软件测试和自动化测试

软件测试概述

软件测试是一种用来促进鉴定软件的正确性、完整性、安全性和品质的过程，也就是在规定的条件下对程序进行操作以发现程序中的错误，衡量软件的品质并对其是否能满足设计要求进行评估的过程。

测试的方法

黑盒测试：测试应用程序的功能，而不是其内部结构或运作。测试者不需具备应用程序的代码、内部结构和编程语言的专门知识。测试者只需知道什么是系统应该做的事，即当键入一个特定的输入，可得到一定的输出。测试案例是依应用系统应该做的功能，照规范、规格或要求等设计。测试者选择有效输入和无效输入来验证是否正确的输出。此测试方法可适合大部分的软件测试，例如集成测试和系统测试。

白盒测试：测试应用程序的内部结构或运作，而不是测试应用程序的功能（即黑箱测试）。在白箱测试时，以编程语言的角度来设计测试案例。测试者输入数据验证数据流在程序中的流动路径，并确定适当的输出，类似测试电路中的节点。

由于时间和成本的约束，软件测试中一个最为关键的问题就是：“**在所有可能的测试用例中，哪个子集能发现最多的错误？**”。所以在设计测试用例时，白盒测试看重程序逻辑覆盖的程度（语句覆盖、条件覆盖、分支覆盖），黑盒测试可以使用等价类划分、边界值分析、因果图分析、错误猜测等方法来设计测试用例。

测试的种类（阶段）

单元测试：对软件组成单元进行测试，其目的是检验软件基本组成单位的正确性，测试的对象是软件设计的最小单位 - 函数。

集成测试：将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。其主要目的是检查软件单位之间的接口是否正确，集成测试的对象是已经经过单元测试的模块。

系统测试：系统测试主要包括功能测试、界面测试、可靠性测试、易用性测试、性能测试。

回归测试：为了检测代码修改而引入的错误所进行的测试活动。回归测试是软件维护阶段的重要工作，有研究表明，回归测试带来的耗费占软件生命周期的1/3总费用以上。

测试驱动开发（敏捷测试）

测试驱动开发包括以下三个步骤：

1. 为未实现的新功能或者改进编写自动化测试。
2. 提供通过所有定义的测试的最小代码量。
3. 重构代码以满足所需的质量标准。

测试驱动开发的好处在于可以有效的防止软件回归以及提供更有质量的代码。还有就是验收测试应该由客户来进行，客户通过对使用场景来设计验收测试，对应用程序是否满足他们的要求进行客观、公正的确认。能够通过单元测试、甚至是系统测试的功能未必能够通过客户的验收测试。

互联网应用和移动应用的测试

互联网应用的测试策略：

1. 表示层测试（内容测试、站点结构测试、用户环境（浏览器、操作系统等））
2. 业务层测试（性能、数据验证、事务、外部服务）

3. 持久层测试（响应时间、数据完整性、容错性）

移动应用的测试策略：

1. 真机测试
2. 基于模拟器的测试

单元（模块）测试

Python的标准库里有为编写单元测试而准备的unittest模块，执行测试时建议使用pytest或nose2。pytest是一款能够自动搜索并执行测试的测试执行工具，并且会输出详细的错误报告。关于单元测试可以看看《Python必会的单元测试框架 - unittest》。

可以安装testfixtures库来辅助单元测试，它整合了多种典型配置器，提供了生成目录、更改系统日期、生成mock对象的功能模块，这些模块能够帮助我们将单元测试与单元测试所依赖的环境分离开。mock是将测试对象所依赖的对象替换为虚拟对象的库，在测试的时候，我们可以为虚拟对象指定其在被调用时的返回值以及是否发生异常等。

tox能便捷地为我们准备好执行测试所需的环境。tox会在多个virtualenv环境中搭建测试 环境，然后在这些环境中执行测试并显示结果。它能够把测试工具的选项及环境变量等内容统 一起来，所以我们只需执行tox命令即能轻松完成所需的测试。

自动化测试

UI自动化测试

桌面端 - PyAutoGui

移动端 - Appnium

Web端 - Selenium

Selenium是实现Web应用程序的功能测试以及集成测试自动化的浏览器驱动测试工具群。和使用浏览器的用户相同，Selenium可以在浏览器进行的鼠标操作、在表单中输入文字、验证表单的值等，利用这一点就可以将手动操作变成自动化操作。

1. Selenium优点

- 自动化测试用例制作简单。Selenium提供了Selenium IDE工具，该工具可以捕获鼠标、键盘的操作，然后通过重放功能来重复这些操作，这样就可以简单的制作测试用例。
- 支持多种浏览器和操作系统。

2. Selenium的组件

- Selenium IDE
- Selenium Remote Control
- Selenium WebDriver

3. 与持续集成工具协作

   持续集成指的是频繁的将代码集成到主干。它的好处主要有两个：

   - 快速发现错误。每完成一点更新，就集成到主干，可以快速发现错误，定位错误也比较容易。

- 防止分支大幅偏离主干。如果不是经常集成，主干又在不断更新，会导致以后集成的难度变大，甚至难以集成。

持续集成的目的，就是让产品可以快速迭代，同时还能保持高质量。它的核心措施是代码集成到主干之前，必须通过自动化测试，只要有一个测试用例失败，就不能集成。编程大师Martin Fowler曾经说过：“持续集成并不能消除Bug，而是让它们非常容易发现和改正。”

可以在Jenkins中安装“Seleniumhq Plugin”插件，这样就可以将Selenium IDE制作的测试用例保存为HTML格式并提供给Jenkins来使用，基本步骤是：

- 在执行测试的机器上，从版本控制系统中下载测试套件和测试用例。
- 在执行测试的机器上下载Selenium Server。
- 从Jenkins的“系统管理”中选择“插件管理”来安装“Seleniumhq Plugin”。
- 在Jenkins的“系统管理”中选择“系统设置”并配置“Selenium Remote Control”下的“HTMLSuite Runner”。
- 新建测试用的Jenkins任务并进行配置，配置的内容包括：浏览器、起始URL、测试套件和测试结果输出文件。

配置完成后，就可以执行Jenkins的“立即构建”了。

除了Selenium之外，WebTest、Splinter和RobotFramework也是Web端测试的选择，其中WebTest可以对WSGI应用执行模拟请求并获取结果，基本上所有WSGI应用的测试都可以用它；Splinter是对Selenium的二次封装，使用上更加方便简单。

接口测试自动化测试

1. requests
2. HttpRunner
3. PyRestTest

其他方面的自动化测试

1. Locust
2. pythem

测试相关工具

1. PostMan
2. AB
3. JMeter
4. LoadRunner
5. Benchmark Factory
6. WAS