**测试用例的设计步骤**

# 前言

       设计[**测试**](http://lib.csdn.net/base/softwaretest)案例的时候，需要有清晰的测试思路，对要测试什么，按照什么顺序测试，覆盖哪些需求做到心中有数。测试用例编写者不仅要掌握[**软件测试**](http://lib.csdn.net/base/softwaretest)的技术和流程，而且要对被测软件的设计、功能规格说明、用户试用场景以及程序/模块的结构都有比较透彻的理解。测试用例设计一般**包括以下几个步骤：**

# 测试需求分析

从软件需求文档中，找出待测试软件/模块的需求，通过自己的分析、理解，整理成为测试需求，清楚被测试对象具有哪些功能。测试需求的特点是：包含软件需求，具有可测试性。测试需求应该在软件需求基础上进行归纳、分类或细分，方便测试用例设计。测试用例中的测试集与测试需求的关系是多对一的关系，即一个或多个测试用例集对应一个测试需求。

# 业务流程分析

软件测试，不单纯是基于功能的黑盒测试，还需要对软件的内部处理逻辑进行测试。为了不遗漏测试点，需要清楚的了解软件产品的业务流程。建议在做复杂的测试用例设计前，先画出软件的业务流程。如果设计文档中已经有业务流程设计，可以从测试角度对现有流程进行补充。如果无法从设计中得到业务流程，测试工程师应通过阅读设计文档，与开发人员交流，最终画出业务流程图。业务流程图可以帮助理解软件的处理逻辑和数据流向，从而指导测试用例的设计。

从业务流程上，应得到以下信息：

A、 主流程是什么

B、 条件备选流程是什么

C、 数据流向是什么

D、 关键的判断条件是什么

# 测试用例设计

完成了测试需求分析和软件流程分析后，开始着手设计测试用例。测试用例设计的类型包括功能测试，边界测试，异常测试，性能测试，压力测试等。在用例设计中，除了功能测试用例外，应尽量考虑边界、异常、性能的情况，以便发现更多的隐藏问题。

黑盒测试的测试用例设计方法有：等价类划分、边界值划分、因果图分析和错误猜测，白盒测试的测试用例设计方法有：语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、多重条件覆盖。在这里主要讨论黑盒测试。在设计测试用例的时候可以使用软件测试用例设计方法，结合前面的需求分析和软件流程分析进行设计：

功能测试：测试某个功能是否满足需求的定义，功能是否正确，完备。

适合的技术：由业务需求和设计说明导出的功能测试、等价类划分

边界测试：对某个功能的边界情况进行测试。

适合的技术：边界值划分

异常测试：对某些功能来说，其边界情况无法简单的了解或某些操作不完全是正确的但又是可能发生的，类似这样的情况需要书写相关的异常测试。

适合的技术：由业务需求和设计说明导出的特殊业务流程、错误猜测法、边界值分析、内部边界值测试。

性能测试：检查系统是否满足在需求中所规定达到的性能，性能主要包括了解程序的内外部性能因素。内部性能因素包括测试环境的配置，系统资源使用状况；外部因素包括响应时间，吞吐量等。

适合的技术：业务需求和设计说明导出的测试

压力测试：压力测试又称强度测试，主要是检查系统运行环境在极限情况下软件运行的能力，比如说给一个相当大的负荷或网络流量给应用软件兼容测试：测试软件产品在不同的平台，不同的工具，相同工具的不同版本下功能的兼容性。

# 测试用例评审

测试用例设计完成后，为了确认测试过程和方法是否正确，是否有遗漏的测试点，需要进行测试用例的评审。

测试用例评审一般是由测试leader安排，参加的人员包括：测试用例设计者、测试leader、项目经理、开发工程师、其它相关开发测试工程师。测试用例评审完毕，测试工程师根据评审结果，对测试用例进行修改，并记录修改日志。

# 测试用例更新完善

测试用例编写完成之后需要不断完善，软件产品新增功能或更新需求后，测试用例必须配套修改更新；在测试过程中发现设计测试用例时考虑不周，需要对测试用例进行修改完善；在软件交付使用后客户反馈的软件缺陷，而缺陷又是因测试用例存在漏洞造成，也需要对测试用例进行完善。一般小的修改完善可在原测试用例文档上修改，但文档要有更改记录。软件的版本升级更新，测试用例一般也应随之编制升级更新版本。测试用例是“活”的，在软件的生命周期中不断更新与完善。