**《软件质量与测试》课程小论文作业**

**浅谈软件测试过程管理方法**

**班 级：1302班**

**姓 名：常明明**

**学 号：U201317460**

浅谈软件测试过程管理方法

**摘 要：**随影响软件测试质量的因素除了测试技术与方法,还包括对测试过程的管理[1]。本文旨在探讨上述过程的管理内容与管理方法,通过提高过程的可视性和可控性即提高过程能力来提高软件测试的质量。

**关键词：**软件测试用例管理；缺陷管理；资源管理；

## 1引言

我国近几年来软件产品的出口正在迅速增长,国内软件产品的应用也越来越广泛,市场对软件质量重要性的认识正由此而逐渐加强[2]。通过开展软件测试,可以充分暴露软件中潜在的各种缺陷,从而提高软件产品质量。在这样的背景下,我国的软件测试行业正在从萌芽状态中逐步发展起来[3]。影响软件测试质量的因素除了测试技术与方法,还包括对测试过程的管理.下面我将谈谈对软件测试过称管理的认识。

## 2 进行软件测试管理的原因

1） 软件测试的工作量要占整个软件开发工作量的40％以上，对于高可靠、高安全的软件来说，这一比例可能会达到60％～70％。因此，软件测试是软件开发过程中的一项重要工作，必须对其进行科学有效的管理。

2） 一项软件测试工作涉及到技术、计划、质量、工具、人员等各个方面，是一项复杂的工作，因此需要对其进行管理。

3）任何软件测试工作都是在一定的约束条件下进行的，要做到完全彻底的测试是不可能。

4）只有系统化、规范化的软件测试才能有效地发现软件缺陷，才能对发现的软件缺陷实施有效的追踪和管理，才能在软件缺陷修改后进行有效的回归测试。

## 3 软件测试过程应该如何管理

### 3.1 测试用例的管理

我们都知道，测试，不管是白盒，黑盒，功能或性能测试都离不开测试用例，可以怎么说，测试用例是一切测试的基础，也是测试的核心地区。测试用例设计的好与坏，完善与不完善都直接影响到测试的效果，产品的质量保证。

下图为一个简单测试用例中心图

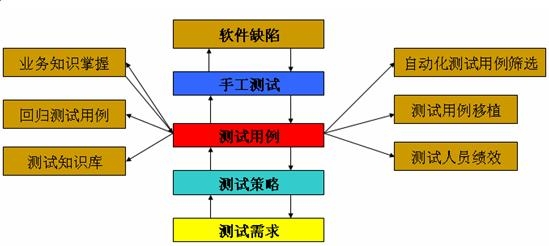


图1 测试用例中心图

上图是与测试用例为核心进行管理，下面进行解释：

1、软件测试的几个关键过程可以通过中间一列进行表示出来，一般测试人员在进行参与项目测试时，首先应该由测试负责人根据软件需求进行测试需求提起，然后通过测试需求来确定项目测试的目标和缺陷判定标准。测试策略是根据测试需求来制定详细规划，最后分发到各个编写测试用例人员手中进行测试用例编写。在进行测试用例评审过程中，可以发现测试用例为中心管理第一点好处，测试用例编写反应出测试人员对需求的理解程度。通过“需求——测试用例”，逐渐达到熟悉软件需求和用例完善。

2、再看第二点，执行测试用例发现软件缺陷，通过图中的“软件缺陷——测试用例”，也构成一个小循环，执行人员在执行测试用例时，能发现测试人员编写用例水平情况，完善程度。而测试用例也能让软件缺陷被发现越多，提供给开发人员的缺陷描述越准确。这也就是第二点好处。

3、“软件缺陷——测试需求”可以看成一个大循环，通过对需求的理解可以设计出测试用例，通过执行测试用例可以发现软件缺陷，反过来也一样，通过软件缺陷可以反应出测试用例是否完善，也能反应出需求的不完善，促进项目产品的功能越来越完善。

4、通过编写测试用例效率，执行测试用例速度情况，都能看出一个测试人员对业务知识的掌握情况，掌握越多，编写用例肯定比较完善，执行人员也能快速执行用例发现问题。通过测试用例编写与执行情况，可以促进业务知识方面进行培训，这是第四点，“业务知识——测试用例”的循环。

5、测试用例是测试人员进行的一项测试工作，也是耗时最长，需要消耗精力最多的测试工作，如何保证后续产品能快速测试并且能保证产品质量，这就需要进行回归测试，可以使用自动化测试进行，但对于没有进行自动化测试的公司来说，从测试用例中挑选一批高质量的回归测试用例，在每次新版本中，进行快速回归测试也是一种不错的做法。

6、当然即使进行自动化测试，也还是需要进行编写自动化测试用例，开始的测试用例如果编写完善，详细的话，一些用例可以直接做为自动化用例，这样也提高了测试效率，第六点。

7、而对于测试部门来说，测试知识库的积累显的至关重要，完善的知识库，不但可以让新员工快速对公司产品测试上手，测试用例库是一个最好的积累，新员工可以通过阅读用例快速掌握产品功能，业务知识，常用的测试手段，用例书写方法等。而且对一些测试技巧也能很好的提高。

对于用例管理的根本问题，我个人认为是分类上，如何有效的维护和优化用例，就是需要前期明确的分类规划，根据分类的优先级一步一步地来完成就可以了，到最后，我们也可以有效把控的测试覆盖度。

当前，我们大致可以把测试用例分称三个方面，分别是功能、UI和业务流程，从这三个角度来进行设计。

1、从功能的角度，功能是每个项目测试的重点，通常在测试人员得到需求文档的时候，我们就开始设计测试用例，那么这个时候需求文档上列出都是功能以及部分一些业务逻辑等，所以在测试用例的第一阶段就是完成功能的用例设计。不过这里，肯定会让很多人疑惑，其实功能、业务还有UI，都是有关联的，而且很多时候无法分解的。这里后面我会举个例子说明哈，但绝非都是可以分类，只是谈谈如何分解的方法，最重要的就是不要遗漏就行。

2、从UI的角度，UI通常是指界面测试，这个应该不难理解，但要想与功能点进行分解，也不是那么容易区分的，所以我们来直观的说明。界面测试，注重样式，外观、整洁、摆放以及易用性，还包括用户体验等。

3、从业务的角度，这个相对来说，还比较好理解，业务通常是指一连串的动作所连接起来的流程，这个流程必须有行为和目标，或者说方向。业务通常是一个项目或者产品设计的核心，当下，越来越多的应用业务流程都是非常复杂，所以对于业务的用例设计，就是考验一个测试人员的业务水平如何。

下面通过一个证券交易平台上的买入和撤单业务，进行具体说明：

业务说明：买入业务包括股票代码、当前价格、买入价格，买入股票数量、确定买入按钮和取消按钮；撤单业务包括选择撤单的未成交业务、撤单成功、撤单失败以及取消撤单按钮；

功能点：买入按钮、取消按钮、选择撤单、撤单按钮和取消撤单按钮等

UI界面测试：股票代码、当前价格、买入价格、买入股票数量，所有的文本框；买入成功/失败的提示框；撤单成功/失败的提示框；撤单成功/失败的业务状态等

业务测试：买入业务，从输入买入表单的数据，到提交表单，到最后买入的表单显示的位置，以及买入提交但未成交，可以撤单，完成撤单的业务，到撤单成功或者失败等，这一连串的工作组合就是一个业务流程。

其实这里就存在一个争议性的问题，对于买入和撤单，既可以作为功能点，也可以作为一个业务逻辑来设计，但从本质上来讲，功能点注重单独的操作，而业务流重的在是一个流程，还需要具体业务去甄别。功能点的设计更主要对这个买入和撤单的按钮本身进行用例设计；而业务则是需要从买入和撤单之前的输入到最后输出这样一个过程来设计。

对于测试来说，用例是基础，对于回归测试、自动化、性能等等都是根本，管理好测试用例，也就是提高测试的工作质量。

### 3.2 软件测试缺陷管理

软件测试的的缺陷管理分为10个阶段。具体如下10个表。

表一 发现阶段

|  |  |  |
| --- | --- | --- |
| 序号 | 发现阶段 | 描述 |
| 1 | 1-系统测试 | 测试环境测试发现有的缺陷 |
| 2 | 2-验收测试 | 内部验收测试发现有的缺陷，由质量部进行 |

表二 严重级别

|  |  |  |
| --- | --- | --- |
| 序号 | 级别 | 描述 |
| 1 | 0-建议 | 建议增加或修改优化操作 |
| 2 | 1-轻微 | 微小的问题，几乎不影响功能；如界面上错误字 |
| 3 | 2-中等 | 次要功能未实现 |
| 4 | 3-较高 | 功能无法完成，配置错误，程序异常错误 |
| 5 | 4-高 | 操作无法完成，数据库对象存在问题 |
| 6 | 5-紧急 | 主机功能都未实现，出现严重的问题 |

表三 缺陷状态

|  |  |  |
| --- | --- | --- |
| 序号 | 状态 | 描述 |
| 1 | New | 测试过程中新提交的缺陷 |
| 2 | Open | 新提交并指派给开发人员处理的缺陷 |
| 3 | Fixed | 开发人员已修复的缺陷，等待测试人员验证 |
| 4 | Reopen | 缺陷修改未达到目标，重新指派给开发人员处理 |
| 5 | Closed | 缺陷已修复并已经通过测试人员验证 |
| 6 | Rejected | 开发拒绝的缺陷，不需要修复或者不是缺陷 |
| 7 | Pending | 当前版本不能修复的缺陷 |
| 8 | Distract | 在上线前仍未修复，后续跟踪的缺陷 |
| 9 | ISN’T BUG | 问题不是缺陷的最终状态 |

表四 缺陷处理权限

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **权限**  **角色** | New | Open | Fixed | Reopen | Closed | Rejected | Pending | Distract |
| 测试人员 | √ | √ | × | √ | √ | √ | × | × |
| 开发人员 | × | × | √ | √ | × | √ | × | × |
| 项目经理 | × | × | √ | √ | × | √ | √ | √ |

表五 上线版本

|  |  |  |
| --- | --- | --- |
| 序号 | 上线版本 | 描述 |
| 1 | 上线时间未定 | 测试任务上线时间不确定的缺陷 |
| 2 | 各系统上线时间 | 各个项目的具体上线时间，确定到日 |

表六 缺陷类型

|  |  |  |
| --- | --- | --- |
| 序号 | 缺陷类型 | 描述 |
| 1 | 01-基础功能未实现 | 页面无法打开，或打开报错，无法正常操作 |
| 2 | 02-提交不完整 | 功能设计说明书中的功能模块部分未提交 |
| 3 | 03-功能未实现 | 指的是功能不可用或虽然功能可用、但某些方面不能像所希望的那样起到相应的作用 |
| 4 | 04-数据丢失或错误 | 指的是对应数据无记录或数据计算错误、数据约束错误等 |
| 5 | 05-操作界面 | 指的是功能可用但是界面不友好或界面信息错误；用户图形、图表界面有错误 |
| 6 | 06-接口问题 | 指的是与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表相互影响的缺陷 |
| 7 | 07-版本和环境配置 | 指的是编译、版本问题或其他支持系统问题 |
| 8 | 08-文档问题 | 指的是设计文档与需求不一致类问题 |
| 9 | 09-致命错误 | 指的是服务器崩溃、死机等系统类问题 |
| 10 | 10-性能问题 | 指的是响应时间不合理、CPU等系统资源被占用率太高 |

表七 缺陷优先级

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 优先级 | 描述 | 平均时间 |
| 1 | 1-低 | 暂时不影响继续测试；可以在方便时解决 | 3天 |
| 2 | 2-中 | 部分功能无法继续测试；需要优先解决 | 2天 |
| 3 | 3-高 | 测试暂停，无法进行；必须立即解决 | 1天 |

表八 缺陷引入阶段

|  |  |  |
| --- | --- | --- |
| 序号 | 引入阶段 | 描述 |
| 1 | 测试 | 缺陷是由测试理解或环境错误造成 |
| 2 | 开发 | 缺陷是由开发错误造成 |
| 3 | 设计 | 缺陷是由设计理解错误造成 |
| 4 | 需求 | 缺陷是由原始需求分析错误引起来的 |

表九 项目

|  |  |  |
| --- | --- | --- |
| 序号 | 项目 | 描述 |
| 1 | XXXX各个具体项目 | 具体的项目名称（如：配电GIS完善化项目） |
| 2 | XXXX各个具体项目 | 具体的项目名称 |

表十 子系统

|  |  |  |
| --- | --- | --- |
| 序号 | 子系统 | 描述 |
| 1 | XXXX子系统名称 | 子系统的项目名称（如：配电WebGIS） |
| 2 | XXXX子系统名称 |
| 3 | XXXX子系统名称 | 子系统的项目名称 |
| 4 | XXXX子系统名称 |

### 3.3 测试资源的分配管理

测试资源的分配在各个阶段是不同的：

1）在初期需要项目经理或测试组长介入进去，为测试项目提供总体方向、制定测试策略、测试计划，申请系统资源

2）在测试前期，需要一些比较资深的测试设计、开发人员，对被测软件的详细了解、测试评估、测试需求的分解，设计测试用例、开发测试脚本

3）在测试中期，主要是测试执行，要看测试自动化实现的程度，如果测试自动化程度高，人力的投入没有明显的增加;如果测试自动化程度低，测试执行的人员要求多，需要比较早的计划，保证足够的资源。

4）在测试后期，资深的测试人员可以抽出部分时间去做新项目的准备工作。

一个有效的软件测试项目管理者，在测试资源的分配上尽量做到合理，既不过于保守，浪费资源，也不过于激进，使资源的使用总是处于紧张状态，随时有“崩盘”的危险。所以，在资源分配和管理中，要做到：

1. 注意合理分配任务，明确规定每一个人在测试工作中的具体任务、职责和权限，每个组员都明确自己该做什么、怎么做、负什么责任、做好的标准是什么。做到人人心中有数，为保证和提高产品质量(或服务质量)提供基本的保证。
2. 在安排任务时，尽量考虑每个人不同的技术特长、能力、性格、工作风格等，因为资源需求的估计依赖于工作量的估计和每个工程师的能力评估。
3. 在不同的测试阶段，可以进行人员的相互调换，起到相互补充、相互督促/控制的作用。
4. 在做出最后安排决定之前，最好和每一个测试人员做一次沟通，达成共识。有良好的意识去关心组员，关注项目组员的情绪，以鼓励为主，不断激励员工，鼓舞士气，发挥每一位员工的潜力，
5. 注重团队的工作效率。人员的安排应该有一个提前量和余量(buffer，10%左右)，因为一个合格的测试人员可能需要一个较长的培训、熟悉产品特性和适应测试流程的过程。

## 4实践心得

这次实践主要难点是使用两个自动化工具QTP以及LoadRunner进行对新闻管理系统的自动化功能测试设计与执行以及性能测试设计与执行。

测试用例包括了测试准备、测试计划、测试需求、测试用例、测试执行、测试缺陷管理、测试报告几部分，其中测试准备、测试计划、测试需求三个方面在以给的新闻管理系统中都已经确定，我们在任务一中进行了测试用例设计、在任务在中进行了测试用例的执行和测试缺陷的管理和报告几部分。在本次任务中，需要的是使用自动化的软件来进行测试用例的执行和缺陷的管理和报告。使用软件来进行测试用例的执行既可以准确的进行测试也可以保证测试的速度，对软件过程的整体进度都是有很大的益处的。

在使用QTP自动化工具测试功能时，我主要负责的部分是注册、登录以及首页显示的功能。首先进行脚本的录入，输入合法的内容进行录入然后在表格中输入设计好的测试用例然后进行脚本的回放，其中每一个出错的地方虽然没有进行错误的统计但是我们可以通过自己的观察看出每一个错误在系统中的提示，从而确定功能的实现程度。例如在注册和登录中可以看到输入密码和用户名不合法时进行提示。

在使用LoadRunner进行性能测试时，在LoadRunner中新建一个脚本，然后点击运行按钮后设置相关信息比如使用什么浏览器或者是测试的网址等，确认后会自动弹出浏览器并且前往将要测试的网址，通过正常的使用即可被记录下来，然后再进行对脚本的分析即可得到相关性能的数据，这个软件也是会有一些问题出现，但是大体还是能很好得测试出软件的性能情况的。

这一次的任务中我们使用了自动化的测试工具以及性能测试工具进行执行时缩短了很多人工测试时浪费的时间，同时让我们更清楚得了解到自己所测试的系统的性能。虽然这两个工具还是有一些问题存在例如测试的过程中我们不能很清楚的知道错误的原因和问题的根本，但是我们可以通过自己进行一些错误提示来更方便地检测缺陷从而进行缺陷的管理和改进。这在整个测试过程中都是非常重要的。

同时，在使用这两个软件中出现了一些兼容问题，比如在QTP的配置过程中，QTP的版本和浏览器的版本不匹配时很容易使浏览器崩溃且不能很好的记录到每一步，同样LoadRunner也是如此，最开始出现了很多不能记录到Action的情况，但是尝试了很多次都没有成功，最后进行了一些浏览器中的修改才得以记录事件的发生情况，这些软件都需要更多的熟悉从而更好地实现功能。希望以后能更多的了解同时也在自己开发的软件中使用到这样的软件来进行自动化的测试。

## 5总结

本文从归纳软件测试质量的影响因素入手,重点分析了过程能力对软件测试质量的影响,进而提出了软件测试项目过程管理方法,并通过案例分析验证该方法的可行性和有效性。本文的逻辑框架主要是由问题分析、管理方法提出、实践总结三大部分组成。首先通过论证过程能力对测试质量的影响阐明过程管理的重要意义。其次通过分析软件测试项目的典型活动流程,提出了对软件测试起主要作用的3个关键过程的管理方法与管理重点。

## 参考文献

1. 聂长海 . 关于软件测试的几点思考[J]. 计算机科学, 2011第二期, 02
2. 蔡一博. 国内软件测试现状分析[J]. 东方企业文化,2010第三期,103
3. 郑丽娜，王威，周悦. 中国第三方软件测试发展现状分析[J]. 软件产业与工程,2012第五期,38-39