**摘要:** 软件测试作为保证软件质量和可靠性的关键技术手段正日益受到广泛的重视。本文以某区贸工局的电子政务系统项目的开发为例，讨论了软件测试中如何选用适当的方法、策略和工具等来提高测试的效率进而提高软件的质量。在该项目中，我采用了全过程测试、突出测试重点和对测试进行度量的原则和策略，主要使用了企业级的自动化测试工具WinRunner。按照全过程测试的理念，测试人员在需求获取阶段就参与到了对需求的讨论之中，并使用JUnit坚持进行单元的回归测试；利用WinRunner的自动记录功能，特别是其支持插入检查点的功能，我们加入了对数据、GUI及位图的检查，有效地完成了数据交换、安全性与访问控制和GUI等的重点测试；而在测试度量方面，通过对发现的缺陷数量的度量，以及测试数据覆盖率与BUG发现率对应表和测试阶段与BUG数量的关系图这两个图表工具的分析，成功指导了我们的测试。

2010年5月，我参与了某区贸工局（以后简称区贸工局）的电子政务系统的开发，项目历时七个月，于2011年1月正式上线。项目组成员共7人，在项目中，我担任项目经理，主要负责项目管理和质量保证等方面的工作。 区贸工局已有近十年的信息系统使用经验，在本系统开发时，该局除一套采用VB+SQL Server2000开发的二层C/S结构的核心业务管理系统外，还有多套业务系统和数据交换系统，主要有：外资审批管理系统、加工贸易电子数据交换平台、加工贸易联网监管电子数据交换系统以及电子公文交换等。上述各系统基本是相互独立的，只在数据库端实现初步的数据共享，但应用的集成性很差。 区贸工局的电子政务系统是一个基于知识管理的全新的集成的管理系统，其应用范围涉及办公自动化、审批业务管理、档案管理、数据交换、互联网站等各个方面。

该系统由门户网站、办公自动化和业务管理三个子系统构成。与原有的业务系统相比，区别主要体现在三个方面：一是全新的体系结构；二是集成性，全面集成原有的各业务系统及数据交换系统；三是以知识管理为主要特征的应用层次上的全面提升，对业务审批的全过程进行监督管理，引入审批要点对相关业务进行智能辅助审批。 为了在加工贸易相关业务领域进一步拓展业务，领导对此项目非常重视，也对项目提出了很高的质量要求。我们开发团队主要通过加强测试来保证软件的质量，并在整个开发过程中遵循了相关的测试原则和策略，包括全过程测试、突出测试重点、对测试进行度量等。   
1．全过程测试 现代的测试已经发展为一个全过程的验证和确认活动，它贯穿了整个开始生命周期的始终。测试的开发和准备必须在编码之前就开始，且在开发过程的每个阶段都保证其过程的质量。全过程测试包含两层含义：一是测试人员要充分关注开发过程，对开发过程的各种变化及时做出响应。例如开发进度的调整会引起测试进度及测试策略的调整，需求的变更会影响测试的执行等。二是测试人员要对测试的全过程进行全程的跟踪。而这主要依靠建立完善的度量与分析机制，通过对自身过程的度量，及时了解过程信息，调整测试策略来实现。 我们在需求获取阶段，测试人员就参与到了对需求的讨论之中。测试人员与开发人员及用户一起讨论需求的完善性与正确性，同时从测试性角度为需求文档提出建议。开发人员则可以通过这些建议，从一个全新的思维角度提出约束，同时需求文档的质量也得以提高。测试组结合前期对项目的把握，很容易就制定出了完善的测试计划和方案，将各阶段产品的测试方法及进度、人员安排进行了策划，大大提高了测试效率。当发生需求变更时，由于测试组密切关注开发过程，因此可以依据变更的需求及时补充和完善测试用例，使测试进度受变更的影响程度降到最低。 在代码阶段，开发人员中使用JUnit进行单元测试。坚持回归测试的理念，编写Test Case并将其组成Test Suite，每增加一个小的新功能或是对代码进行了小的修改，都立即运行一遍Test Suite，体现了全过程测试的原则。   
2．突出测试重点 根据项目特点及用户的关注程度，在集成测试和系统测试阶段，我们把测试的重点放在了数据交换、安全性与访问控制和GUI三个方面，同时结合功能测试，采用的工具主要是企业级的自动化测试工具WinRunner。 我们利用WinRunner的自动记录功能，直接使用业务系统完成标准的业务操作流程来创建测试。记录测试过程中插入检查点，这也是我们进行测试的关键部分。首先是插入数据库的检查点，主要用于检查验证数据交换以及一些审批流程及权限的控制，即除具体数据外还有一些标记位，如已发监察系统，已补交告知，业务件目前的审批状态、办理人，业务受理时所设定的业务件的审批级别、业务件最终执行的审批级别、系统按预置条件识别出的业务件审批级别等。另外，对于我们不太熟悉且没有完整的数据接口定义的外资审批管理系统，我们也借助了WinRunner来跟踪数据的变化情况，取得了很好的效果。还有就是插入了GUI和位图的检查点。对于应用程序级的安全性与访问控制的功能实现，我们使用GUI测试来考察。通过定义生成的GUI Map以及相关的标记数据，来检查系统是否正确实现了相关的访问控制。对于用户非常关注的风险级别的显示，则通过定义的位图检查点予以测试。用户指定了风险级别的标识图标，且图标需在相关业务的每一个操作页面中的固定位置出现，而这些都可以利用设置位图检查点方便地进行自动的测试。   
3．进行测试度量 在测试中，我们首先对不同功能点的测试数据覆盖率和发现问题数进行度量，以分析测试用例的充分度与BUG发现率之间的关系。我们制作了测试数据覆盖率与BUG发现率对应表，表中对应每个模块，填入功能点数、测试数据组数、测试数据覆盖率和BUG的用例发现率等数据。对类似模块进行对比发现，某一功能点上所覆盖的测试数据组越多，BUG的用例发现率越高。例如对于“数据写监察系统”和“外资审批管理系统数据读写”这两个类似的模块，其对应的功能点分别为9个和15个，采用测试数据分别为105组和87组（相应的测试数据覆盖率为11.7和5.8），对应的BUG的用例发现率则为40%和18%。因为测试是不可能无限进行下去，我们结合工作量、用例执行时间等因素进行统计分析，把我们测试数据覆盖率定在5组/功能点—10组/功能点间，对于较重要或较复杂的模块则取其上限值，使我们的测试策略的制订有了足够的依据。

我们在测试过程中对不同阶段的BUG数量也进行了度量，并分析测试执行是否充分。我们把测试分为初测、细测和回归测试三个阶段，并记录三个测试阶段发现的BUG数量，绘制测试阶段与BUG数量的关系图。其中横坐标为测试阶段，纵坐标为严重级别以上的问题数量，通过此图来对测试进行分析。初测阶段三个点取得的严重问题数依次为10、13、10，细测阶段五个点取得的严重问题数依次为20、18、20、12、5。图中的问题数量折线，从初测至细测的开始阶段呈现出曲折上升，但从细测的后半段至回归测试阶段，则呈现出稳步下降的趋势。而测算的数据也表明，在相同间隔内发现的BUG数量呈收敛状态。通过对缺陷的度量及跟踪，为测试提供了有力的指导，不仅表明了我们的测试是充分的，而且在BUG数量收敛的状态下结束细测也是恰当的。

由于采用了适当的测试方法、策略和工具，我们的测试取得了不错的效果，有力地保证了项目的质量，但也存在一些需改进的方面。一是WinRunner的使用有待加强。 WinRunner的确是非常强大的测试工具，但要充分发挥其效用也并不容易。我们在录制GUI Map File时，使用的是per Test的模式，我们就发现其功能不够强，不易描述对象的改变，且效率比较低，而Global模式就能有效地解决这些问题。二是测试环境配置及准备不够充分。完善测试数据的准备不够充分。本项目中各子系统内、外的接口复杂，业务系统流程分支也多，因此需准备的测试数据量比较大，但我们对此估计有点不足，测试数据准备不够充分。 在项目总结时，为了解决出现的问题，并进一步提高测试效率，我们决定加强测试的管理工作。其中包括了重视测试环境的搭建，使用测试用例管理系统和构建软件缺陷跟踪库等，而这也是目前我们加强知识管理的组成部分之一。另外，如何用好现有工具，充分发挥其效用也需我们在今后的实践中不断探索，不断总结。