰谷交替加载等。

系统性能指标

- 1 请求响应时间：客户端浏览器向 Web 服务器提交一个请求到收到响应之间的间隔时间。
- 2 事务响应时间：事务可能由一系列请求组成，事务的响应时间就是这些请求完成处理所花费的时间。
- 3 数据吞吐量：单位时间内客户端和服务端之间网络上传输的数据量，对于 Web 服务器，数据吞吐量可以理解为单位时间内 Web 服务器成功处理的 HTTP 页面或 HTTP 请求数量。

性能测试的基本过程

- 1 确定性能测试需求，包括确定哪些性能指标要度量，以及系统会承受哪些负载。
- 2 根据测试需求，选择测试工具和开发相应的测试脚本。
- 3 建立性能测试负载模型，就是确定并发虚拟用户的数量、每次请求的数据量、思考时间、加载方式和持续加载的时间等。
- 4 执行性能测试。通过多次运行性能测试负载模型，获得系统的性能数据。
- 5 提交性能测试报告，包括性能测试方法、负载模型和实际执行的性能测试、性能测试结果及其分析等。

压力测试

- 压力测试是模拟实际应用的软硬件环境及用户使用过程的系统负荷，长时间或超大负荷地运行测试软件，来测试被测试系统的性能、可靠性、稳定性等。
- 压力测试的目的就是在软件投入使用前或软件负载达到极限前，通过执行可重复的负载测试，了解系统可靠性、系统瓶颈等，以提高软件系统的可靠性、稳定性，减少系统的宕机时间和因此带来的损失。

压力测试

- 1 测试压力估算
- 2 测试环境准备
 - 1 压力稳定性测试
 - 2 破坏性加压测试
- 3 问题的分析
- 4 累计效应

容量测试

- 1 容量测试 (Capacity test), 预先分析出反映软件系统应用特征的某项指标的极限值;
- 2 容量测试的完成标准可以定义为: 所计划的测试已全部执行, 而且达到或超出指定的系统限制时没有出现任何软件故障。

安全性测试

- 1 两种级别的安全性
- 2 测试目标
- 3 测试范围
- 4 安全性测试方法
- 5 完成标准

两种级别的安全性

- 应用程序级别的安全性
- 系统级别的安全性

- 1 应用程序级别的安全性，包括对数据或业务功能的访问；系统级别的安全性，包括对系统的登录或远程访问；
- 2 应用程序级别的安全性可确保，在预期的安全性情况下，操作者只能访问特定的功能或用例，或者只能访问有限的数据库；
- 3 系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。

测试目标

- 1 应用程序级别的安全性：核实现操作者只能访问其所属用户类型已被授权访问的那些功能或数据。
- 2 系统级别的安全性：核实只有具备系统和应用程序访问权限的操作者才能访问系统和应用程序。

测试范围

- 1 确定并列出各用户类型及其被授权访问的功能或数据。

安全性测试方法

- 1 静态的代码安全测试
- 2 动态的渗透测试
- 3 程序数据扫描

完成标准

- 1 各种已知的操作者类型都可访问相应的功能或数据，而且所有事务都按照预期的方式，并在先前的应用程序功能测试中运行了所有的事务。

可靠性测试

- 软件可靠性是软件系统在规定的时间内及规定的环境条件下，完成规定功能的能力。
 - 1 规定的时间；
 - 2 规定的环境条件；
 - 3 规定的功能。

可靠性测试

- 1 可靠性测试方法
- 2 可靠性数据收集
- 3 可靠性测试结果的评估
- 4 可靠性评估报告

容错性测试

- 1 输入异常数据或进行异常操作，以检验系统的保护性
- 2 灾难恢复性检测

容错性测试

- 1 故障转移与数据恢复
- 2 测试目标
- 3 测试范围
- 4 完成标准
- 5 需考虑的特殊事项

小结

- 功能测试
- 非功能测试

小结

■ 从质量的 3 个维度看系统测试

- 1 可靠性
- 2 功能
- 3 性能

从质量的 3 个维度看系统测试

质量维度	测试类型
可靠性	<p>完整性测试：侧重于评估测试对象的强壮性（防止失败的能力），语言、语法的技术兼容性以及资源利用率的测试。该测试针对不同的测试对象实施和执行，包括单元和已集成单元。</p> <p>结构测试：侧重于评估测试目标是否符合其设计和构造的测试。通常对基于 Web 的应用程序执行该测试，以确保所有的链接都已连接、显示正确的内容以及没有孤立的内容。</p>

从质量的 3 个维度看系统测试

质量维度	测试类型
功能	<p>配置测试：侧重于确保测试对象在不同的硬件和/或软件配置上按预期运行的测试。该测试还可以作为系统性能测试来实施。</p> <p>功能测试：侧重于核实测试对象按计划运行，提供需求的服务、方法或用例的测试。该测试针对不同的测试对象实施和执行，包括单元、已集成单元、应用程序和系统。</p> <p>安装测试：侧重于确保测试对象在不同的硬件和/或软件配置上，以及在不同的条件下（磁盘空间不足或电源中断）按预期安装的测试。该测试针对不同的应用程序和系统实施和执行。</p> <p>安全测试：侧重于确保只有预期的主体才可以访问测试对象、数据（或系统）的测试。该测试针对多种测试对象实施和执行。</p> <p>容量测试：侧重于核实测试对象对于大量数据（输入和输出或驻留在数据库内）的处理能力的测试。容量测试包括多种测试策略，如创建返回整个数据库内容的查询；或者对查询设置很多限制，以至不返回数据；或者返回每个字段中最大数据量的数据条目。</p>

从质量的 3 个维度看系统测试

质量维度	测试类型
性能	<p>基准测试：一种性能测试，该测试将比较（新的或未知的）测试对象与已知的参照负载和系统的性能。</p> <p>竞争测试：侧重于核实测试对象是否可以接受多个主体对相同资源（数据记录、内存等）请求的处理。</p> <p>负载测试：一种性能测试，用于在测试的系统保持不变的情况下，核实和评估系统在不同负载下操作极限的可接受性。测评包括负载和响应时间的特征。如果系统结合了分布式架构或负载平衡方法，将执行特殊的测试以确保分布和负载平衡方法能够正常工作。</p> <p>性能曲线：在该测试中，将监测测试对象的计时配置文件，包括执行流、数据访问、函数和系统调用，以确定并解决性能瓶颈和低效流程。</p> <p>强度测试：一种性能测试，侧重于确保系统可在遇到异常条件时按预期运行。系统面对的工作强度可以包括过大的工作量，不充足的内存，不可用的服务、硬件或过低的共享资源。</p>

致谢

谢谢，欢迎提问！