|  |  |  |  |
| --- | --- | --- | --- |
| **深圳市国泰安教育技术股份有限公司** | **版本** | **密级** | **页数** |
| V1.2 | 机密 | 共97页 |
| 测试体系管理流程指引手册（白皮书） | | |

**测试体系工作指引手册（白皮书）**

|  |  |  |  |
| --- | --- | --- | --- |
| **作者** | 吴妍 | **编制日期** | 2015-08-26 |
| **审核** | 严乔方 | **审核日期** | 2015-09-02 |
| **批准** |  | **批准日期** |  |



国泰安教育技术股份有限公司

版权所有侵权必究

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改描述 | 作者 |
| 2012-12-30 | V1.0.1 | 初稿完成 | 杜建民 |
| 2013-1-3 | V1.0.2 | 根据评审意见进行修订 | 杜建民 |
| 2013-2-28 | V1.0.3 | 根据评审意见进行修订 | 杜建民 |
| 2015-6-29 | V1.1 | 根据评审意见进行修订 | 冉 静 |
| 2015-8-26 | V1.2.1 | 根据部门现有组织架构及工作流程进行修订和补充 | 吴妍 |
| 2015-9-2 | V1.2.2 | 根据评审意见进行修订 | 吴妍 |

目录

[1 测试体系-总体概述 8](#_Toc428967233)

[1.1 测试中心组织架构 8](#_Toc428967234)

[1.2 测试职能部门定位 8](#_Toc428967235)

[1.2.1 测试中心 8](#_Toc428967236)

[1.2.2 教育/机构测试部门 9](#_Toc428967237)

[1.2.3 顶层设计部 9](#_Toc428967238)

[1.2.4 测试技术部 9](#_Toc428967239)

[1.3 测试人员的岗位职责 10](#_Toc428967240)

[1.3.1 总经理 10](#_Toc428967241)

[1.3.2 副总经理 10](#_Toc428967242)

[1.3.3 测试总监 10](#_Toc428967243)

[1.3.4 测试副总监 10](#_Toc428967244)

[1.3.5 技术总监 11](#_Toc428967245)

[1.3.6 技术副总监 11](#_Toc428967246)

[1.3.7 部门经理 11](#_Toc428967247)

[1.3.8 部门副经理 12](#_Toc428967248)

[1.3.9 部门测试主管（PL） 12](#_Toc428967249)

[1.3.10 测试项目经理（TPM） 13](#_Toc428967250)

[1.3.11 自动化测试工程师 14](#_Toc428967251)

[1.3.12 性能测试工程师 14](#_Toc428967252)

[1.3.13 系统测试工程师(之)测试设计工程师 15](#_Toc428967253)

[1.3.14 系统测试工程师(之)测试执行工程师 15](#_Toc428967254)

[1.4 测试员工职业素质要求 15](#_Toc428967255)

[1.4.1 测试人员的基本素质要求 18](#_Toc428967256)

[1.4.2 测试管理人员的基本素质要求 19](#_Toc428967257)

[1.5 能力要求 19](#_Toc428967258)

[1.5.1 系统能力 19](#_Toc428967259)

[1.5.2 设计能力 20](#_Toc428967260)

[1.5.3 管理能力 20](#_Toc428967261)

[1.5.4 沟通能力 21](#_Toc428967262)

[2 测试体系-行为规范 21](#_Toc428967263)

[2.1 新入职员工自我介绍邮件 21](#_Toc428967264)

[2.2 新员工日报格式规范 22](#_Toc428967265)

[2.3 TM工作周报 22](#_Toc428967266)

[2.4 员工工作周报 22](#_Toc428967267)

[2.5 测试项目周报 23](#_Toc428967268)

[2.6 个人月度计划 23](#_Toc428967269)

[2.7 项目测试报告 23](#_Toc428967270)

[2.8 员工加班 24](#_Toc428967271)

[2.9 员工请假 24](#_Toc428967272)

[2.10 部门周例会机制 25](#_Toc428967273)

[2.11 测试中心招聘规范 25](#_Toc428967274)

[2.12 测试中心录用规范 26](#_Toc428967275)

[3 测试体系-制度建设 28](#_Toc428967276)

[3.1 培训管理制度 28](#_Toc428967277)

[3.2 各文档审查制度 30](#_Toc428967278)

[4 测试体系-缺陷等级划分标准 31](#_Toc428967279)

[5 测试体系-系统测试 35](#_Toc428967280)

[5.1 系统测试流程 35](#_Toc428967281)

[5.2 立项与需求阶段 35](#_Toc428967282)

[5.2.1 产品立项阶段 35](#_Toc428967283)

[5.2.2 项目立项阶段 36](#_Toc428967284)

[5.2.3 需求规格评审阶段 36](#_Toc428967285)

[5.2.4 需项目经理支持的活动 37](#_Toc428967286)

[5.2.5 需求评审技巧 37](#_Toc428967287)

[5.3 测试需求分析与计划阶段 38](#_Toc428967288)

[5.3.1 测试计划阶段 38](#_Toc428967289)

[5.3.2 测试计划重点项指导 39](#_Toc428967290)

[5.3.3 测试需求分析 40](#_Toc428967291)

[5.3.4 需项目经理支持的活动 41](#_Toc428967292)

[5.3.5 计划(报告)技巧PPT附件 41](#_Toc428967293)

[5.4 测试设计阶段 41](#_Toc428967294)

[5.4.1 设计测试用例 41](#_Toc428967295)

[5.4.2 需项目经理支持的活动 42](#_Toc428967296)

[5.4.3 测试用例编写技巧 42](#_Toc428967297)

[5.4.4 用例编写技巧PPT附件 44](#_Toc428967298)

[5.5 测试阶段 44](#_Toc428967299)

[5.5.1 转测试申请 44](#_Toc428967300)

[5.5.2 测试执行 44](#_Toc428967301)

[5.5.3 回归测试 45](#_Toc428967302)

[5.5.4 阶段性测试报告 45](#_Toc428967303)

[5.5.5 同行审查 46](#_Toc428967304)

[5.5.6 需项目经理支持的活动 47](#_Toc428967305)

[5.6 测试总结 48](#_Toc428967306)

[5.6.1 系统测试报告 48](#_Toc428967307)

[5.6.2 测试报告重点项指导 48](#_Toc428967308)

[5.6.3 需项目经理支持的活动 49](#_Toc428967309)

[5.7 测试验收 49](#_Toc428967310)

[5.8 测试归档 50](#_Toc428967311)

[5.9 测试工作总结 50](#_Toc428967312)

[5.10 缺陷跟踪表 51](#_Toc428967313)

[5.11 发布标准 51](#_Toc428967314)

[5.11.1 正式版本发布标准 52](#_Toc428967315)

[5.11.2 演示版本发布标准 52](#_Toc428967316)

[5.11.3 争议性问题 52](#_Toc428967317)

[6 测试体系-性能测试 52](#_Toc428967318)

[6.1 测试主要对象概述 52](#_Toc428967319)

[6.2 性能测试流程 53](#_Toc428967320)

[6.3 设计阶段 54](#_Toc428967321)

[6.3.1 与项目经理确认测试的功能点 54](#_Toc428967322)

[6.3.2 设计测试方案/用例 54](#_Toc428967323)

[6.4 实施阶段 55](#_Toc428967324)

[6.4.1 录制/编写测试脚本 55](#_Toc428967325)

[6.4.2 运行脚本 56](#_Toc428967326)

[6.4.3 测试结果分析 56](#_Toc428967327)

[6.5 性能测试报告 57](#_Toc428967328)

[6.6 测试归档 57](#_Toc428967329)

[6.7 测试过程所需要的专业知识 58](#_Toc428967330)

[6.7.1 测试人员具备的专业知识 58](#_Toc428967331)

[6.8 测试过程的每周四点 58](#_Toc428967332)

[7 测试体系-兼容性测试 59](#_Toc428967333)

[7.1 测试主要对象概述 59](#_Toc428967334)

[7.2 兼容性测试流程 61](#_Toc428967335)

[7.3 设计阶段 62](#_Toc428967336)

[7.3.1 确定测试的功能点 62](#_Toc428967337)

[7.3.2 设计测试方案/用例 62](#_Toc428967338)

[7.4 实施阶段 63](#_Toc428967339)

[7.5 兼容性测试报告 63](#_Toc428967340)

[7.6 测试过程所需要的专业知识 64](#_Toc428967341)

[7.7 测试过程的每周四点 64](#_Toc428967342)

[7.8 浏览器兼容性常用测试工具 64](#_Toc428967343)

[8 测试体系-安全性测试 66](#_Toc428967344)

[8.1 测试主要对象概述 66](#_Toc428967345)

[8.2 安全性测试流程 70](#_Toc428967346)

[8.3 设计阶段 71](#_Toc428967347)

[8.3.1 确定安全性测试点 71](#_Toc428967348)

[8.3.2 设计测试方案/用例 72](#_Toc428967349)

[8.3.3 设计内容评审 72](#_Toc428967350)

[8.4 实施阶段 73](#_Toc428967351)

[8.5 安全性测试报告 73](#_Toc428967352)

[8.6 测试过程所需要的专业知识 74](#_Toc428967353)

[8.7 测试过程每周的四点 75](#_Toc428967354)

[8.8 测试经验知识库 75](#_Toc428967355)

[8.8.1 过去犯错经验库 75](#_Toc428967356)

[8.8.2 案例库 76](#_Toc428967357)

[8.8.3 经典记录库 76](#_Toc428967358)

[8.8.4 窍门库 77](#_Toc428967359)

[9 测试体系-专项测试 78](#_Toc428967360)

[9.1 卸载与安装测试 78](#_Toc428967361)

[9.2 登陆操作测试 79](#_Toc428967362)

[9.3 添加操作测试 80](#_Toc428967363)

[9.4 删除操作测试 80](#_Toc428967364)

[9.5 查询操作测试 80](#_Toc428967365)

[9.6 软件加密测试 81](#_Toc428967366)

[10 测试体系-绩效考核 81](#_Toc428967367)

[11 测试体系-FAQ 86](#_Toc428967368)

[11.1.1 什么是“软件测试“？ 86](#_Toc428967369)

[11.1.2 为什么软件会有毛病？ 86](#_Toc428967370)

[11.1.3 怎么对WEB站点进行测试？ 87](#_Toc428967371)

[11.1.4 客户/服务器环境如何影响测试？ 89](#_Toc428967372)

[11.1.5 在应用软件中出现了需求中所没有的功能怎么办? 89](#_Toc428967373)

[11.1.6 如果需求一直在变怎么办？ 89](#_Toc428967374)

[11.1.7 如果时间不够，无法进行充分的测试怎么办? 90](#_Toc428967375)

[11.1.8 什么时候停止测试？ 91](#_Toc428967376)

[11.1.9 遇到软件中错误太多，以至完全无法进行测试时怎么办? 91](#_Toc428967377)

[11.1.10 在发现错误之后应当做什么? 92](#_Toc428967378)

[11.1.11 什么是测试用例？ 93](#_Toc428967379)

[11.1.12 什么是测试计划？ 93](#_Toc428967380)

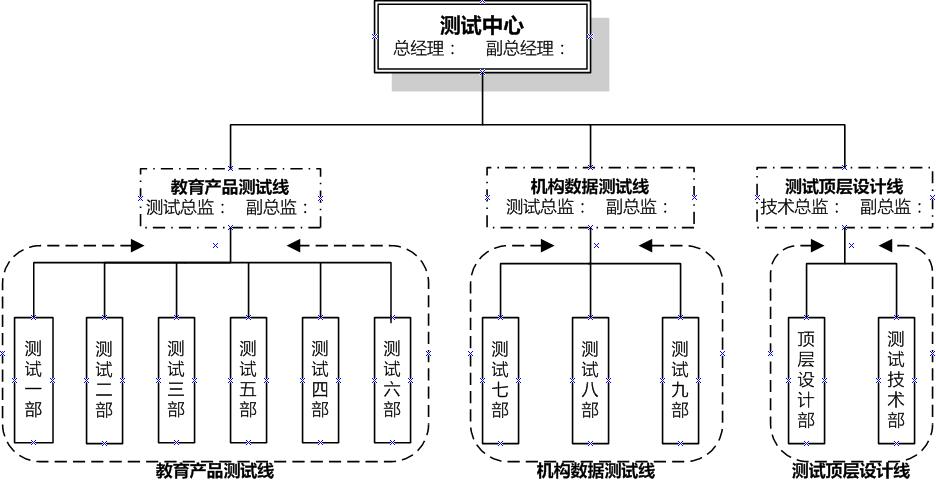
[11.1.13 软件测试包括哪些内容？ 95](#_Toc428967381)

[11.1.14 测试也有版本控制吗？ 96](#_Toc428967382)

[11.1.15 其它注意事项 96](#_Toc428967383)

# 测试体系-总体概述

## 测试中心组织架构



## 测试职能部门定位

### 1.2.1 测试中心

* 设计有效的组织架构与运作机制，制订并优化专业科学的产品测试流程、标准及规范；
* 测试团队人力资源优化整合，合理引进人才与内部提拔相结合，致立于打造一支专业化、职业化测试队伍，提升测试团队的整体专业素养；
* 负责承担公司教育线、机构数据线系列产品的系统测试，包括功能和性能测试；
* 结合公司产品业务特点，制定各业务测试组的绩效考核承诺卡，量化测试岗位员工绩效考核指标，建立以结果为导向的绩效考核体系和员工晋升机制；
* 负责培养并提升测试人员的技能，为业务线输出合格的测试项目经理及测试人员；
* 组织各业务测试部门制定产品测试月度计划，并监督计划得到贯彻执行；
* 测试分析与设计、自动化技术的研究，并使研究成果得到普及与应用；
* 协助产品经理组织产品的验收测试工作。

### 1.2.2 教育/机构测试部门

软件测试是软件开发过程中的重要组成部分，业务测试部门主要肩负着如下责任：

* 在项目前期、需求文档确立基线前对需求文档进行评审，从用户体验和测试的角度提出看法与建议；
* 针对测试需求进行相关测试技术的研究；
* 编写合理的测试计划，并与项目整体计划有机地整合在一起；
* 根据产品需求，进行全面的测试需求分析;
* 编写规范、高效、覆盖率高的测试用例；
* 认真仔细地实施测试工作，提交规范、高效的缺陷，进行缺陷跟踪与分析，协助缺陷解决，直至缺陷得到闭环处理；
* 编写测试报告，客观反馈研发过程、软件版本质量情况；
* 对测试整个过程进行总结，完善和优化测试流程，提高和改进测试方法和技术。

### 1.2.3 顶层设计部

* 提升整个测试中心的测试分析、设计能力，为业务线输出合格的测试项目经理及测试人员；
* 培养性能测试测试人员，解决各业务线性能测试测试需求；
* 组织有效的组织架构与运作机制，制订并优化专业科学的产品测试流程、标准及规范；
* 对提升测试效率的技术及工具进行研究及引用，并使研究成果得到普及与应用。
* 组织各测试技能培训，如测试需求分析、用例设计培训，TPM工作培训，性能测试培训等。

### 1.2.4 测试技术部

* 建立并规范化自动化测试流程，从需求提出，到可行性分析，到自动化实施形成一套完整、规范的操作流程。
* 培养自动化测试人员，解决各业务线自动化测试的需求，提升测试效率；
* 对提升测试效率的技术及工具进行研究及引用，并使研究成果得到普及与应用；
* 对各业务线提出的测试技术需求进行支援，解决技术难点；
* 组织自动化测试培训，及引进效率提升工具及技术培训。

## 测试人员的岗位职责

### 总经理

1. 全面负责测试中心管理工作，主管对外事务;
2. 负责测试技术部门的管理工作。

**对整个测试中心工作质量及工作效率负责**

### 副总经理

1. 协助总经理处理中心日常管理工作，如上传下达，与周边部门协调及突发性任务处理;
2. 协助总经理处理项目管理工作，如项目文档审批，项目过程及质量监管;
3. 协助总经理进行中心团队建设，如规划中心总体架构、中高级人才招聘复审；
4. 协助总经理监督分管部门的人员安排和利用情况，合理协调各部门之间的相关协助工作。

**对协助事务的成果负责**

### 测试总监

1. 负责分管部门测试项目的管理工作，如项目及人员安排的统筹管理;
2. 负责测试项目的过程监控，如掌握项目进度，协助项目风险及问题解决;

（3）负责分管部门的团队建设工作，如分管部门的人员结构规划，及初、中、高级人员招聘复审。

**对所负责的项目质量及工作效率负责**

### 测试副总监

1. 协助测试总监对分管部门进行管理工作；
2. 协助测试总监监控测试项目。

**对分管项目的质量及工作效率负责**

### 技术总监

1. 部门日常管理工作；
2. 监控部门负责的测试项目。
3. 管理测试技术部和顶层设计部，负责技术总体架构规划与顶层设计；
4. 监督测试技术部和顶层设计部门的工作开展，使专业测试技术应用和顶层设计工作开展落地各部门。
5. 负责技术部和顶层设计部团队建设工作，如人员结构规划，及初、中、高级人员招聘复审。

**对整个测试中心的技术能力和服务水平负责。**

### 技术副总监

1. 协助技术总监部门日常管理工作；
2. 协助监控部门负责的测试项目。
3. 协助技术总监管理测试技术部和顶层设计部，负责技术总体架构规划与顶层设计；
4. 协助监督测试技术部和顶层设计部门的工作开展，使专业测试技术应用和顶层设计工作开展落地各部门。

**对协助事务的成果负责。**

### 部门经理

1. 负责部门的日常管理与监督工作；
2. 负责部门的技术指导工作；
3. 负责部门的人员招聘与培养；
4. 负责部门项目进度跟近，风险处理，及资源协调；
5. 负责部门输出文档编写指导及审核工作;
6. 负责部门月度计划制定，员工月度计划复审、实施监督工作;
7. 负责部门每个项目工时及人力投入的审核工作，合理，高效的利用资源；
8. 负责部门员工绩效考核的组织与复审工作；
9. 负责部门培训需求收集及计划制定，并实施培训工作。

**对部门测试工作质量和工作效率负责**

### 部门副经理

1. 协助部门经理处理部门日常管理工作，如上传下达，与周边部门协调及突发性任务处理；
2. 协助部门经理进行人员招聘与培养；
3. 协助部门经理进行技术指导工作；
4. 兼职测试项目经理岗位职责；
5. 协助部门经理开展部门培训工作。

**对协助事务及项目的质量和工作效率负责**

### 部门测试主管（PL）

1. 每月底对业务组下月开展的项目工作进行统筹安排，规划好各项目的人力及时间，并为每个项目指定相应TPM；
2. 每月初对组内成员的月度计划进行审查，并督促其执行。
3. 组织周例会：包括周例会议题收集、发会议通知、发会议纪要；
4. 人员招聘：对本业务组现有人力资源进行评估，结合部门对本业务测试组的总体人力资源规划，协助部门经理进行人员增补招聘；
5. 业务库建设：组织组员创建并维护本组业务知识库，并编写业务培训通用计划。
6. 人员培养：积极开展本业务特色培训，为项目培养合格的TPM（测试项目经理）及业务测试人员；
7. 考核：负责组员的季度、年度考核审核；
8. 每周五汇总组内项目进展周报，对项目风险和问题进行跟踪解决。
9. 负责本组内自动化需求的提取，并与技术部门自动化测试工程师的对接工作；
10. 兼任测试项目经理工作。

**对业务测试组工作质量和工作效率负责**

### 测试项目经理（TPM）

1. 全权负责项目的测试任务分配工作；
2. 指定测试设计工程师、测试执行工程师；
3. 协调测试项目的内部沟通，监控测试的进度与质量；
4. 保持与项目经理的有效沟通，了解项目的进展，代表测试组与其他角色组进行沟通；
5. 制定测试计划、编写测试日程表，完成测试工作量预估；
6. 组织测试人员，对项目所有文档进行评审，并收集测试输出文档的评审意见，开展评审会议，输出评审报告。
7. 编写产品（项目）系统测试报告，报告客观的反馈项目过程、质量及风险;
8. 创建测试配置库，并及时将各测试活动阶段交付件入库、负责申请测试项目缺陷库；
9. 负责测试环境准备，组织测试实施；
10. 汇总测试文档，如测试需求分析、测试用例、评审意见、测试执行表单；
11. 兼任系统测试工程师的工作；
12. 每周五收集本周成员的工作输出成果，编写项目进展周报，并客观反馈项目风险。
13. 每两周对部门成员的工作周报进行评分；
14. 中、大型项目结项后，组织组员编写测试总结，开展总结会议。

**对测试项目组的工作质量和效率负责**

### 自动化测试工程师

1. 收集各业务部门自动化测试需求，输出可行性分析结果；
2. 实施自动化测试，如业务熟悉，开发代码，执行自动化测试；
3. 通过内部培养方式，指导业务部门测试工程师进行自动化测试，提升整个测试中心测试效率及自动化测试能力；
4. 协助各部门解决测试中心的技术瓶颈；
5. 引进提升测试效率的工具，并进行推广应用;
6. 每月组织测试效率提升讨论会议，对本月效率提升活动总结，并收集各业务部门的新需求。
7. 负责自动化测试及新技术培训工作。

**对项目/模块的测试效率提升负责**

### 性能测试工程师

1. 收集、评审性能测试需求；
2. 搭建性能测试环境，编写性能测试脚本；
3. 制定完善的性能测试计划及用例，输出性能测试相关文档（性能测试方案及报告）；
4. 执行测试场景，根据测试结果进行性能瓶颈分析和风险估计，并提供性能改进意见；
5. 及时向TPM反馈性能进展，协助项目组性能问题解决；
6. 通过内部培养方式，指导业务部门测试工程师进行性能测试，提升整个测试中心性能测试能力；
7. 协助各部门解决测试中心的技术瓶颈；
8. 性能测试相关技术培训，研究性能测试方法和工具；
9. 致力于性能测试领域的前瞻性研究、探索及创新。

**对项目/模块的性能结果负责。**

### 系统测试工程师(之)测试设计工程师

1. 参与项目前期需求阶段评审，输出各需求文档的评审意见;
2. 进行测试需求分析，并参与其它成员编写用例的评审工作；
3. 编写测试用例，并参与其它成员负责模块的评审工作；
4. 参加重点项目主要业务模块的测试执行；
5. 指导测试执行工程师的测试执行；
6. 编写重要项目的业务流程图，和业务测试方法文档；
7. 协助部门测试主管，对组员进行业务培训；
8. 每周五提供本周工作想关数据，如编写需求分析数、编写用例数、执行用例数、发现缺陷数、跟踪关闭缺陷数等，协助测试项目经理进行数据收集和度量。

**责任：对模块/系统的测试质量负责**

### 系统测试工程师(之)测试执行工程师

1. 参与用例设计，对其它成员编写的用例进行评审；
2. 协助测试项目经理进行环境搭建；
3. 根据测试用例，进行测试执行，认真填写测试执行表单；
4. 提交产品缺陷报告单并进行缺陷跟踪与分析，直至缺陷报告单得到闭环处理；
5. 对系统的安装部署文档、用户手册进行测试；
6. 每周五提供本周工作相关数据，如编写用例数、执行用例数、发现缺陷数、跟踪关闭缺陷数等，协助测试项目经理进行度量。

**责任：对模块/系统的测试质量负责**

## 测试员工职业素质要求

一个优秀的测试工程师要将错误尽可能的在项目测试的早期发现，在项目需求阶段，就需要深入参与，提出设计缺陷、漏洞，促使减少缺陷修复成本。测试过程中，对于一些不易重复出现的错误找到规律，帮助开发人员定位问题，能够对代码进行一定的检查，同时，测试工程师还要对各种编程语言、数据库都有一定的了解，有编程的概念。软件测试工程师需要具备以下职业素质。

* **责任心**

由于目前的软件测试行业处于初级阶段，还没有很好的量化指标对软件测试活动做出衡量。当前普遍存在将测试工程师发现的缺陷多少作为绩效考核指标，这种考核方式存在弊端，不够全面。软件测试本身是一个主观色彩很强的工作，测试程师在测试活动中需尽可能地模拟软件产品最终用户的业务流程来进行测试，但实际工作中，是不太可能做到。大多情况下，测试结果是基于测试工程师根据项目文档和自己对软件产品的理解基础上得出的。

我们需要软件测试人员有极强的责任心，确保自己的工作质量。

* **沟通能力**

测试人员需要与很多人员进行沟通，项目经理、开发人员、客户、市场人员等都是测试人员需要沟通的对象，而且在面对不同人员，你需要不同的语气、不同的态度，与开发人员交往就需要技巧了，我们需要从专业知识角度考虑，比如当我们发现的缺陷开发人员不认可的时候，我们如何从理论、实际应用以及缺陷可能引发的后果等角度去阐述缺陷，使他们认同我们的观点。所做的阐述要有理有据，而不是强词夺理，更不是争吵。从某种程度是软件测试与软件开发是相对立的，但目标是一致的：为客户提供满意的产品。

测试人员有时又需要跟客户进行交流，得与客户要谈得来，处处为客户着想，客户就是上帝，与上帝说话要和颜悦色。从非技术的角度出发，能将枯燥的、难懂的技术概念，介绍给客户，帮助他们理解软件，掌握软件的使用。并从系统的实际使用角度来考虑问题，帮助客户解决，又不触犯公司的利益。

好的测试工程师能在这两种环境中应付自如，不仅提高了自身的工作效率，也为公司赢得了客户满意度。

* **合作精神**

在软件研发过程中需要多部门协同工作，包括市场、研发、测试、运维等部门，一个高质量的软件产品从设计到发布，是团队努力劳动和智慧的结晶。软件测试工作从其工作内容来看，是极具破坏性的工作，而开发则是建设性的，从这点可以看出，如何做沟通，如何很好的协同工作，将决定我们的工作质量。所以测试人员需要具备高度的团队合作精神。

* **耐心、细心、信心**

软件测试工作中，需要我们有极大的耐心、细心、信心。很多人在不了解软件测试真正的工作内容时，会认为这项工作是枯燥的，无味的。其实不然，每个人工作需要一种成就感，这种成就感让我们时刻保持着工作的激情。当我们完成了比较高效的用例，在软件产品测试初期，发现了大量缺陷的时候，这时成就感就是油然而生。然而，这只是开始，随着测试的深入，发现缺陷越来越难，不是所有的缺陷都能容易地找出，这个时候，就需要我们耐心、细心了。

* **时时保持怀疑态度与破坏的精神，并且有缺陷预防的意识**

测试工程师要有怀疑一切的态度，不为每一次的“狡辩“而放弃自己的立场，记住，一切以数据说话！

测试人员不能总是以常规的思路来测试软件，要设计一些非常规的、相反的测试用例来不断地折磨软件产品，要破坏性地测试，并且不要停止你的怀疑。

软件测试工程师作为软件质量保证的一部分，一定要有质量防范意识，也就是要有缺陷预防意识。要不断总结，加强缺陷预防的能力，提高软件测试的工作效率。

* **要有严谨、敢于承担责任、稳重的做事风格**

思维严密，什么问题都要考虑到，当然除了做事认真仔细，也要有承担责任的勇气，在漫长的项目实施过程中，或大或小的错误在所难免，我们可以原谅错误，但不喜欢狡辩，要敢于承认错误。

* **善于自我总结、自我督促**

应该说软件测试是一种即繁琐又枯燥无味的工作，做多了你会觉得似乎一成不变，对自己的能力没有提高，这时候就需要你作自我督促,并经常作一些阶段性的总结，新的技术新的方法新的工具层出不穷，要让自己跟上技术发展的脚步，善于将新技术新方法新工具应用到测试工作当中。

* **不断学习的能力**

软件测试工作其实对软件测试工程师的技能要求很高，比如说编程语言，不一定要精通，但我们必须要了解大部分的编程语言，要具备软件开发的思想。同时，测试工程师还要掌握众多的业务知识，所以，学习能力对测试工程师来说是非常重要的。

* **抗压能力**

软件测试工作对抗压能力有一定的要求，对于项目时间紧张、任务繁重、风险及问题较多的情况下，能调整心态、不慌乱，工作抓重点，最大程度上保障项目的质量。

### 测试人员的基本素质要求

|  |  |
| --- | --- |
| **素质项** | **条件要求** |
| 基本素质 | 1. 沟通能力、自信心、幽默感、记忆力（挖掘以往错误）、耐心、怀疑精神、自我督促、洞察力（发现重点）； 2. 广泛的经验（有见解）； 3. 表达能力和问题描述能力； 4. 会提问，会寻求帮助； 5. 逻辑思维能力； 6. 愿意改变自身存在的缺点，并不断的自我提升； 7. 保持良好的心态，熟知测试职业四利原则，即：“对公司有利“、”对团队有利“、”对下属有利“、”对自己有利“； 8. 具有一定抗压能力。 |
| 专业素质 | 1. 对于系统测试，把握需求放在第一位，对产品熟练，能够快速熟练新的产品需求，要有很强的需求理解能力； 2. 测试基础：明确测试流程中各个阶段的工作，对测试识知熟悉度决定了测试流程管理的规范性，测试工作的质量； 3. 测试方案的分析设计能力、测试用例的设计能力（测试用例的覆盖率、优先级等）； 4. 测试工具的使用（包括测试管理和测试执行工具，也包括开发工具的能力）； 5. 编程能力、数据库知识、网络知识，操作系统知识。 |

### 测试管理人员的基本素质要求

|  |  |
| --- | --- |
| **素质项** | **条件要求** |
| 基本素质 | 必须具务测试人员的基本素质要求外，还需具备以下：   1. 处理日常事务的能力和处理突发事件的能力。 |
| 专业素质 | 必须具务测试人员的专业素质要求外，还需具备以下：   1. 团队协作能力，与各个小组之间的沟通能力； 2. 统帅全局，管理决定了工作质量，尤其是测试经理，需要管理协作测试的能力。一般来说，技术上的问题都不是问题，目前的软件更缺乏行之有道的管理； 3. 风险识别和处理风险的能力； 4. 对测试时间、质量和成本的控制能力； 5. 每周四点的把控能力（难点、关键点、风险点、问题点）； 6. 领导一个测试组织的能力，该组织必须坚强有力、独立自主、办事规范、没有偏见； 7. 吸引并留住杰出测试专业人才的能力。 |

## 能力要求

### 系统能力

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **岗位** | **测试经理** | **测试主管** | **测试项目经理** | **自动化测试工程师** | **性能测试工程师** | **系统测试工程师** |
| Windows | 3 | 3 | 3 | 4 | 4 | 2 |
| Linux | 3 | 2 | 2 | 3 | 3 | 1 |
| Oracle | 3 | 3 | 3 | 3 | 3 | 2 |
| MySQL | 3 | 3 | 3 | 3 | 3 | 2 |
| SQL Server | 3 | 3 | 3 | 3 | 3 | 2 |
| DevSuite | 3 | 3 | 3 | 3 | 3 | 3 |
| QTP或其他自动化工具/框架 | 3 | 2 | 2 | 4 | 3 | 0 |
| LoadRunner | 3 | 3 | 2 | 3 | 4 | 1 |

系统能力级别说明：

1级：对系统的特征、概念有一些基本的了解，并了解和其他系统的基本区别，并使用过系统；

2级：：能使用系统的基本命令，熟悉系统的简单安装配置。了解其他系统的区别；

3级：能使用系统的复杂命令，熟悉系统的复杂安装配置，能对系统进行日常维护；

4级：对系统出现的任何问题能迅速解决；

5级：对系统内核有深刻的理解。

### 设计能力

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **岗位** | **测试经理** | **测试主管** | **测试项目经理** | **自动化测试工程师** | **性能测试工程师** | **系统测试工程师** |
| Office | 4 | 3 | 3 | 3 | 3 | 2 |
| Visio | 3 | 3 | 3 | 3 | 3 | 2 |
| Mindmanager | 4 | 3 | 3 | 3 | 3 | 3 |
| MSProject | 3 | 3 | 3 | 3 | 3 | 1 |
| 测试相关文档 (计划、报告等) | 4 | 4 | 4 | 3 | 3 | 2 |

设计能力说明：

1级：有所了解

2级：熟悉

3级：熟练

4级：有创意的发挥

### 管理能力

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **岗位** | **测试经理** | **测试主管** | **测试项目经理** | **自动化测试工程师** | **性能测试工程师** | **系统测试工程师** |
| 测试项目进展周报 | 4 | 3 | 3 | 3 | 3 | 2 |
| 测试项目估计 | 4 | 3 | 3 | 3 | 3 | 1 |
| 测试项目计划 | 4 | 3 | 3 | 3 | 3 | 2 |
| 测试项目风险管理 | 4 | 3 | 3 | 3 | 3 | 1 |
| 测试项目进度控制 | 4 | 3 | 3 | 3 | 3 | 1 |
| 测试项目数据度量 | 4 | 3 | 3 | 3 | 3 | 2 |

管理能力说明：

1级：有所了解

2级：熟悉

3级：熟练

4级：有创意的发挥

### 沟通能力

（1）对事不对人，沟通过程做到有理有节，以数据说话；

（2）鼓励面对面沟通，避免信息传递产生误解；

（3）沟通过程学会换位思考；

（4）忌讳教条式思维，在大原则下学会中庸之道；

（5）针对存在争议性问题提交团队决策上高明之举；

（6）对重要沟通结论以邮件方式发送给相关人员，做到结果互通。

# 测试体系-行为规范

## 新入职员工自我介绍邮件

**邮件标题格式：**【新人报到】测试X部-姓名

**邮件正文内容：**包括但不限于：姓名，出生年月，所学专业，毕业时间，工作经历，入职部门，入职岗位，培养导师，入职感想，兴趣爱好，其他可进行自由发挥。

**邮件主送对象：c2-测试中心**

**邮件抄送对象：测试中心管理层**

**备注：**新员工入职当天，由部门经理或指定人员进行要求说明

## 新员工日报格式规范

**邮件标题格式：**【新人日报】测试X部-姓名工作日报（YYYYMMDD），请审阅！

**邮件正文内容：**包括但不限于：当天工作事项、任务描述、花费工时，成果量化；次日计划事项、预期目标、工作开展难点。按公司提供的模板填写，将内容贴到邮件正文中。模板获取路径：http://svn-e.gtadata.com:8080/svn/Test\_Department/02 公共知识/01 流程与规范/日报、周报模板

**邮件主送对象：**培养导师（必选）

**邮件抄送对象：**部门经理/副经理（必选），测试中心总经理（必选）、项目经理（必要时可选）

**备注：**新员工入职当天，由部门经理或指定人员进行要求说明。导师要对新人的日报每日进行跟进，对下个工作日开展难点进行解决。

## TM工作周报

**邮件标题格式：**【工作周报】测试中心-测试六部\_XXX工作周报（YYYY.MM.DD），请审阅！

**邮件正文内容：**按公司模板填写，将内容贴到邮件正文中

**邮件主送对象：**g-测试经理群

**邮件抄送对象：** g-研发部经理群

**备注：**同时上传至http://svn-e.gtadata.com:8080/svn/Test\_Department/04 工作周报对应部门目录中。所有新同事请参考历史的模板进行编写

## 员工工作周报

**标题格式：**国泰安个人周报\_YYYMMDD\_XXX

**备注：以周为单位按公司要求模板填写，**周五下班前上传至hhttp://svn-e.gtadata.com:8080/svn/Test\_Department/04 工作周报 测试项目对应部门目录中。所有新同事请参考历史的模板进行编写

## 测试项目周报

**邮件标题格式：**【测试X部】YYYY年第XX周测试项目周报(YYYMMDD)，请审阅！

**邮件正文内容：**包括但不限于：项目当前进度，主要问题与风险，本周工作亮点

**邮件主送对象：**本部门全体成员，[如:测试一部邮件群为c206-测试一部](mailto:如测试一部邮件群为test-t1@gtadata.com)；

g-测试经理群；对应项目经理。

**邮件抄送对象：**测试中心总经理，助总

备注：按模板以周为单位填写，每周五16：00之前通过邮件发出，同时上传至http://svn-e.gtadata.com:8080/svn/Test\_Department/04 工作周报对应部门的目录中。所有新同事请参考历史的模板进行编写

## 个人月度计划

**标题格式：**测试中心本月工作完率率及下月工作规划X月\_XXX

**备注**：按模板以月为单位 在每月第一周周四下班前，编写完个人工作月度计划，同时上传http://svn-e.gtadata.com:8080/svn/Test\_Department/10 月度计划对应的部门目录中。模板请参考SVN上的历史数据

## 项目测试报告

**报告审批环节（所有测试报告统一由TPM编写，部门经理审批，中心管理层批准）**

**邮件标题格式：**【项目测试报告】XXX项目测试报告，请审核/审批！

**邮件正文内容：**将报告中的结论章节贴到邮件正文中

**邮件主送对象：**部门经理或指定人员

**邮件抄送对象：**测试中心总经理、副总经理

**报告发布环节**

**邮件标题格式：**【项目测试报告】XXX项目测试报告，测试结论（通过/不通过），请审阅！

**邮件正文内容：**将报告中的结论章节贴到邮件正文中

**邮件主送对象：**产品经理、项目经理、QA

**邮件抄送对象：**部门经理，测试中心总经理，副总经理

## 员工加班

**邮件标题格式：**【加班申请】测试X部-姓名XX项目加班申请，请审批！

**邮件正文内容：**加班日期，加班时间，加班事由

**邮件主送对象：**TPM，项目经理

**邮件抄送对象：**部门经理

**备注：**各四级部门员工加班必须是部门经理根据项目经理意见及项目需要进行指令性加班，同时汇报到测试中心总经理处进行报备，没有事前报备的加班申请将一律视为员工个人原因导致加班行为，将不作审批驳回处理。同时加班必须按公司要求的时间在岗并打考勤卡。

## 员工请假

**邮件标题格式：**【请假申请】测试X部-姓名请假申请，请审批！

**邮件正文内容：**请假日期，请假时间，请假事由，工作委托。

**邮件主送对象：**项目经理，TPM

**邮件抄送对象：**部门经理、测试中心管理层

**备注：1.**请假按1h为单位核计，原则上所有进项目组的同事请假需求项目经理审批通过才可提交OA至部门经理审批，部门不批准生理假，情况特殊可以作为病假处理。

**2.**如未进入项目组的测试人员，需找导师请假，并通过才可提交OA至部门经理审批。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试X部（张三）2015.4.27至2015.5.9请假期间工作委托 | | | | |
| **委托人姓名** | **所在部门** | **职  位** | **联系方式** | **EMAIL** |
| 张三 | 测试五部 | 部门经理 | 15852362541 | [XXX.gtafe.com](mailto:jian.jiang@gtafe.com) |
| **被委托人姓名** | **所在部门** | **被委托工作内容** | **联系方式** | **EMAIL** |
| 李四 | 测试线 | 智慧校园及部门事助处理 | 13528758694 | [YYY.gtafe.com](mailto:jian.jiang@gtafe.com) |
| 王五 | 教育产品测试一部 | 1、E线研发例会：每周一  3、测试线例会 | 13420947108 | [zzz.gtafe.com](mailto:jian.jiang@gtafe.com) |
| **邮 件 委 托** | 是 | 工作委托期间请将邮件主送给被委托人，抄送给委托人 | | |
| **工作委托时限** | 2015.4.27至2015.5.9 | | | |
| **备       注** | 如有急事可手机联系！ | | | |

## 部门周例会机制

**邮件标题格式：**【会议通知】测试X部X周例会通知

**邮件正文内容：**

【会议主题】测试中心X 部周例会

【会议时间】2014/10/15　XX:XX-XX:XX（am）pm

【会议地点】XX

【与会人员】XX

【本周主持】XX

【会议记录】XX

【会议议题】

议题1：上周遗留问题跟近；

议题2：本周项目进展及风险；

议题3：个人纠结问题反馈及讨论；

**邮件主送对象：**测试团队，项目经理

**邮件抄送对象：c206-测试XX部,**部门经理， 助总，测试中心总经理

**备注：对于部门异地人员，部门经理每周发起一个视频会议或YY会议，做好纠结问题讨论及解决。**

## 测试中心招聘规范

**邮件标题格式：**【简历下载邀约】测试X部-累计X份简历，请协助下载并邀约，谢谢！

**邮件正文内容：中国人才热线 –ID-岗位、前程无忧–ID-岗位、智联招聘–ID-岗位**

|  |  |  |  |
| --- | --- | --- | --- |
| 搜索日期 | 简历提供商 | ID | 岗位 |
| 20150504 | 中国人才热线 | JR028036218R90250001000 | 副经理 |
| 20150504 | 中国人才热线 | JR028036218R90250001000 | 高级测试工程师 |

**说明：简历筛选原则：**

1. 近两周简历被更新；
2. 本科及本科以上；

**邮件主送对象：**CSBP(人事专员)比如黄韧琪,部门经理

**邮件抄送对象：**测试中心管理层（可选）

面试前准备：

1. 面试官在收到部门招聘接口人当前面试通知邮件，了解应聘人员简历，收到前台通知后打印应聘人员简历并前往面谈；

面试流程：

1. 初级工程师：第一面试人为业务组PL，第二面试人为副经理，第三面试人为部门经理；
2. 中级工程师：第一面试人为业务组及部经理，第二面试人为测试中心副总经理，第三面试人为测试中心总经理；
3. 高级工程师：第一面试人部门经理，第二面试人为测试中心总经理，第三面试人为研发中心总经理群面。

面试评价

1. 每一面试环节，面试官都要写上面试评语及建议，并将面试材料交给下一面面试官；

面试结果记录：

1. 部门招聘接口人跟进应聘结果，并在SVN配置库上进行登记与备案http://svn-e.gtadata.com:8080/svn/Test\_Department/01 团队建设/01 XXX部 下的/02 人员招聘；
2. 面试通过人员应聘资料交给部门负责人进行是否录用评估。

应聘人员企业文化感想

1. 部门助收到应聘人员的企业文化感想后，发送部门经理，测试中心管理

## 测试中心录用规范

**邮件标题格式：**【区域申报录用】测试X部-XXX录用申请，请领导审批

**邮件正文内容：**

|  |  |  |
| --- | --- | --- |
| 一．**姓名**，通过面试评估，符合目前深圳测试X部的XX测试工程师的要求，现拟录用，以下是对其评价与薪资建议： | | |
| **初试评价** | 两年以上专职性能测试工程师经验，熟悉http，有调优经验，态度诚实，熟悉性 | 初试官：XX |
| **复试评价:** | 对http协议的性能测试较为精通，测试经验丰富，对Linux系统，oracle数据库比较熟悉人；工作态度较好，学习能力较强 | 复试官：XX |
| **复试评价:** | 与岗位职责吻合匹配，沟通良好，态度诚恳，给人感觉踏实可靠，专业技术优秀，有过CS架构测试经验，对http接口测试理解精通，可提升教育线测试效率 | 复试官：XX |
| **姓名：薪资建议** | | |
| **学历/专业** | 本科-深圳大学/计算机科学与技术 | |
| **规属部门** | 测试资源部 | |
| **段位** | T５ | |
| **薪酬待遇** | XXX（试用期80%　）~XXX（转正） | |
| **入职导师** | XXX | |
| **坐位号** | 19栋3楼3FC-69 | |

备注说明:

1.深圳总部部门经理拟录用发放抄送：[黄韧琪renqi.huang@gtafe.com](mailto:黄韧琪renqi.huang@gtafe.com)；（只接收纸制的）; 对于高级测试工程师，部门经理级别需要附加文化感想。薪酬超过8K需要邢总审批，第一学历非一本，或超过1万的需要陈老师审批。

2.合肥部门经理拟录用发放抄送对象：[范紫燕ziyan.fan@gtafe.com](mailto:范紫燕ziyan.fan@gtafe.com)；[孟昕澜xinlan.meng@gtafe.com](mailto:孟昕澜xinlan.meng@gtafe.com);杨慧hui.yang3@gtafe.com；对于高级测试工程师、部门经理级别需要附加文化感想；薪酬超过7K要抄送给陈老师审批。

3.长沙部门经理拟录用发放抄送对象：[罗银yin.luo@gtafe.com](mailto:罗银yin.luo@gtafe.com); [柳文wen.liu@gtafe.com](mailto:柳文wen.liu@gtafe.com)；对于高级测试工程师、部门经理级别需要附加文化感想；薪酬超过8K要抄给陈老师审批

# 测试体系-制度建设

## 培训管理制度

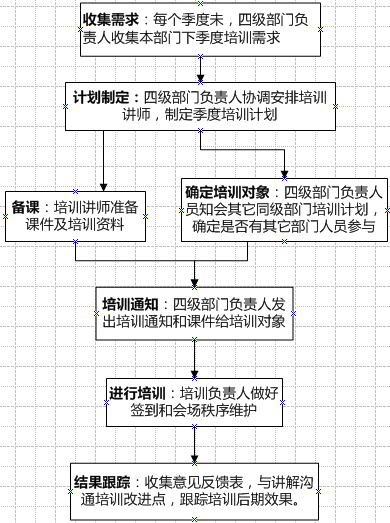
**目的：**为了规范和促进测试部门员工培训工作，提升技术研发人员的职业技能和素质，提高公司的测试技能水平，从而提高产品质量，特制定本制度。

**适用范围：**本制度适用于公司研发中心体系测试部门的所有员工。

**培训类别与内容：**研发中心体系的培训类别包括新员工培训、岗位技能培训、岗位业务培训等，具体如下表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **培训类别** | **培训对象** | **培训目的** | **培训内容** | **培训责任人** | **培训方式** |
| 新员工培训 | 新员工 | 为帮助新员工尽快了解和熟悉公司，尽快融入公司环境和进入工作角色 | (1)公司概况、企业文化组织结构、管理层人员  (2)部门人员及环境熟悉  (3)公司各项规章制度 | 入职导师 | 授课、参观、操作指导 |
| 岗位技能培训 | 所有在岗员工 | 为增强员工技能，提高工作质量和效率，减少工作失误 | 包含但不限于  (1)流程工作流程  (2)测试设计  (3)性能测试  (4)自动化测试  (5)兼容性测试 | 专项培训讲师 | 授课、操作指导 |
| 岗位业务培训 | 业务组内员工 | 为满足业务性很较的项目需要，在业务组内，安排业务培训 | (1)项目背景及专业术语  (2)项目业务流程  (3)项目测试方法和重点 | 业务组内资源测试人员 | 项目指导 |
| 转岗培训 | 岗位调动人员 | 为工作轮换、横向调整和晋升做准备 | (1)新岗位基本情况  (2)新岗位技能培训 | 新岗位部门负责人 | 操作指导 |
| 部门内部培训 | 部门内部员工 | 为满足实际工作需要，对员工进行小规模、灵活实用的培训 | 部门内各项工作、流程培训 | 部门经理/主管 | 操作指导 |
| 员工自我培训 | 员工自身 | 提高自身素质和业务能力 | 员工自身相关知识、技能 | 员工自身 | 自我培训 |

**培训实施流程：**



**培训资源管理：**

培训讲师档案管理：技术研发人员培训讲师分为内部讲师和外部讲师，由秘书处建立培训讲师档案。

（1）公司内部讲师管理：研发中心体系内部讲师为技术研发高级工程师，各级主管人员负有培训下属员工的义务和责任；

（2）外部讲师管理：外部讲师是公司聘请的授课讲师，其课酬根据实际情况和培训预算确定，公司通过培训效果的评估决定是否继续聘请该讲师。

培训教材管理：培训教材包括内部教材和外部教材，教材的形式可以是书面文字、电子文档、录音、录像等形式。

内部培训教材统一需在培训开展后，上传至SVN基线：

http://svn-e.gtadata.com:8080/svn/Test\_Department/02 公共知识/07 培训。教材内容可包含：

1. 工作过程中的经验分享与教训总结。
2. 公司重大事件案件。
3. 培训师组织开发培训教材。
4. 人力资源部经理与培训专员组织开发培训教材。

外部培训教材，引入后由秘书处管理：

1. 公司聘请外部机构进行培训的，外部机构应提供教材，教材由培训专员统一归档管理。
2. 公司员工参加外派公开课程的，应在培训结束一周内将教材的原件或复印件交由培训专员存档管理。

培训设施设备管理：培训设施设备的建设、购置、维护和管理由公司人力资源部负责。

## 各文档审查制度

|  |  |  |
| --- | --- | --- |
| **制度名称** | **制度规则** | **责任人** |
| 个人工作周报检查评分 | 时间规则：每两周对部门人员周报进行检查并评分 评分规则及模板：详细参照测试中心周报评分模板 | TPM轮流检查 |
| 个人工作月报检查评分 | 时间规则：每月初对上月总结和本月计划进行检查并评分 评分规则及模板：详细参照测试中心月报评分模板 | PL轮流检查 |
| 项目流程符合度检查评分 | 时间规则：每两周部门在研项目流程符合度进行检查 评分规则及模块：详细参照测试中心项目流程符合度评分模板 | TPM轮流检查 |
| 测试计划、系统报告评审 | 规则：每个项目计划、报告由PL进行评审（PL也可指定有经验TPM），评审通过后再发给TM进行评审，最终由部门总经理或副总经理批准。 | TPM+PL+TM |
| 性能测试方案、报告评审 | 规则：每个项目性能测试方案、测试报告由测试技术部经理（可指定有经验的成员）进行评审，评审通过后再发给TM进行评审，最终由部门总经理或副总经理批准。 | PM+PL+TM |
| 测试需求分析、用例评审 | 规则：每个项目需求分析、用例需要邀请PL进行内部评审（PL也可指定有经验TPM），内部评审通过后再发出项目正式评审，TM对各项目进行抽查，抽查结果作为TPM和PL的考核参考条件。 | TPM+PL |

# 测试体系-缺陷等级划分标准

* Bug等级划分：

致命、严重、一般、轻微、建议性

* 判断准则说明：

（1）作为测试人员，只关注其严重性，对客户带来的影响，以及给公司形象、利益等

带来的影响

（2） 在open Bug时，划分等级不考虑项目其他因素，例如项目进度，代码修改风险等，这些因素在后续Bug跟踪管理中讨论。

* Bug等级说明：

致命：死机，数据丢失，主要功能完全丧失，系统悬挂等错误

严重：，导致严重的问题，或致命的错误声明

一般：次要功能丧失，主要功能丧失不太严重，如提示信息不太准确

轻微：微小的问题， 给用户来带益处或者良好体验的建议

建议性：很微小的问题，对功能几乎没有影响，产品及属性仍可使用

* Bug等级描述举例 ：

（1）致命，描述举例：

* + - 软件无法安装，软件无法运行。
      * 基本功能无法实现或遗漏，如某一应用程序启动不了或关键功能无法运行，关键数据错失较多。
      * 功能设计与需求严重不符。
      * 系统崩溃，如应用程序死掉、应用程序异常退出、通讯意外中断或系统进入死循环，死机，冻结。
      * 造成用户数据丢失或破坏。
      * 用户说明书、操作手册，版本信息等直接影响客户或者造成客户不满，影响公司、产品形象的问题。
      * 版权信息等设计到法律方面，对公司、产品声誉造成影响或者带来法律纠纷的问题。
      * 内存使用错误造成系统崩溃等严重问题，如内存泄漏、内存溢出、数组越界等。
      * 导致无法测试的错误（Block testing），造成了项目进度的延误（即使非软件研发，是其他硬件或者用户、产品造成的问题，也要作为“灾难”，具体后续跟进处理由会议商议决定）。
      * 严重花屏。
      * 严重性能问题，如操作实时失败、[数据库](javascript:;)读写效率低。
      * 升级脚本错误，使升级失败。
      * 进程资源不能释放。

（2） 严重，描述举例：

* 部份功能未实现（非基本功能），软件按关闭出现异常。
* 次要功能实现错误（非基本功能）。
* 基本功能在某种操作方法下无法实现，但是其他方法可以实现（有workaround）。
* 影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统稳定性。
* 系统刷新错误。
* 功能设计与需求略微不符。
* 系统所提供的功能或服务受明显的影响，但未造成用户数据丢失或破坏。
* 程序抛出异常信息没有处理，如空指针、通讯异常等。
* 较严重性能问题，如无法达到市场普遍使用情况下的性能需求。
* 内存使用错误未造成系统崩溃等严重问题，如内存泄漏、内存溢出、数组越界等。
* GUI长时间僵死没有响应。
* 致命的错误声明。
* 安全性问题，例如允许非法用户对系统进行操作。
* 查询结果与查询条件不符。
* 数据窗口的TAB顺序混乱。
* 与其他常规软件不兼容，造成异常。
* 卸载异常。

（3） 一般，描述举例：

* 次要功能存在部分问题。
* 界面存在明显缺陷，设计不友好、不完善，使得某些用户无所适从。
* 安装时的小问题 ，或者安装后文件不全、文件错误造成次要功能无法实现数据的单位不一致；
* 删除操作未给提示，有可能给用户带来风险。
* 系统操作不方便。
* 性能方面达不到市场满载需求，需待改进，例如高峰期速度明显变慢。
* 边界条件下错误。
* 提示信息错误（包括未给出信息、信息提示错误等）。
* 长时间操作或等待，未给用户提示（注意区别性能方面的“严重”类Bug）。
* 光标跳转设置不好，鼠标（光标）定位错误。
* 窗口的最小化、还原、最大化功能不能实现。
* 数据窗口的TAB冗余。

（4）轻微，描述举例：

* 界面格式等不规范
* 辅助说明描述不清楚
* 操作时未给用户提示；
* 可输入区域和只读区域没有明显的区分标志
* 个别不影响产品理解的错别字
* 文字排列不整齐等一些小问题
* 表达不清楚或令人困惑的消息框、弹出的窗口
* 整个系统中控件的名称或快捷键不统一
* 按键与描述、提示不符
* 报表的表格线未对齐或预览、打印的内部不全
* 数据窗口中的数据显示不完整 （根据项目性质，可能分类不一样，数据类测试，需要提高等级）
* 数据格式不统一（根据项目性质，可能分类不一样，数据类测试，需要提高等级）
* 数据窗口的TAB

（5）建议性，描述举例：

* 一些不影响现有需求的建议
* 美观方面的建议，如：滚动条问题（当页面包括子页面时，不应该出现双滚动条的现象）

# 测试体系-系统测试

## 系统测试流程



## 立项与需求阶段

### 产品立项阶段

由产品经理组织召开产品立项评审工作及会议，并确定项目经理及测试项目经理。

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 《可行性分析报告》、《立项报告》 |
| 工作内容及目标 | 1. 评审《可行性分析报告》、《立项报告》，确定项目是否立项开发。 2. 跟进评审意见得到闭环处理 |
| 退出标准 | 确定测试项目经理 |
| 责任人 | 测试经理（指定测试项目经理） |
| 测试团队产出物 | 评审意见 |

### 项目立项阶段

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 《客户需求列表》、《客户需求说明书》、快速原型（可选） |
| 工作内容及目标 | 1. 了解项目背景，识别项目干系人。 2. 明确客户测试需求，提出可测试性需求。 3. 与产品经理、项目经理及QA共同确认质量目标，特别是性能质量标准。 4. 要求产品经理对线下评审意见给出合理的解释或答疑。 5. 跟踪评审意见得到闭环处理。 6. 给出初步人力数测试执行周期 7. 建立测试库，对项目文档进行基线 8. 给测试成员申请项目库存权限 |
| 退出标准 | 需求明确，无测试盲点 |
| 责任人 | 测试项目经理 |
| 测试团队产出物 | 评审意见、测试库建立、项目库权限申请 |

### 需求规格评审阶段

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 需求规格说明书 |
| 工作内容 | 1. 测试团队成员提出需求规格说明书中不清楚、不完整及存在异议之处，编写人员解答并修改。 2. 跟进评审意见得到闭环处理 |
| 退出标准 | 1. 所有人员对需求无异议 |
| 责任人 | 1、评审意见 |

评审意见事项：

1、需求定义基本完成，此时应在评审会议召开之前发给测试团队，预留时间给测试相关人员熟悉、理解；

2、部门经理和测试项目经理一起，指定测试代表，对需求规格进行评审工作，把关需求，避免出现需求不明确和测试盲点。

### 需项目经理支持的活动

1. 和TPM一起向测试经理申请测试资源。
2. 统一为测试人员申报项目相关权限，如SVN、项目管理工具
3. 评审测试TPM提交的测试工作量预估，在此基础上来统一评估项目周期。

### 需求评审技巧

**重点关注项：**

1. 是否存在需求不明确、需求不可测、需求不可实现，这些往往导致后续工作难于开展或经常出现变更；
2. 目标性需求明确：定义了整个系统需要达到的目标；
3. 功能性需求完整：定义了整个系统必须完成的任务；
4. 非功能性需求完整：定义了性能需求，兼容性需求，稳定性需求；
5. 操作性需求具体：定义了完成每个任务的具体的人机交互。

**过程注意事项：**

1. 做好评审前的沟通和准备：评审所需的资料准备齐全，需预留足够时间进行预评审；
2. 先沟通好目标，再进行细节的落实：当需求量大时，需在需求形成过程中分阶段进行评审，而不是需求最终形成后再进行评审。保证了需求形成过程中不偏离方向，不出现大错误，同时降低了需求返工的风险，提 高了最终评审质量。
3. 预 评审和正式评审相结合：在进行正式评审之前，需提前做好预评审，对疑问和问题进行批注，带着问题来参加正式评审，而不是参加培训或临时查看文档，以提高正式评审的效率。
4. 做好评审后的跟踪工作，做到问题闭环处理：切忌评审完毕后，没有对问题进行跟踪，而无法保证评审结果的落实，使前期的评审努力付之东流。评审意见修改后，对评审意见的处理方式和最终的文档，需发送给评审意见提出方进行确认。

## 测试需求分析与计划阶段

### 测试计划阶段

测试需求分析文档和项目开发计划文档完成后，测试组需要编写测试计划文档、制定测试策略及预估测试过程中的风险，并设计出合理的规避风险的策略，为后续的测试工作提供直接的指导。

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 客户需求说明书，需求规格说明书，项目开发计划，项目配置表 |
| 工作内容 | 1. 依据项目背景及要求，确定测试环境。 2. 测试需求——需要测试组测试的范围，各个测试需求的测试优先级。 3. 测试策略——确定项目的测试计划内容，整体测试的测试方法和每个测试需求的测试方法，同时做好测试进度安排及人员调整。 4. 测试资源——估算出测试所花费的人力资源，本次测试所需要用到的人力、硬件、软件、技术等资源。 5. 测试组角色——明确测试组内各个成员的角色和相关责任。 6. 可交付工件——在测试组的工作中必须向项目组提交的产物，包括测试计划、测试报告等。 7. 风险管理——列举出测试工作所可能出现的风险。 8. 产品性能指标。 9. 测试计划编写完毕后，必须提交给项目组全体成员，并由项目组组中各个角色组联合评审。 10. QA组织项目成员，对流程配置进行裁剪和确认。 |
| 退出标准 | 1. 测试工作量预估完成，由三位及以上专家预估取平均。 2. 测试计划、测试日程由项目组评审并通过。 3. 测试人员及角色确定。 4. 在项目开发过程中，要适时的对测试计划进行跟踪，以评估此计划的完整性、可行性，在项目结束时还要最后评估一下测试计划的质量 |
| 责任人 | 测试项目经理 |
| 测试团队产物 | 工作量预估表单、测试质量目标分析表、测试计划、测试计划评审报告、测试日程表 |

### 测试计划重点项指导

（1）确定测试策略：包括要进行的测试阶段（单元测试、集成测试和系统测试）以及要执行的测试类型（功能测试、性能测试、兼容性测试，安装卸载测试，文档测试等）。明确测试的工作范围，需要测试的对象、达到的指标等。

（2）确定测试资源：包括测试环境、测试工具、软硬件资源、人力配备。

（3）确定测试任务：根据需求，细化测试任务，划分优先级，以及辅助任务清单。如：培训等

（4）评估测试工作量：目前没有任何一种方法能准确的评估出软件测试工作的工作量，只能以持之以恒的统计和分析以往类似项目，尽量分解细化需求点，用例数量来进行经验预估，并尽量以三位及以上人员一起预估，取平均值。

（5）确定时间进度：确定各阶段任务安排和资源分配，依据项目总体时间安排，形成进度计划，为每个里程碑规定合理的目标、测试起始/中止时间。注意，通常情况下，功能性需求和非功能性需求的测试存在先后顺序，不能并行。

（6）评估风险：对测试计划中所有要执行的内容进行潜在的风险分析并给出规避措施，每人风险点确定好干系人。风险点常有：版本提交延迟，测试人员不足，需求变化太快，测试技能不足等。

（7）确定测试过程评估方法：测试过程，每天/周/月，哪些报告人，需对测试工作进展/问题、风险/过程活动质量评估。

详细可参照公司测试计划编写模板。

### 测试需求分析

客户需求说明书评审通过并基线后，测试组需要编写测试需求分析报告文档、制定测试总体策略，并设计出合理的规避风险的策略，为后续的测试工作提供直接的指导。

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 《客户需求说明书》、快速原型（可选）、《需求规格说明书》（可选） |
| 工作内容 | 1. 根据《客户需求说明书》、快速原型（可选）、《需求规格说明书》（可选），进行测试需求分析，编写测试思维导图； 2. 确定测试的范围，估算出测试所需要的测试工期和各个测试需求规格的测试优先级； 3. 组织测试内部评审，和项目组评审，并对评审意见做到闭环处理。 |
| 退出标准 | 测试需求分析报告由项目组评审并通过 |
| 责任人 | 测试项目经理、测试需求分析人员 |
| 测试产物 | 测试需求分析\_思维导图、评审测试需求分析评审报告 |

### 需项目经理支持的活动

1. 项目配置流程表裁剪时，通知测试TPM参加以便对测试环节的工作做适当裁剪
2. “主持“《需求规格说明书》评审过程：

* 提前2天左右发出初稿，通知测试人员参加预评审
* 正式评审时，及时通知测试人员参加
* 会后落实《需求规格说明书》评审意见的闭环解决，并邮件通知测试人员检查确认。及时将需求SVN基线化。"

1. 基线化后需求规格说明书如有更新，请及时告知测试TPM（明确更新点），以便测试获得最新的需求。
2. 该阶段如果测试工作量预估有更新调整，请及时评估处理。
3. 主持《项目计划书》、项目日程表（除测试阶段）评审，记得请测试TPM参与。

### 计划(报告)技巧PPT附件



## 测试设计阶段

### 设计测试用例

在需求分析文档确立基线后，测试组需要针对项目的测试需求编写测试用例，在实际的测试中，测试用例将是唯一实施标准。在用例的编写过程中，具体的任务和责任人如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 测试需求明确，测试计划，需求文档，系统设计，测试需求分析\_思维导图 |
| 工作内容 | 1. 根据测试需求分析\_思维导图，参照需求文档，编写测试用例 2. 组织测试内部评审，和项目组评审，并对评审意见做到闭环处理 |
| 退出标准 | 1. 测试用例需要覆盖所有的测试需求 2. 准入/准出用例挑选完成 3. 测试用例需进行评审并通过 4. 项目进行过程中，适时根据需求变更来对测试用例进行维护 |
| 责任人 | 测试项目经理，测试用例设计人员 |
| 测试团队产物 | 1. 系统测试用例、集成用例（可选）、稳定性测试用例、数据同步用例（可选） 2. 准入/准出测试用例 3. 主流程用例 4. 测试用例评审报告 |

### 需项目经理支持的活动

（1） 项目有架构设计方案的，需要主持架构设计方案说明书评审，邀请测试人员参加；

（2） 主持系统设计文档评审，邀请测试人员参加；

（3） 提交必要的项目设计文档给测试人员进行用例设计，如接口说明书、数据库字段等；

（4） 评审测试计划、性能方案；

（5） 评审测试日程表（在编写过程中向TPM提供协助，如确定每轮提交测试时间等），并将评审通过的测试日程表更新到项目日程表进行维护；

（6） 协助TPM组织测试需求分析评审；

（7） 协助TPM组织测试用例评审，特别要求开发团队中给出评审意见，确认准入测试用例；

（8） 组织开发团队进行准出测试：

（9） 必须执行测试团队提供的全部准入测试用例。将执行结果详细记录在准出测试执行结果表单，主要记录每条用例：通过与否、不通过原因、执行人。

### 测试用例编写技巧

**测试用例包含要素：**

测试用例主要包含的要素有：测试用例编号（与需求编号、测试需求分析点为一对多关系）、模块、标题（简要描述测试点）、优先级、预设条件（用例操作前需要具备的条件）、测试步骤、测试数据（用例执行时需要输入的数据）、预期结果、用例类型（集成用例、准入用例、系统测试用例、自动化用例）、用例状态（已评审、未评审）、用例编写人

**测试用例设计方法：**

常见的用例设计 方法有如下几种：

1. 等价类划分：将测试点划分成多个子集，再从每个子集中选择有代表的数据进行测试，通过降低测试的数目去实现“合理的”覆盖。
2. 边界值分析：一般与等价类划分结合起来。但它不是从一个等价类中任选一个例子作为代表，而是将测试边界情况作为重点目标，选取正好等于，刚刚大于或刚刚小于边界值的测试数据。
3. 错误推断法：根据经验或直觉，推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的测试用例。
4. 因果图：等价类划分和边界值方法分析都只是孤立地考虑各个输入数据的测试功能，而没有考虑多个输入数据的组合引起的错误。因果图即考虑输入条件间的组合关系，得出输出条件。因果图方法最终生成的就是判定表，它适合于检查程序输入和合种组合情况。
5. 路径覆盖：将软件系统的流程看成是路径，把所有主流程、分支流程以路径方式设计 用例，可以降低测试用例设计难度，又能搞清各种流程。

**测试用例涵盖面：**

测试用例不能只关注系统功能，还需要涵盖UI界面（易用性、提示友好、整体布局、错别字）、容错、兼容（浏览器，操作系统）、各模块关联、性能、安装升级、安全性、日志相关。

### 用例编写技巧PPT附件



## 测试阶段

### 转测试申请

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 测试设计内容评审完毕，开发团队编码工作完成，并已完成内部测试； |
| 工作内容 | 1. 开发组填写测试申请单，向测试组提交测试内容。 2. 测试小组检查转测试交付件的完整性和可测性。 3. 对待测软件进行准入测试，准入通过率至少达到90%，且无严重功能阻塞。 |
| 退出标准 | 转测试交付件已基线、经测试组预测试通过 |
| 责任人 | 产品（项目）经理，测试项目经理，质量管理部经理 |
| 测试团队产物 | 准入用例测试执行表单，准入测试结果 |

### 测试执行

实施测试用例将花费测试组大部分时间，这些工作都是建立在前期很多计划工作的基础上。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 1. 测试经理确定可测试的模块，并分配给执行人员 2. 根据测试计划及日程表，确定测试完成日期 |
| 工作内容 | 1. 测试执行工程师根据测试计划中分配给自己的测试任务和提供的测试用例，执行相应的测试用例。 2. 记录测试用例的执行结果。 3. 提交缺陷。 |
| 退出标准 | 测试用例中的所有任务被执行，结果被记录。 |
| 责任人 | 测试组成员（测试项目经理负责检查和汇总所有人员的执行表单） |
| 测试团队产物 | 测试执行表单，提交缺陷 |

### 回归测试

在每轮测试结束之后，由测试组重新获取修改后的最新版本，进行回归测试。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 缺陷管理工具上，所有缺陷都被解决，并提交回归测试申请，版本相关文档基线，版本准入测试通过。 |
| 工作内容 | 1. 在缺陷管理工具上，等待验证的缺陷进行回归测试。 2. 按照测试计划中规定的回归测试的策略对产品进行回归测试，回归测试原则上包含上版本失败+上版本阻塞+主要流程用例，时间允许可包含全部用例。 3. 记录用例执行结果。 4. 对于新发现的缺陷，提交到缺陷管理工具中。 |
| 退出标准 | 1. 回归测试所运行的用例全部通过 2. 缺陷全部验证 3. 测试人员名下无等待验证和处理的缺陷 |
| 责任人 | 测试工程师 |
| 测试 | 测试执行表单，提交缺陷 |

### 阶段性测试报告

在约定的测试周期完成之后，测试项目经理需要总结此次测试的结果，编写阶段性测试报告。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了预定周期的测试任务 |
| 工作内容 | 测试项目经理根据本轮测试的结果，编写阶段性测试报告，主要应包含以下内容：   1. 测试报告的版本。 2. 测试的人员和时间。 3. 缺陷情况：测试组在这轮测试中所有处理的缺陷，目前活动缺陷的数量及其状态分类，及这些缺陷去向和归属人。 4. 测试评估——写明本测试版本的质量情况。 5. 急待解决的问题——写明当前项目组中面临的最优先的问题。 |
| 退出标准 | 在每轮测试结束之后应尽快将符合标准的测试报告发给全项目组，以利于项目干系人了解当前项目进展及版本的质量情况。 |
| 责任人 | 测试项目经理 |
| 测试团队产物 | 阶段性测试报告 |

### 同行审查

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 回归测试结束，所有缺陷都被关闭。 |
| 工作内容 | 1. 进行对测试组所测试项目或产品的测试审查工作，基本原则:不依据所设计测试用例,进行自由测试；测试时间保持在3个正常工作日以内；如发现严重缺陷，则一轮测试结束后，更新版本，执行回归测试。 2. 提交测试纪录（主要体现覆盖的功能和流程）。 3. 编写同行审查总结报告(报告以简单为好)。 |
| 退出标准 | 同行审查没有新的缺陷或没有严重缺陷产生 |
| 责任人 | 项目组干系人 |

### 需项目经理支持的活动

1. 必要的测试环境搭建协助
2. 每轮提交测试的版本，都要配合走转测试过程：

* 项目经理通过电子邮件，创建并提交版本转测试申请给配置管理员审查，抄送给测试经理、测试代表、开发经理和产品经理。邮件附件必须附：转测试申清单(特别提醒需写明：软硬件要求、可测功能或不可测功能；修改了哪些bug；本次特别修改功能点；测试建议等)、准出测试执行结果表单、部署手册。其中部署手册建议提早编写，在第一、二次转测试申请时提交。
* 配置管理员对转测试版本进行审查：通过，将版本转测试交付件入项目配置库，并进行基线。之后将邮件转发给测试项目经理组织测试实施；不通过，将版本转测试申清单返回给项目经理，同时抄送给测试经理、测试代表、开发经理和产品经理。项目经理解决存在的问题后，重新提交申请（进行第1步）
* 测试项目经理组织预测试（0.5天），提交预测试结论与数据报告：通过，将邮件转送给配置管理员、项目经理，抄送给测试经理、开发经理和产品经理；不通过，给出不通过原因，将版本返回给项目经理。项目经理解决存在的问题后，重新提交申请（进行第1步）

1. 缺陷部分：

* 关注缺陷提交与修复进度，对于修复不及时现象的有效解决
* 跟踪致命、严重级别缺陷进展，异常情况下的必要干预
* 根据实际情况，定期组织争议缺陷讨论，引导项目组形成一致决议。"

1. 关注测试周报与例会反馈问题，及时安排解决
2. 关注测试报告：阶段测试报告（反馈的重要问题、争议问题、风险）、系统测试报告、性能测试报告

## 测试总结

### 系统测试报告

在回归测试结束之后，测试项目经理将要编写系统测试报告，对测试进行总结，并且提交给全体项目组成员，为产品的后续工作提供重要的信息支持。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了所有的测试阶段工作，同行审查结束。 |
| 工作内容 | 1. 收集测试过程中所有度量数据，编写测试报告（详细可参考2.5） 2. 如果遗留缺陷，需组织CCB进行决策。 3. 组织项目组成员评审测试报告，并输出评审报告 |
| 退出标准 | 测试项目经理完成了符合标准的测试报告，发送给全项目组。 |
| 责任人 | 测试项目经理 |
| 测试团队产物 | 系统测试报告、系统测试报告评审报告 |

### 测试报告重点项指导

（1）测试环境：编写出测试过程中使用的硬件配置，软件版本，并画出测试组网图，用于阅读人员了解整个测试环境。

（2）过程活动参与人员及工时：列出每个活动参与的人员及投入的时间，用于阅读人员了解对整个测试过程投入情况。

（3）测试版本：列出所有提交测试的版本，用于阅读人员了解整测试过程提交的版本数量。

（4）测试覆盖：包含但不限于功能测试覆盖、安装/卸载测试覆盖、文档测试覆盖、服务器兼容测试覆盖、浏览器兼容测试覆盖、稳定性测试覆盖。用于阅读人员了解测试过程中的涉及面。

（5）测试方法及工具：列出使用的测试方法和工具，用于阅读人员了解具备的测试技能。

（6）测试数据收集：包含但不限于编写用例数、准入版本质量、执行用例数、发现缺陷数。用于阅读人员了解测试规模及输出成果。

（7）测试质量统计：计算出各项指标，如进度偏差、工作量偏差、缺陷发现密度、用例评审缺陷密度、用例设计效率，用例执行效率，总结测试过程的同时，为其它项目评估提供数据依据。

（8）测试结论：此点为重中之重，列出最终测试版本，客观评价软件质量情况，明确给出“通过”或“不通过”的结论。

（9）关键风险：提出项目存在隐性和显性的风险。通常缺陷被遗留到用户群体后，随着用户增加，缺陷会被无形中放大。所以我们对系统的质量一定要严格，不容忽视任何一个风险。

（10）测试建议：站在研发人角度，对风险提出解决或规避措施。站在用户角度，对系统提出中肯的建议。

### 需项目经理支持的活动

1. 关注测试报告：系统测试报告、性能测试报告、组织开发人员进行评审。
2. 参与测试评估与总结。

## 测试验收

测试验收工作是在以上工作全部结束后，对测试的过程，效果进行验收，宣布测试结束。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 系统测试报告 、需求文档 |
| 工作内容 | 验收组成员，对本次测试进行验收，验收内容包括：   1. 测试效果验收——测试是否达到预期目的。 2. 测试文档验收——测试过程文档是否齐全，可信，符合标准。 3. 测试评估——从总体对测试的质量进行评估。 4. 测试建议——对本次测试工作指出不足，需要在以后工作中改进的地方。 5. 宣布测试结束——测试验收组成员签字宣布本次测试结束。 |
| 退出标准 | 测试验收通过，测试验收会议记录整理完毕 |
| 参与人员 | 验收组人员，质量管理部经理，测试项目经理，产品（项目）经理 |
| 责任人 | 产品（项目）经理 |

## 测试归档

测试归档是在测试验收结束宣布测试有效，结束测试后，对测试过程中涉及到各种标准文档进行归类，存档。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试验收通过 |
| 工作内容 | 测试过程涉及到的文档，归档到测试项目库存中，主要包括以下文档（必须）   1. 系统测试需要分析\_思维导图 2. 系统测试计划 3. 系统测试用例 4. 测试执行记录 5. 系统测试报告 6. 阶段性测试报告 7. 测试总结报告 |
| 退出标准 | 全部文档归类完毕，版本号封存 |
| 责任人 | 测试项目经理 |

## 测试工作总结

测试总结工作是在以上的工作全部结束以后，它的目的是评估本次测试工作，总结经验，促进测试流程及规范的改进和提高，同时也提醒全体测试组成员在以后的工作中需注意的问题。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 项目验收工作完成。 |
| 工作内容 | 由质量管理部经理，测试项目经理召开项目测试工作总结会议，会议内容主要为：   1. 测试项目经理对项目期间的整个测试组的工作情况进行总结，指出测试工作中存在的问题，同时也对工作中表现好的地方给与肯定。（具体包括整个测试情况、流程实施、人员安排、测试方法等） 2. 参与本次项目测试工作的所有成员个人体会和建议。 3. 讨论测试工作中出现的问题，寻求更好的解决办法。 4. 宣布解散测试小组。 |
| 退出标准 | 所提问题寻求到较好解决方式，测试小组解散 |
| 参与人员 | 质量管理部所有成员 |
| 责任人 | 测试项目经理 |
| 测试团队产物 | 测试工作总结文档 |

## 缺陷跟踪表

测试验收结束后，跟踪产品在试运行阶段暴露出来的新缺陷，以及已提交的缺陷是否再次发生。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了所有的测试执行工作，测试验收通过，产品试运行、运行。 |
| 工作内容 | 1. 已发现缺陷是否再次发生 2. 是否有新一轮测试中未发现的缺陷 3. 是否有新一轮测试中已发现但未修改的缺陷 |
| 退出标准 | 缺陷跟踪报告 |
| 责任人 | 产品经理、项目实施经理 |

## 发布标准

软件产品发布分为正式版和演示版两个标准。

### 正式版本发布标准

遗留缺陷加权分≤3，且通过项目组成员讨论，确定可作为遗留缺陷。所有遗留缺陷在发布前项目经理需要给出有效的规避措施。各类缺陷加权分值如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **致命性缺陷** | **严重性缺陷** | **一般性缺陷** | **提示性缺陷** | **建议类缺陷** |
| 3 | 2 | 1 | 0.2 | 0.2 |

软件产品未经测试通过，不允许发布正式商用版。

### 演示版本发布标准

遗留缺陷不影响软件主要流程，各类缺陷经项目组成员确认，不影响软件演示效果。

作为演示版本发布的软件，不允许用于正式商用版本。

### 争议性问题

如开发团队对测试结论有争议，不能通过协商解决的，由CCB成员（验收组成员）会议协调解决，CCB成员通常包含：产品部门经理和产品经理、研发部门经和项目经理、测试部门经理和项目测试经理。测试团队和开发团队应无条件服从仲裁结果。

# 测试体系-性能测试

## 测试主要对象概述

性能测试对象主要包括：性能指标、并发性、负载、压力及稳定性测试。应当在需求阶段就关注性能需求。不要在事后才想到性能测试。

性能测试的目的：为了验证系统是否达到用户提出的性能指标，同时发现系统中存在的性能瓶颈，起到优化系统的目的。

性能测试指标的来源：用户对各项指标提出的明确需求；如果用户没有提出性能指标，则根据用户需求、测试设计人员的经验来设计各项测试指标。（需求+经验）

主要的性能指标：服务器的各项指标（CPU、内存占用率、I/O等）、后台数据库的各项指标、网络流量、响应时间。

性能BUG观点：

1. 性能测试就像人在无风情况下跑步（正常情况下的性能指标）；
2. 压力测试就像人在微风中跑步（在正常的基础上加大多少百分比压力的性能指标）；
3. 负载测试就像人在强风中跑步（不断加压，直到系统崩溃）。

HTTP观点：

1. 负载测试是正常情况下持续的加压；
2. 压力测试是直接加压达到一个极限值。

统一的观点：

1. 性能测试、压力测试、负载测试密不可分，可统称为性能测试。

## 性能测试流程



## 设计阶段

### 与项目经理确认测试的功能点

该阶段主要是测试设计人员，项目经理和开发人员一起确定比较容易出问题的一些功能点，再根据这些功能点进行筛选按照《性能测试计划模板》的格式编写《性能测试计划》，《性能测试方案模板》的格式编写《性能测试方案》，按照《性能测试用例模板》的格式编写《测试用例文档》。

在编写文档期间，关于每个功能点的“正常数据量”，“正常用户量”，“正常并发量”需要和项目经理仔细讨论，并记录在测试用例文档内；如果遇到项目经理不太清楚的功能点，测试设计人员可以根据经验来做出判断。

该阶段需要测试设计人员对项目的业务和需求比较了解，也需要一定的项目经验。

在产品立项阶段，与项目经理确认性能测试的功能点，并明确测试环境部署图、软件硬版本及性能测试指标。具体任务和责任如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 立项报告，用户需求说明书 |
| 工作内容 | 根据开发计划编写测试计划，根据用户需求与项目经理确认性能测试的功能点，并明确测试环境部署图、软件硬版本及性能测试指标。 |
| 退出标准 | 1. 已输出测试计划； 2. 性能测试功能点已明确； 3. 性能测试环境及部署图已确定； 4. 性能指标已明确。 |
| 责任人 | 性能测试负责人，项目经理 |

### 设计测试方案/用例

在系统需求规格说明书确立基线以后，测试组针对项目的测试需求编写测试计划和测试方案/用例，在实际的测试中，测试方案/用例将是唯一实施标准。在方案/用例的编写过程中，具体的任务和责任人如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 性能测试需求明确，性能测试需求文档，测试计划 |
| 工作内容 | 根据测试需求设计测试方案/用例 |
| 退出标准 | 1. 测试方案/用例需要覆盖所有的测试需求 2. 测试方案/用例需要进行评审并通过 3. 项目进行过程中，适时的根据需求变更来对测试方案/用例进行维护 |
| 责任人 | 性能测试负责人 |
| 性能测试团队产物 | 性能测试日程安排、性能测试方案/用例、评审报告 |

## 实施阶段

该阶段通过测试工具录制、运行脚本，把运行的结果按照《性能测试报告模板》的格式填写《性能测试报告》。

### 录制/编写测试脚本

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 性能测试日程、性能测试方案/用例评审通过； |
| 工作内容 | 1. 根据测试方案/用例编写所需测试脚本； 2. 对测试脚本进行调试。   3、准备测试数据 |
| 退出标准 | 测试脚本调试通过、测试数据准备完成 |
| 责任人 | 测试开发工程师 |
| 性能测试团队产物 | 性能测试脚本、测试数据生成语句或数据库备份 |

### 运行脚本

实施测试用例将花费测试组大部分时间，这些工作都是建立在前期很多计划工作的基础上。

该阶段对测试实施人员的要求比较低，只需要经过短时间内培训，在专业性能测试人员的指导下就可以进行完整测试实施。此阶段工作量比较大，但是，此阶段又是整个性能测试流程中比较重要的阶段，测试实施人员犯的错误会直接影响到后期的分析，所以，还是要对这个阶段的工作特别细心，特别重视。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 性能测试日程表，规定每轮性能测试结束时间。 |
| 工作内容 | 1. 测试执行工程师根据测试计划中分配给自己的测试任务和提供的脚本及用例，实施相应的测试。 2. 记录实施测试的结果。 3. 提交缺陷。 |
| 退出标准 | 测试用例中的所有任务场景被执行，结果被记录。 |
| 责任人 | 测试组成员 |
| 性能测试团队产物 | 性能测试执行结果，提交缺陷。 |

### 测试结果分析

在约定的测试周期完成之后，测试项目经理需要总结此次测试的结果，编写阶段性测试报告。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了预定周期的测试任务 |
| 工作内容 | 1. 测试项目经理根据此轮测试的结果，编写阶段性测试报告 2. 分析结果，提出性能优化建议 |
| 退出标准 | 在每轮测试结束之后应尽快将符合标准的测试报告发给全项目组 |
| 责任人 | 测试项目经理 |

## 性能测试报告

该阶段也是测试小组内部完成。测试分析人员根据得到的《性能测试报告》，“测试脚本运行结果”进行分析，按照《性能测试分析报告模板》的格式填写《性能测试分析报告》。

目前，我们的分析，是功能点的响应时间，事务处理能力， 服务器 CPU 和内存来的一些参数来对功能点进行分析，得到该版本的项目在该测试平台的一个整体的性能报告。

该阶段需要测试分析人员根据一些数据，一些图表来判断项目的性能，对测试分析人员有比较高的分析能力。如果找到性能上出现问题的功能点，在测试分析人员的能力范围内，需要定位问题的原因。

在回归测试结束之后，测试项目经理将要编写性能测试报告，对性能测试进行总结，并且提交给全体项目组，为产品的后续工作提供重要的信息支持。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了所有的测试实施工作,同行审查结束. |
| 工作内容 | 测试项目经理根据测试的结果，按照测试总结报告的文档模板编写测试报告 |
| 退出标准 | 测试项目经理完成了符合标准的测试报告，发送给全项目组。 |
| 责任人 | 测试项目经理 |

## 测试归档

测试归档是在结束测试后，对测试过程中涉及到各种标准文档进行归类，存档。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试验收通过 |
| 工作内容 | 归类、存档测试过程涉及到的文档，主要包括以下文档（必须）   1. 性能测试计划 2. 性能测试方案/用例 3. 性能测试脚本，背景数据 4. 性能测试报告 5. 性能阶段性测试报告 6. 性能测试总结报告 |
| 退出标准 | 全部文档归类完毕，版本号封存 |
| 责任人 | 测试项目经理 |

## 测试过程所需要的专业知识

## 测试人员具备的专业知识

1. 掌握性能测试理论知识；
2. 熟悉常用性能测试工具使用技巧，如：LoadRunner、HTTPWatch
3. 具备编写C脚本语言的能力；
4. 测试背景数据创建能力，如：熟悉系统表结构及关系，编写sql生成背景数据；
5. 了解系统性能计数器；
6. 较强的分析能力。
7. 了解网络知识；
8. 对项目业务和需求比较了解；
9. 有一定的项目测试经验。

## 测试过程的每周四点

难点：

1. 测试背景数据模型的建立与模拟数据的生成；
2. 多终端的模拟；
3. 测试结果分析。

关键点：

1. 测试环境必须是尽可能的去除外界人为干扰，最好能单独访问该网站；
2. 测试各个阶段中，每个阶段的测试人员只需要了解前一个阶段测试结果；
3. 性能测试是在功能测试完成之后进行；
4. 性能测试计划、方案一般与测试用例统一在一个文档里；
5. 测试环境应尽量与用户环境保持一致；
6. 性能测试一般使用测试工具和测试人员编制测试脚本来完成，性能测试的环境应单独运行尽量避免与其他软件同时使用；
7. 性能测试的重点在于前期数据的设计与后期数据的分析。

风险点：

1. 多业务交叉混合场景的模拟；
2. 功能不稳定，频繁变更。

问题点：

1. 受相关网关的制约；
2. 参数化不能存在空格；
3. 参数化要注意参数的设置方式。

# 测试体系-兼容性测试

## 测试主要对象概述

兼容性测试的目的：被测试项目在不同的操作系统平台上正常运行，包括在同一操作系统平台的不同版本上正常运行；能与相关的其他软件或系统的“和平共处”；能在指定的硬件环境中正常运行；能在不同的网络环境中正常运行。

软件兼容性测试主要对象包括

1. 操作系统/平台兼容性测试；
2. 应用软件之间的兼容性测试；
3. 不同浏览器之间的兼容性测试；
4. 不同类型的数据库兼容性测试；
5. 软硬件配合的兼容性测试；
6. 不同分辨率兼容性测试；
7. 其他方面，例如操作系统语言、传输协议、代理服务器、防火墙、自身产品集成。

* **操作系统/平台兼容性测试**

市场上有很多不同的操作系统类型，最常见的有Windows、Unix、Macintosh、Linux等。对于普通用户来讲，最常用的是Windows操作系统。Windows操作系统包括Windows XP，windows 2003，windows 2008，Windows7等等。我们应当针对当前客户主流操作系统版本进行兼容性测试，在确保主流操作系统版本兼容性测试的前提下，在对非主流操作系统版本进行测试，尽量保证项目的操作系统版本的兼容性测试的完整性。

* **应用软件之间兼容性测试**

主要考察两项内容：一是软件运行需要哪些应用软件支持，及应用软件的不同版本。二是判断与其他常用软件一起使用，是否会造成其他软件运行错误或本身不能正确实现其功能。

* **不同浏览器之间的兼容性测试**

针对B/S结构软件，大多客户端都使用浏览器。因此，浏览器是Web客户端最核心的构件，不同厂商、不同版本的浏览器对某些构件和设置的适应性，也是软件兼容性测试的重点之一。最常见的浏览器有：IE8/9/10/11、谷歌、火狐、360等等，有时对同一浏览器的不同小版本号的兼容情况都会有所不同。

需要在确保客户主流浏览器的兼容性测试通过的前提下，在对非主流浏览器（含版本）进行测试，尽量保证项目的浏览器的兼容性测试的完整性。

* **不同类型的数据库兼容性测试**

现在很多软件都需要有数据库系统的支持，对此类软件应考虑对不同数据库平台的支持能力，如从BD2平台替换到MSSQL平台时，软件是否可直接挂接，或者提供相关的转换工具。还有新旧数据转换的是否存在问题，软件是否提供新旧数据转换的功能。例如，当软件升级后可能会定义新的数据格式或文件格式，这就涉及到对原有格式的支持及更新，原有用户记录在新格式下是否依然可用等。另外，还需要测试转换过程中数据的完整性与正确性。

* **软硬件配合的兼容性测试**

考察软件对运行硬件环境有无特殊说明，有些软件可能在不同的硬件环境中，出现不同的运行结果或是根本就不能执行。

* **不同分辨率兼容性测试**

分辨率的测试是为了页面的版式在不同的分辨率模式下能正常显示，字符符合要求而进行的测试。现在常见的分辨率是1440\*900，1366\*768、1280\*800，1024\*768。对于需求规格说明书中规定的分辨率，测试必须保证测试通过。对于需求规格说明书中没有规定分辨率的项目，测试应该在完成主流分辨率的兼容性测试的前提下，尽可能进行一些非主流分辨率的兼容性测试，在一定程度上保证大部分。

## 兼容性测试流程

通常情况下，兼容性测试流程在系统测试过程流程中一起进行



## 设计阶段

### 确定测试的功能点

该阶段主要是测试设计人员与项目经理、产品经理一起确定兼容性测试点，再根据这些测试点进行优先级排序并编写《测试方案》和《测试用例》。

具体任务和责任如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 立项报告，用户需求说明书 |
| 工作内容 | 根据开发计划编写测试计划，根据用户需求和项目经理、产品经理共同确定兼容性测试的点，并了解项目所采用的技术框架。 |
| 退出标准 | 1. 已输出测试计划； 2. 兼容性测试项已明确； 3. 兼容性测试环境及部署图已确定。 |
| 责任人 | 测试组成员，项目经理 |
| 测试团队产物 | 兼容性测试需求分析 |

### 设计测试方案/用例

在系统需求规格说明书确立基线以后，测试组针对项目的测试需求，编写测试计划和测试方案/用例，在实际的测试中，测试方案/用例将是唯一实施标准。在方案/用例的编写过程中，具体的任务和责任人如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 测试需求明确，需求文档，测试计划 |
| 工作内容 | 根据测试需求设计测试方案/用例 |
| 退出标准 | 1. 测试方案/用例需要覆盖所有的测试需求 2. 测试方案/用例需要进行评审并通过 3. 项目进行过程中，适时的根据需求变更来对测试方案/用例进行维护 |
| 责任人 | 测试组成员 |
| 测试团队产物 | 兼容性测试用例 |

## 实施阶段

实施测试用例将花费测试组大部分时间，这些工作都是建立在前期很多计划工作的基础上。

该阶段对测试实施人员的要求比较低，只需要经过短时间内培训，在测试经理的指导下就可以进行完整测试实施。此阶段工作量比较大。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试项目经理于前一工作日定出当日的测试计划，确定可用的测试用例。 |
| 工作内容 | 1. 测试执行工程师根据测试计划中分配给自己的测试任务和提供的测试用例，实施相应的测试。 2. 记录实施测试的结果，提交当日测试纪录。 3. 提交缺陷。 |
| 退出标准 | 测试用例中的所有任务被执行，结果被记录。 |
| 责任人 | 测试组成员 |

## 兼容性测试报告

该阶段需要测试分析人员兼容性测试结果来评估系统的兼容性，对测试分析人员有比较高的分析能力。如果找到兼容性的问题点，在测试分析人员的能力范围内，需要定位问题的原因。

在回归测试结束之后，测试项目经理将要编写兼容性测试报告，对兼容性测试进行总结，并且提交给全体项目组，为产品的后续工作提供重要的信息支持。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了所有的测试实施工作,同行审查结束. |
| 工作内容 | 测试项目经理根据测试的结果，按照测试总结报告的文档模板编写测试报告 |
| 退出标准 | 测试项目经理完成了符合标准的测试报告，发送给全项目组。 |
| 责任人 | 测试项目经理 |

## 测试过程所需要的专业知识

* 各种测试手段和测试工具的应用能力；
* 熟悉各种兼容性测试工具；
* 要有兼容性测试的经验；
* 对各种与兼容性测试相关的软件要熟悉，如：操作系统、浏览器、数据库等等

## 测试过程的每周四点

**难点：**

* 遍历所有的兼容性测试；
* 兼容性测试覆盖度分析；
* 兼容性测试或许无法做到完全的质量保证。

**关键点：**

* 已确认的兼容性测试环境部署与模拟；
* 尽早介入兼容性测试需求分析；
* 借助于工具进行兼容性测试。

**风险点：**

* 兼容性测试分析不全面；

**问题点：**

* 兼容性测试环境无法满足。

## 浏览器兼容性常用测试工具

* IETester

IETester是一个免费的WebBrowser控件，让您有渲染和IE8的JavaScript引擎，IE7和IE 6在Windows 7，Vista和XP的IE5.5中，以及在同一进程中安装的IE浏览器。新的v0.4.2：新增开发选项来启用/禁用浏览器的功能！V0.4.11版本还提供DeBugBar插件。

DebugBar是功能强的IE插件,类似于Firefox浏览器的一款开发类插件Firebug.从各个不同的角度剖析Web页面内部的细节层面，给Web开发者带来很大的便利.

目前对于个人用户免费.企业用户收费.

主要有以下功能：

* 查看网页DOM树,HTTP(S),网页javascript脚本函数归类展示.效验网页中的框架.网页属性的详细信息；
* 提供网页源码与IE整理后代码的对比；
* 能在无需改变显示器分辨率的前提下用不同分辨率显示页面.并能对页面进行缩放；
* 功能对任意位置取色并将HEX码复制到剪贴板中便于保存；
* 可将当前页面截图通过邮箱发送出去。
* IE netrenderer

IE Netrenderer是一个firefox插件，用于检查你的网站在IE浏览器中的呈现情况，包括各个常用版本的检测。IE NetRenderer：能实时生成你的网站在 Internet Explorer 5.5 、6.0 和 7.0 下的截图。

IE NetRenderer便是一个专门检验网站页面在不同版本的IE中展示效果的网络服务，可以快速查看IE5.5, IE6和IE7下的页面效果，而且还能同时查看在IE6，IE7下的叠加效果和差异效果，很有特色。

* Viewlike.us!

ViewLike.us 是一个可以让你输入网址，让你观看在不同屏幕分辨率下网站的显示效果。

其可供调整的屏幕解析度有：iPhone、Wii Browser、800×600、1024×768、1152×864、1280×800、1400×900、1600×1200与1920×1200。

* Browsera

Browsera 是一个可测试您的网站的跨浏览器布局的工具，您会看到您网站上存在的兼容性错误。

六大优点：发现布局问题、找到javascript错误、更方便的测试网站、不需登陆就能测试需登陆的网站、无需安装、测试动态页面。

* IE Collection

Utilu IE Collection 是个专为程序设计与网页设计师制作的各版本 Internet Explorer 浏览器整合安装包，里面包含了IE 1.0、1.5、3.0、5.5、6.0、7.0、8.0...等等版本的IE网络浏览器。主要目的是让设计师在做好网页后在不同版本的IE浏览器里面测试版面样式与功能，看看是否版面有乱掉或功能不正常…等问题。

在IE Collection中各个版本的IE浏览器都是各自独立的，可以单独执行，此外还提供了Internet Explorer Developer Toolbar工具与Firebug Web Development Extension for Internet Explorer外掛程序，可以让我们通过这些工具检查网页内容的格式、版面错乱或程序错误...等。

安装完成之后，跟ie浏览器一样，没有差别。

注：安装时不要选，建立桌面快捷文件。否则桌面会出现一大片的ie快捷文件。

# 测试体系-安全性测试

## 测试主要对象概述

软件安全性是一个广泛而复杂的主题，每一款新的软件总可能有完全不符合所有已知模式的新型安全性缺陷出现。要避免因安全性缺陷问题受各种可能类型的攻击是不切实际的。在软件安全测试时，运用一组好的原则来避免不安全的软件上市、避免不安全软件受攻击，就显得十分重要。

安全性测试的主要目的：查找软件自身程序设计中存在的安全隐患，并检查应用程序对非法侵入的防范能力，根据安全指标不同测试策略也不同，如果遵循相同的原则，去证明软件的安全性，将有利于软件安全测试的工作规范的进行，有利于软件安全测试工作的发展。

软件安全性测试对象包括程序、网络、数据库安全性测试。根据系统安全指标不同测试策略也不同。

* **用户程序安全的测试要考虑问题包括：**

1. 明确区分系统中不同用户权限;
2. 系统中会不会出现用户冲突;
3. 系统会不会因用户的权限的改变造成混乱;
4. 用户登陆密码是否是可见、可复制;
5. 是否可以通过绝对路径登陆系统(拷贝用户登陆后的链接直接进入系统);
6. 用户推出系统后是否删除了所有鉴权标记，是否可以使用后退键而不通过输入口令进入系统。

* **系统网络安全的测试要考虑问题包括：**

1. 测试采取的防护措施是否正确装配好，有关系统的补丁是否打上;
2. 模拟非授权攻击，看防护系统是否坚固;
3. 采用成熟的网络漏洞检查工具检查系统相关漏洞;
4. 采用各种木马检查工具检查系统木马情况;
5. 采用各种防外挂工具检查系统各组程序的客外挂漏洞。

* **数据库安全考虑问题：**

1. 系统数据是否机密(比如对银行系统，这一点就特别重要，一般网站就没有太高要求);
2. 系统数据的完整性;
3. 系统数据可管理性;
4. 系统数据的独立性;
5. 系统数据可备份和恢复能力(数据备份是否完整，可否恢复，恢复是否可以完整)。

**外部输入恶意数据可以直接构成严重的软件安全漏洞常规方法如下：**

1. 命令注入 ；
2. 跨站脚本 ；
3. 拒绝服务 ；
4. HTTP响应截断 ；
5. 路径操纵 ；
6. 资源注入 ；
7. 配置操纵 ；
8. SQL注入 ；
9. 访问控制：数据库 ；
10. 日志伪造

**目前主要安全测试方法有：**

1. 静态的代码安全测试：主要是通过对软件系统的源代码进行安全扫描，根据程序中数据流，控制流，语义等信息与其特有软件安全规则库进行匹对，从中找出代码中潜在的安全漏洞。静态的源代码安全测试是非常有用的方法，它可以在编码阶段找出所有可能存在安全风险的代码，这样开发人员可以在早期解决潜在的安全问题。而正因为如此，静态代码测试比较适用于早期的代码开发阶段，而不是测试阶段。同时，由于关系到开发部门，测试部门，管理部门等多个部门的工作，在实际的贯彻实施工作上有一定的难度。
2. 动态的渗透测试：渗透测试也是我们常用的安全测试方法。使用自动化工具或者人工的方法模拟黑客的输入，对应用系统进行攻击性测试，从中找出运行时刻所存在的安全漏洞。这种是测试的特点就是真实有效，一般找出来的问题都是正确的，也是较为严重的。但渗透测试一个致命的缺点就是，由于模拟的测试数据只能到达有限的测试点，覆盖率很低。覆盖率只能达到20%-30%。
3. 程序数据扫描：一个有高安全性需求的软件，在运行过程中数据是不能遭到破坏的，否则就会导致缓冲区溢出类型的攻击。数据扫描的手段通常是进行内存测试，内存测试可以发现许多诸如缓冲区溢出之类的漏洞，而这类漏洞使用除此之外的测试手段都难以发现。例如，对软件运行时的内存信息进行扫描，看是否存在一些导致隐患的信息，当然这需要专门的工具来进行验证，手工做是比较困难的。

**安全性测试常用技术：**

**反向安全性测试过程：**大部分软件的安全测试都是依据缺陷空间反向设计原则来进行的，即事先检查哪些地方可能存在安全隐患，然后针对这些可能的隐患进行测试。因此，反向测试过程是从缺陷空间出发，建立缺陷威胁模型，通过威胁模型来寻找入侵点，对入侵点进行已知漏洞的扫描测试。好处是可以对已知的缺陷进行分析，避免软件里存在已知类型的缺陷，但是对未知的攻击手段和方法通常会无能为力。

1. 建立缺陷威胁模型。建立缺陷威胁模型主要是从已知的安全漏洞入手，检查软件中是否存在已知的漏洞。建立威胁模型时，需要先确定软件牵涉到哪些专业领域，再根据各个专业领域所遇到的攻击手段来进行建模。
2. 寻找和扫描入侵点。检查威胁模型里的哪些缺陷可能在本软件中发生，再将可能发生的威胁纳入入侵点矩阵进行管理。如果有成熟的漏洞扫描工具，那么直接使用漏洞扫描工具进行扫描，然后将发现的可疑问题纳入入侵点矩阵进行管理。
3. 入侵矩阵的验证测试。创建好入侵矩阵后，就可以针对入侵矩阵的具体条目设计对应的测试用例，然后进行测试验证。

**正向安全性测试过程：**为了规避反向设计原则所带来的测试缺乏完备性，需要一种正向的测试方法来对软件进行比较完备的测试，使测试过的软件能够预防未知的攻击手段和方法。

1. 先标识测试空间。对测试空间的所有的可变数据进行标识，由于进行安全性测试的代价高昂，其中要重点对外部输入层进行标识。例如，需求分析、概要设计、详细设计、编码这几个阶段都要对测试空间进行标识，并建立测试空间跟踪矩阵。
2. 精确定义设计空间。重点审查需求中对设计空间是否有明确定义，和需求牵涉到的数据是否都标识出了它的合法取值范围。在这个步骤中，最需要注意的是精确二字，要严格按照安全性原则来对设计空间做精确的定义。
3. 标识安全隐患。根据找出的测试空间和设计空间以及它们之间的转换规则，标识出哪些测试空间和哪些转换规则可能存在安全隐患。例如,测试空间愈复杂，即测试空间划分越复杂或可变数据组合关系越多也越不安全。还有转换规则愈复杂，则出问题的可能性也愈大，这些都属于安全隐患。
4. 建立和验证入侵矩阵。安全隐患标识完成后，就可以根据标识出来的安全隐患建立入侵矩阵。列出潜在安全隐患，标识出存在潜在安全隐患的可变数据，和标识出安全隐患的等级。其中对于那些安全隐患等级高的可变数据，必须进行详尽的测试用例设计。

**正向与反向测试的区别：**正向测试过程是以测试空间为依据寻找缺陷和漏洞，反向测试过程则是以已知的缺陷空间为依据去寻找软件中是否会发生同样的缺陷和漏洞，两者各有其优缺点。反向测试过程主要的一个优点是成本较低，只要验证已知的可能发生的缺陷即可，但缺点是测试不完善，无法将测试空间覆盖完整，无法发现未知的攻击手段。正向测试过程的优点是测试比较充分，但工作量相对来说较大。因此，对安全性要求较低的软件，一般按反向测试过程来测试即可，对于安全性要求较高的软件，应以正向测试过程为主，反向测试过程为辅。

## 安全性测试流程



## 设计阶段

### 确定安全性测试点

该阶段主要是测试设计人员根据公司已有规范和经验库确定安全性测试点，再根据这些测试点进行优先级排序并编写《测试方案》和《测试用例》。

在编写文档期间，要参考已有的安全性测试规范和经验库。

该阶段需要测试设计人员对项目所采用的技术架构比较了解。

具体任务和责任如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 立项报告，用户需求说明书 |
| 工作内容 | 根据开发计划编写测试计划，根据用户需求和项目经理、产品经理共同确定安全性测试的点，并了解项目所采用的技术框架。 |
| 退出标准 | 1. 已输出测试计划； 2. 安全性测试项已明确； 3. 安全性测试环境及部署图已确定。 |
| 责任人 | 测试组成员，项目经理 |

### 设计测试方案/用例

在系统需求规格说明书确立基线以后，测试组针对项目的测试需求编写测试计划和测试方案/用例，在实际的测试中，测试方案/用例将是唯一实施标准。在方案/用例的编写过程中，具体的任务和责任人如下：

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 测试需求明确，需求文档，测试计划 |
| 工作内容 | 根据测试需求设计测试方案/用例 |
| 退出标准 | 1. 测试方案/用例需要覆盖所有的测试需求 2. 测试方案/用例需要进行评审并通过 3. 项目进行过程中，适时的根据需求变更来对测试方案/用例进行维护 |
| 责任人 | 测试组成员 |

### 设计内容评审

该阶段主要是测试小组内部完成。测试实施人员在得到分析阶段编写的《安全性测试用例》

测试计划及测试方案/用例的设计工作完成后，需通知项目组干系人组织评审会议。在这之前需要将待评审的内容发给相关人员熟悉和理解。

|  |  |
| --- | --- |
| **过程要点** | **详细说明** |
| 输入条件 | 测试计划、测试方案/用例完成 |
| 工作内容 | 评审测试计划内容的正确性及合理性：   1. 测试环境、测试资源； 2. 测试需求范围，各个测试需求的优先级； 3. 测试策略及风险管理等；   评审测试方案/用例：   1. 测试方案/用例优先级 2. 测试方案/用例基于需求的覆盖程度 |
| 退出标准 | 测试计划及测试方案/用例评审通过 |
| 责任人 | 同行测试组，项目经理 |

## 实施阶段

实施测试用例将花费测试组大部分时间，这些工作都是建立在前期很多计划工作的基础上。

该阶段对测试实施人员的要求比较低，只需要经过短时间内培训，在测试经理的指导下就可以进行完整测试实施。此阶段工作量比较大。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试项目经理于前一工作日定出当日的测试计划，确定可用的测试用例。 |
| 工作内容 | 1. 测试执行工程师根据测试计划中分配给自己的测试任务和提供的测试用例，实施相应的测试。 2. 记录实施测试的结果，提交当日测试纪录。 3. 提交缺陷。 |
| 退出标准 | 测试用例中的所有任务被执行，结果被记录。 |
| 责任人 | 测试组成员 |

## 安全性测试报告

该阶段需要测试分析人员安全性测试结果来评估系统的安全性，对测试分析人员要求有比较高的分析能力。如果找到安全性的问题点，在测试分析人员的能力范围内，需要定位问题的原因。

在回归测试结束之后，测试项目经理将要编写安全性测试报告，对安全性测试进行总结，并且提交给全体项目组，为产品的后续工作提供重要的信息支持。

|  |  |
| --- | --- |
| **过程要点** | **详细描述** |
| 输入条件 | 测试组完成了所有的测试实施工作,同行审查结束. |
| 工作内容 | 测试项目经理根据测试的结果，按照测试总结报告的文档模板编写测试报告 |
| 退出标准 | 测试项目经理完成了符合标准的测试报告，发送给全项目组。 |
| 责任人 | 测试项目经理 |

## 测试过程所需要的专业知识

做好软件安全性测试的必要条件是：一是充分了解软件安全漏洞，二是评估安全风险，三是拥有高效的软件安全测试技术和工具。

（1）充分了解软件安全漏洞：评估一个软件系统的安全程度，需要从设计、实现和部署三个环节同时着手。

（2）安全性测试的评估：当完成安全性测试后，软件是否能够达到预期的安全程度呢？这是安全性测试人员最关心的问题，因此需要建立对测试后的安全性评估机制。一般从以下两个方面进行评估。①安全性缺陷数据评估。如果发现软件的安全性缺陷和漏洞越多，可能遗留的缺陷也越多。进行这类评估时，必须建立基线数据作为参照，否则评估起来没有依据就无法得到正确的结论。②采用漏洞植入法来进行评估。漏洞植入法和可靠性测试里的故障插入测试是同一道理，只不过这里是在软件里插入一些有安全隐患的问题。采用漏洞植入法时，先让不参加安全测试的特定人员在软件中预先植入一定数量的漏洞，最后测试完后看有多少植入的漏洞被发现，以此来评估软件的安全性测试做得是否充分。

（3）采用安全测试技术和工具：可使用专业的具有特定功能的安全扫描软件来寻找潜在的漏洞，将已经发生的缺陷纳入缺陷库，然后通过自动化测试方法来使用自动化缺陷库进行轰炸测试。

（4）熟悉常用安全性测试工具使用技巧，如：CSRFTester

（5）掌握跨站脚本攻击技术。

（6）网络知识；

（7）Web安全性测试知识；

（8）对项目业务和需求比较了解；

（9）有一定的项目经验。

## 测试过程每周的四点

**难点：**

（1）安全性测试覆盖率分析；

（2）敏感数据的定义；

（3）安全性漏洞分析；

（4）测试结果分析。

**关键点：**

（1）各类型异常数据的构造与模拟；

（2）尽早介入安全性分析；

（3）测试环境应尽量与用户环境保持一致；

（4）借助于工具进行安全性测试。

**风险点：**

（1）安全性测试分析不全面；

（2）安全性测试技术不够成熟。

**问题点：**

（1）对未知的攻击手段和方法无法预知。

## 测试经验知识库

### 过去犯错经验库

1. 页面中的文本输入框未对非法转意字符进行判断，如：<tr>test</tr>,导致响应页面被变形等；
2. IE地址栏中未对用户权限进行控制，输入绝对请求信息可跳过权限控制而直接进入系统。

### 案例库

**默认账号/密码在产品发布后未曾要求修改：**一般系统都会设置默认账号/密码，若首次登陆时系统时未做修改密码提示，在产品试运行阶段大多情况下用户不会对密码过行修改，将对系统存在一定的风险。

**页面脚本中记录了明文的用户账号/密码：**曾经测试过WEB产品，在多个模块中的登陆成功页面脚本中均保留了用户的账号和密码信息，对用户个人信息存在较大的安全隐患。

### 经典记录库

**缓冲区溢出：**缓冲区溢出已成为软件安全的头号公敌，许多实际中的安全问题都与它有关。造成缓冲区溢出问题通常有以下两种原因。①设计空间的转换规则的校验问题。即缺乏对可测数据的校验，导致非法数据没有在外部输入层被检查出来并丢弃。非法数据进入接口层和实现层后，由于它超出了接口层和实现层的对应测试空间或设计空间的范围，从而引起溢出。②局部测试空间和设计空间不足。当合法数据进入后，由于程序实现层内对应的测试空间或设计空间不足，导致程序处理时出现溢出。

**加密弱点：**这几种加密弱点是不安全的：①使用不安全的加密算法。加密算法强度不够，一些加密算法甚至可以用穷举法破解。②加密数据时密码是由伪随机算法产生的，而产生伪随机数的方法存在缺陷，使密码很容易被破解。③身份验证算法存在缺陷。④客户机和服务器时钟未同步，给攻击者足够的时间来破解密码或修改数据。⑤未对加密数据进行签名，导致攻击者可以篡改数据。所以，对于加密进行测试时，必须针对这些可能存在的加密弱点进行测试。

**错误处理：**一般情况下，错误处理都会返回一些信息给用户，返回的出错信息可能会被恶意用户利用来进行攻击，恶意用户能够通过分析返回的错误信息知道下一步要如何做才能使攻击成功。如果错误处理时调用了一些不该有的功能，那么错误处理的过程将被利用。错误处理属于异常空间内的处理问题，异常空间内的处理要尽量简单，使用这条原则来设计可以避免这个问题。但错误处理往往牵涉到易用性方面的问题，如果错误处理的提示信息过于简单，用户可能会一头雾水，不知道下一步该怎么操作。所以，在考虑错误处理的安全性的同时，需要和易用性一起进行权衡。

**权限过大：**如果赋予过大的权限，就可能导致只有普通用户权限的恶意用户利用过大的权限做出危害安全的操作。例如没有对能操作的内容做出限制，就可能导致用户可以访问超出规定范围的其他资源。进行安全性测试时必须测试应用程序是否使用了过大的权限，重点要分析在各种情况下应该有的权限，然后检查实际中是否超出了给定的权限。权限过大问题本质上属于设计空间过大问题，所以在设计时要控制好设计空间，避免设计空间过大造成权限过大的问题。

### 窍门库

如何测试XSS漏洞？

方法一：查看代码，查找关键的变量, 客户端将数据传送给Web 服务端一般通过三种方式 Querystring, Form表单，以及cookie. 例如在ASP的程序中，通过Request对象获取客户端的变量

|  |
| --- |
| <%  strUserCode = Request.QueryString(“code”);  strUser = Request.Form(“USER”);  strID = Request.Cookies(“ID”);  %> |

假如变量没有经过htmlEncode处理， 那么这个变量就存在一个XSS漏洞

方法二：准备测试脚本

|  |
| --- |
| "/><script>alert(document.cookie)</script><!--  <script>alert(document.cookie)</script><!--"onclick="alert(document.cookie) |

在网页中的Textbox或者其他能输入数据的地方，输入这些测试脚本， 看能不能弹出对话框，能弹出的话说明存在XSS漏洞。

在URL中查看有那些变量通过URL把值传给Web服务器， 把这些变量的值退换成我们的测试的脚本。 然后看我们的脚本是否能执行。

方法三：自动化测试XSS漏洞：现在已经有很多XSS扫描工具了。 实现XSS自动化测试非常简单，只需要用HttpWebRequest类。 把包含XSS测试脚本。发送给Web服务器。 然后查看HttpWebResponse中，我们的XSS测试脚本是否已经注入进去了。

# 测试体系-专项测试

## 卸载与安装测试

* 卸载测试检查注意事项

1. 文件：安装目录里的文件及文件夹（如：程序安装在几处的)；
2. 非安装目录（向系统其它地方添加的文件及文件夹）；
3. 快捷方式－（桌面，菜单，任务栏，系统栏,控件面板，系统服务列表等）；
4. 复原方面－卸载后，系统能否恢复到软件安装前的状态（包含目录结构、动态库,注册表，系统配置文件,驱动程序,关联情况等）(专门的测试工具regsnap)
5. 卸载方式－程序自带卸载程序/系统的控件面板卸载/其它自动卸载工具（如:优化大师)；
6. 卸载状态－程序在运行/暂停/终止等状态时的卸载；
7. 非正常卸载情况－卸载软件过程中，取消卸载进程，然后，观察软件能否继续正常使用；
8. 冲击卸载－在卸载的过程中，中断电源，然后，启动计算机后，重新卸载软件，如果软件无法卸载，则重新安装软件，安装之后再重新卸载；
9. 卸载环境－不同的(操作系统，硬件环境，网络环境等)下进行卸载；
10. 卸载后,该系统是否对其他的应用程序造成不正常影响（如操作系统,应用软件等)。

* 安装测试检查注意事项

1. 程序安装正确，安装后能正确运行；
2. 安装手册给的所有步骤得到验证；
3. 安装过程中所有缺省选项得到验证；
4. 测试各种不同的安装组合，并验证各种不同组合的正确性(包括参数组合，控件执行顺序组合，产品安装组件组合，产品组件安装顺序组合(如b/s)等)；
5. 安装过程中异常配置或状态（非法和不合理配置）情况进行了测试(如：断电；数据库终止，网络终止等)；
6. 安装后是否能产生正确的目录结构和文件，文件属性正确；
7. 安装后动态库是否正确；
8. 安装后没有生成多余的目录结构，文件，注册表信息，快捷方式等；
9. 安装测试应该在所有的运行环境上进行验证(手册上指定如：操作系统，数据库，硬件环境，网络环境等)；
10. 自动安装还是手工配置安装；
11. 至少要在一台笔记本上进行安装/卸载测试，因为有很多产品在笔记本中会出现问题，尤其是系统级的产品；
12. 安装该系统是否对其他的应用程序造成不正常影响（如操作系统,应用软件等)。

## 登陆操作测试

1. 用户名和密码都符合要求（格式上的要求）；
2. 用户名和密码都不符合要求（格式上的要求）；
3. 用户名符合要求，密码不符合要求（格式上的要求）；
4. 密码符合要求，用户名不符合要求（格式上的要求）；
5. 用户名或密码为空；
6. 数据库中不存在的用户名，不存在的密码；
7. 数据库中存在的用户名，错误的密码；
8. 数据库中不存在的用户名，存在的密码；
9. 输入的数据前存在空格；
10. 输入正确的用户名密码以后按[enter]是否能登陆。

## 添加操作测试

1. 要添加的数据项均合理，检查数据库中是否添加了相应的数据；
2. 留出一个必填数据为空；
3. 按照边界值等价类设计测试用例的原则设计其他输入项的测试用例；
4. 不符合要求的地方要有错误提示；
5. 是否支持table键；
6. 按enter是否能保存；
7. 若提示不能保存，也要察看数据库里是否多了一条数据。

## 删除操作测试

1. 删除一个数据库中存在的数据，然后查看数据库中是否删除；
2. 删除一个数据库中并不存在的数据，看是否有错误提示，并且数据库中没有数据被删除；
3. 输入一个格式错误的数据，看是否有错误提示，并且数据库中没有数据被删除；
4. 输入的正确数据前加空格，看是否能正确删除数据；
5. 什么也不输入；
6. 是否指出table键；
7. 是否支持enter键。

## 查询操作测试

* 精确查询：

1. 输入的查询条件为数据库中存在的数据，看是否能正确地查出相应得数据
2. 输入正确的查询条件以前加上空格，看是否能正确地查出相应的数据
3. 输入格式或范围不符合要求的数据，看是否有错误提示
4. 输入数据库中不存在的数据
5. 不输入任何数据
6. 是否支持table键
7. 是否支持enter键

* 模糊查询：

1. 输入一些字符，看是否能查出数据库中所有的相关信息

## 软件加密测试

软件加密分软加密和硬加密，但无论哪种加密方式，都是从以下几点开展测试：

1. 软件试用期内，检查软件是否能正常运行
2. 软件试用期已过，检查软件是否能正常运行
3. 在使用期内的加密密钥，看软件是否能正常运行
4. 在已过期的加密密钥，看软件是否能正常运行
5. 永久型的加密密钥，通过修改使用时间，看软件是否存在过期现象
6. 错误的加密密钥，看软件是否能正常运行
7. 在多台PC机上，使用同一个加密密钥，看软件是否能正常运行
8. 在一台PC机上，使用多个加密密钥，看软件是否能正常运行

# 测试体系-绩效考核

**考核制度**

**目的：**为加强对研发人员的管理，提高研发人员的工作积极性和工作绩效，确保公司经营活动的顺利进行，特制定本制度。

**适用范围：**本制度适用于公司研发技术类测试岗员工考核

**职责分工：**研发测试岗人员考核工作职责分工如下表所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **考核对象** | **考核小组成员** | **考核指标** | **评分权重** | **审批人** | **组织者** | **考核频率** |
| 四级部门 | 研发管理中心 | 业绩指标 | 45% | 三级部门总经理 | 秘书处 | 季度 /年度 |
| 三级部门副总经理 | 通用指标 | 55% |
| 部门主管/测试项目经理 | 研发管理中心 | 业绩指标 | 50% | 四级部门经理 | 秘书处 | 季度 /年度 |
| 上级主管 | 通用指标 | 50% |
| 员工 | 测试项目经理 | 业绩指标 | 80% | 四级部门经理 | 秘书处 | 季度/年度 |
| 上级主管 | 通用指标 | 20% |
| 备注 | 秘书处是产管会绩效考核工作的管理部门，负责各部门的绩效考核培训、监督实施、结果统计与分析、申诉处理等工作 | | | | | |

**绩效考核工作要求：**

1. 目标要求明确、量化、可行。
2. 对绩效目标的完成情况要求定期评估、考核，并进行绩效面谈与辅导。
3. 绩效考核的结果要求定期公布。
4. 目标制定、绩效评估、绩效考核要经考评小组集体讨论通过，具体的面谈、辅导由直接主管负责一对一进行。
5. 参加评估、考核的人员，在结果未获得批准前，不准泄漏任何有关信息；结果获得批准后，不准泄漏讨论过程的任何信息。

**考核目标与内容的设定**

个人季度/年度考核对象：公司规定部门个人季度/年度考核对象为下列员工以外的部门所有员工。

1. 新入职员工；
2. 因其他特殊原因经公司批准可不参加年度考核的员工，如季度。

考核目标制定原则：

1. 以结果为导向原则，依据公司总体目标和部门目标层层分解，设立个人目标。
2. SMART原则，即目标要具体（specific）、可衡量（measurable）、可达到（attainable）、相关的（relevant）、有时间限定（time-based）。
3. 目标数量适中原则，目标不要太多，最多6～8个。

考核内容：研发中心体系测试岗考核内容请参照下表。

|  |  |  |
| --- | --- | --- |
| **考核对象** | **考核内容** | **考核频率** |
| 四级部门 | 1. 部门项目进度偏差率 2. 部门项目流程符合度 3. 部门项目有效bug率 4. 部门测试过程文档质量 5. 主动离职率 6. 部门协作度 7. 培训完成率 8. 制度及流程改造能力及创新能力 9. 部门员工敬业度 10. 团队核心素质能力 | 季度/年度 |
| 主管/项目测试经理 | 1. 项目流程符合度； 2. 项目进度偏差率； 3. 有效bug率； 4. 日常文档质量； 5. 专业技能、组织能力、沟通协调能力、执行力、解决问题能力。 | 季度/年度 |
| 员工 | 1. 日常文档质量 2. 项目工作进度； 3. 项目工作质量； 4. 执行力、工作态度、学习能力、专业技能。 | 季度/年度 |

**考核实施**

员工、主管考核实施流程：秘书处对研发主管和员工进行月度、半年度、年度考核，其考核流程如下。

1. 研发四级部门与秘书处根据部门实际工作要求，就当期主要工作任务、考核标准、指标权重等内容进行讨论，制定“绩效考核表”，交三级部门总经理批准后实施；
2. 秘书处向三级部门总经理、四级部门总监、其他主管、下属员工等发放研发中心体系主管的“绩效考核表”，向研发中心体系主管及其他同级同事发放研发中心体系员工的“绩效考核表”。
3. 相关人员根据研发中心体系主管、员工的工作表现填写考核表。
4. 秘书处汇总考核表，统计、计算考核得分及按要求划分考核等级，并将考核结果报中心体系三级部门总经理审批。
5. 秘书处将经审批后的考核结果向被考核者公开。
6. 被考核者与直接领导进行考核沟通，提出工作改进计划。

中心体系四级部门考核实施流程：秘书处对四级部门进行月度、半年度、年度考核，其考核流程如下。

1. 中心体系三级部门总经理与四级部门根据产管会经营计划和部门实际工作要求，就四级部门当期主要工作任务、考核标准、指标权重等内容进行讨论，共同制订“绩效考核表”，交秘书处审核，经研发中心体系主管领导批准后实施。
2. 秘书处向研发中心体系主管领导、中心体系三级部门、四级部门发放四级部门“绩效考核表”。
3. 相关人员根据四级部门的工作表现填写考核表。
4. 秘书处汇总考核表，统计、计算考核得分，并将考核结果报研发中心体系主管领导审批。
5. 秘书处将经审批后的考核结果向四级部门经理公开。
6. 研发中心体系主管领导与四级部门经理进行考核沟通，提出工作改进计划。

**考核结果管理与应用**

部门考核结果运用：部门绩效考核结果将作为个人季度、年度考核的内容之一，以不同的权重计入个人季度、年度考核结果，具体如下表所示。

|  |  |  |
| --- | --- | --- |
| 职级（研发中心体系） | 四级部门经理 | 员工 |
| 部门考核得分占个人考核得分权重 | 100% | 20% |

考核结果分级：根据考核成绩，秘书处将员工考核的结果分为4个等级，详细见下表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评分等级 | A+（优秀） | A（合格） | B（待改进） | C（不合格） |
| 占部门人员比例 | 20% | 65% | 15% | 5% |

部门人员考核结果运用：部门人员考核结果将影响员工的岗位晋升与降级、段位等级升降、奖金分配、培训机会等。

绩效改进: 对于考核结果排在后5%的员工，公司需与其签订下一季度绩效目标，要求其限期改进绩效；对于连续两次考核排名在后5%的员工，公司将予以辞退。

季度奖金计发：奖金基数参照人力资源部制定的部门奖金管理制度而定，绩效系数根据考核结果确定，具体见下表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 等级说明 | A+（优秀） | A（良好） | B（基本称职） | C（不合格） |
| 员工绩效系数 | 1.4 | 1 | 0.6 | 0 |
| 部门绩效系数 | 2.4 | 1.6 | 1.0 | 0 |

段位等级评价标准参与如下附件



**考核申诉管理**

* 考核申诉处理部门

（1）公司人力资源部是员工考核申诉的日常管理部门，被考核人如对考核结果不清楚或者持有异议，可填写“考核申诉表”向人力资源部提出申诉。

* 考核申诉时间

（1）员工个人对考核结果有意见的，在得知考核结果后七个工作日内，可向再上一级领导提出申诉。

（2）员工申诉超过申诉时间期限的，公司将不予受理。

* 申诉流程

（1）员工个人对考核结果有意见，填写“考核申诉表”，申诉表的内容应包括申诉人姓名、部门、申诉事由等。

（2）秘书处接到申诉后，须在五个工作日内做出是否受理的答复；对于申诉事项无客观事实依据而仅凭主观臆断的申诉不予受理。

（3）受理的申诉事件，首先由秘书处对申诉内容进行调查，然后与研发中心体系总经理进行协调、沟通。不能协调的，报研发中心体系主管领导处理。

公司董事长拥有申诉的最后决定权，各相关人员须按照其指示进行处理。

# 测试体系-FAQ

### 什么是“软件测试“？

软件测试就是在受控制的条件下对系统或应用程序进行操作并评价操作的结果。也就是说，如果用户面对着应用程序的 A 界面，在使用硬件 B 的时候做 C 操作，那么 D 结果应该出现。所谓受控制的条件应该包括正常条件和非正常条件。应该故意地去促使错误的发生，也就是事情在不该出现的时候出现或者在应该出现的时候没有出现。从本质上说，软件测试是“探测”。

在如何负责质量保障和软件测试的责任方面，各个机构有不同的做法。有时候，由一个小组或者个人来负责。常见的办法是项目组包括了测试人员和开发人员，他们在一起合作工作，由项目负责人来对质量保障进行总负责。这取决于该机构的大小和该机构的商务结构。

### 为什么软件会有毛病？

1. 交流错误或者没有进行交流──指出应用软件应当做什么和不应当做什么 (针对该应用软件的用户需求)。
2. 软件的复杂性──现在的应用软件十分复杂，以至于在当前的软件开发环境下，缺乏经验人不可能理解它。窗口类型的界面、客户／服务器和分布式的应用、数据通信、巨大的关系数据库、以及应用软件规模庞大，所有这一切都造成软件/系统的复杂性呈指数型的增长。除非进行仔细的设计，使用面向目标的技术会使得项目更加复杂而不是简单。
3. 编程错误──和所有的人一样，程序员也会犯错误。
4. 需求改变──客户恐怕不明白改变需求的影响，也许是知道但依然需要变更──会导致重新设计、重订工程进度表、对其他项目的影响、已完成的工作需要重做或者放弃、对硬件需求的影响等等。如果在项目中出现许多小的改变或一个大的改变，在项目各部分中出现已知或未知的相关的问题，可能会相互影响并导致出现问题。而且，不断地变更也会增加软件的复杂性，可能会导致错误的出现。这样就会影响技术人员的积极性。在一些快速变化的商业环境里，持续变更需求的影响是致命的。在这种情况下，管理者必须知道它的危险性。质量保障和测试工程师必须与此相适应，并安排持续的广泛的测试，以克服不可避免产生的问题。
5. 时间压力──因为有许多猜测成分，软件开发项目的进度很难安排得理想。当最后期限快到的时候，压力逐渐增大，错误随之产生。
6. 自负──人们通常喜欢这样说：

* “没问题”
* “简单”
* “我用几个小时就能完成”
* “修改老代码应该是简单的事情”

而不是说：

* “那个新增加的工作很复杂，可能会导致出现许多错误。”
* “我们没有把握，但我们会尽力。”
* “我在实际检查之前无法估计需要多长时间完成。”
* “我们一下子还不能估计旧的代码是如何的。”

如果有太多的“没问题”，必然会出问题。

1. 代码的文档质量差──糟糕的维护和修改，或者写得很差的文档，也会导致错误。在许多企业，管理人员并不鼓励程序员把代码写得清楚、易读。事实上，恰恰相反，他们主要注意尽快完成编码，认为如果别人不明白代码，就会更安全一点 (“难写的代码必然难读”)。
2. 软件开发工具──视觉工具、类库、编译器、编辑工具等等，都会把本身的故障引入或者产生质量差的文档，从而导致故障的产生。

### 怎么对WEB站点进行测试？

Web 站点也是一种客户/服务器类型的应用──有 Web 服务器和客户“浏览器”。应当考虑在 html页面、TCP/IP 通信、网间联结、防火墙、在 Web 页面运行的应用程序 (例如 java applets、java 描述语言、插入应用)、以及在服务器一端运行的应用程序 (例如 cgi 描述语言、数据库接口、日志应用软件、动态页面生成器、等等)。此外，有各种各样的服务器和浏览器，它们的版本也各不相同，相互之间的差别有小有大，链路的速度也各不相同，加上快速变化的技术以及各种标准和协议，都应当加以考虑。最终的结果是对 web 站点的测试成为一个主要的持续的努力过程。其他应当考虑的问题有：

1. 期望加在服务器上的负载 (即单位时间内点击的次数)，以及在此负载下所需要的性能 (例如 web 服务器的响应时间、数据库查询响应时间)。还要考虑使用何种工具来进行测试 (例如 web 负载测试工具、其他已有的合适的工具、web 自动下载工具，等等)。
2. 谁是最终的用户? 他们会使用何种浏览器? 它们所用的链接类型和速度怎样? 它们是单位内的用户 (可能使用高速链接和类似的浏览器) ，还是因特网上的用户 (使用不同的链接速度和浏览器)?
3. 在客户端所期望的特性 (即页面出现速度要多快，动画速度要多快，是否能装入并运行applets)?
4. 在进行维护/升级时是否允许停机? 多长时间?
5. 使用何种安全措施 (防火墙、密码、口令等)? 对它们有何期望? 它们可被检查吗?
6. 对站点与因特网链接的可靠性的要求是怎样的? 对系统备份、冗余链接以及测试有何影响?
7. 管理站点内容的升级需要那些步骤? 为维护、跟踪、和控制页面内容、图形、链接等，需要做什么?
8. 使用那种 HTML 规范? 严格程度如何? 允许用多少种浏览器?
9. 是否有标准或要求规定页面外观? 或者有适用于站点的全部或部分图形的标准?
10. 如何对内部和外部链接确认和升级? 频度如何?
11. 是否能对生产系统进行测试? 是否需要对系统单独测试? 浏览器的高速缓存、浏览器的选项变化、拨号链接的变化、以及在因特网上的“拥塞”问题等，是否在测试中都考虑到了?
12. 服务器的日志和报告是怎样扩展，以适应用户要求的? 是否把它们也当作一个系统部件? 它们是否需要测试?
13. 对 cgi 程序、applets、java 描述语言、ActiveX 等组件如何进行维护、跟踪、控制和测试?
14. 以下是一些有用的指导方针, 这仅仅是主观看法，在一定情况下，也许适用，也许不适用：
15. 页面最大为 3-5 帧，除非全部内容都是针对一个题目且不可分的。如果内容太多，需要在页面内建立链接。
16. 一个站点全部页面的版面和元素布局风格应当一致，使得用户知道自己仍在同一站点内。
17. 应当使页面尽量与浏览器相独立，否则应当提供相应的浏览器。
18. 所有的页面都应有指向页面以外的链接，不应当出现页面的死胡同。
19. 页面的主人、修改日期、与机构里有关人员的联系等内容，都应包括在每个页面内。

### 客户/服务器环境如何影响测试？

客户/服务器应用软件是相当复杂的，这是因为在客户、数据通信、硬件、以及服务器之间是互相依赖的。从而使对测试的要求也扩大了。当时间紧张的时候 (这是通常的情况)，重点应当放在集成测试和系统测试上。此外，进行负载/压力/性能测试，也有助于了解客户/服务器应用软件的局限和能力。市场上有一些工具可用来评估这类测试。

### 在应用软件中出现了需求中所没有的功能怎么办?

应当做出认真的努力，来确定在应用软件中是否存在大量未曾期望的或隐藏的功能，它可能表明在软件开发过程中存在着深层次的问题。如果该功能不是该应用软件的目标所必需的，应当把它删除，因为可能会有设计者或客户未曾考虑到的未知的影响或派生问题 (dependencies)。如果不删除，就应增加设计内容，以便据此增加测试或回归测试。管理人员应当认识到，计划以外的功能必然导致风险的增加。如果计划以外的功能只是为了改善用户界面，其风险不会太大。

### 如果需求一直在变怎么办？

这是一个常见的令人头疼的问题。

1. 如果可能，尽早与承担该项目风险的人接触，以便了解需求会怎样改变，从而可以尽早地改变测试计划和策略。
2. 如果在对应用程序进行初始设计时多考虑一些适应性，那么以后在发生需求的改变时，就不需要再为改变做很多事情了。
3. 好的代码注释和好的文档有助于开发人员做出相应的改变。
4. 只要有可能，就应使用快速原型，以帮助用户确认他们的需求，从而减少变更。
5. 在项目的时间表中应当留出余量，以应付可能出现的变更。
6. 尽量把新的需求纳入应用软件的“下一版”，而把原始需求作为“第一版”。
7. 通过谈判，把易于实现的新的变更列入项目，而把难于实现的新需求列入该应用软件的以后的版本。
8. 要确保让客户和管理人员了解变更对进度表的影响、所带来的风险、以及因变更所引起的大量资金消耗。
9. 在应付改变时，应在为建立自动测试而作的努力和重新进行测试所做的努力之间取得平衡。
10. 在设计自动测试脚本时，试图使其有一些灵活性。
11. 在对应用软件进行自动测试时，要把注意力集中在看来不大会改变的部分。
12. 对变更进行适当的风险分析，以减少回归测试的要求。
13. 在设计测试用例时要有一定的灵活性。做到这一点并不容易，所以要降低测试用例的详细程度，或者只建立高级的通用型的测试计划。
14. 少注意详细的测试计划和测试用例，要把重点放在专门的测试上。

### 如果时间不够，无法进行充分的测试怎么办?

使用风险分析，确定测试的重点。

由于很少有机会对一个应用软件进行所有可能的测试 (包括所有可能的事件组合、所有的相关性、或者一切可能出错的东西)，对大多数软件开发项目来说，利用风险分析是适当的。这需要判断技能、常识、感觉和经验。如果有正当理由，也可采用正式的方法。需要考虑下列因素：

1. 对于该项目的用途而言，哪种功能最重要?
2. 哪种功能对用户最明显?
3. 哪种功能对安全影响最大?
4. 哪种功能对用户最有用?
5. 对客户来说，该应用软件的哪个部分最重要?
6. 在开发过程中，该应用软件的哪个部分可以最先测试?
7. 哪一部分代码最复杂，容易导致出现错误?
8. 哪一部分的应用程序是在急迫或在惊恐的情况下开发出来的?
9. 哪一部分程序与过去项目中引起问题的部分相类似/有关?
10. 哪一部分程序与过去项目中需要大量维护的部分相类似/有关?
11. 需求和设计的那些部分不清楚或不容易读?
12. 开发人员认为在应用软件中哪些部分是高风险的?
13. 哪些问题能造成最差的发行?
14. 哪些问题最能引起用户抱怨?
15. 哪些测试可以容易地覆盖多种功能?
16. 哪些测试在覆盖高风险部分的测试时使用时间最少?

### 什么时候停止测试？

很难说。许多大型应用软件非常复杂，又被用于相互依赖的环境中，进行完全的测试是根本不可能的。一般根据下列因素来决定何时停止测试：

1. 截止日期 (发布截止日期、测试截止日期、等等)
2. 一定比例的测试用例已经完成
3. 测试预算耗尽
4. 代码/功能/需求已经达到指定的程度
5. 错误已低于一定比例
6. β或α测试期结束

### 遇到软件中错误太多，以至完全无法进行测试时怎么办?

在这种情况下，测试人员最好仔细研究报告，看错误或阻塞型的问题最初出现的地方，并把注意力放在最严重的错误上。由于这类问题会影响进度，并能反映出软件开发过程中的深层次的问题 (例如，部件测试或集成测试不充分、设计不良、不适当的建立和发布过程，等等)，所以应当通知管理人员，并向他们提供一些证明文件。

### 在发现错误之后应当做什么?

错误应当被通报并分配给开发人员，以便进行改正。在问题解决之后，应当对被修改的部分重新进行测试。而且应当根据情况进行回归测试，以检查被修改的部分是否在其他部分引出新的问题。如果使用了问题跟踪系统，可以缩短这些过程。商业市场上有许多问题跟踪/管理的软件工具可供选择。

下面是一些在问题跟踪过程中应当考虑的内容：

1. 整理出有关错误的完整的信息，以便使开发人员能够了解错误、知道错误是如何起作用的。如果有必要，可以再现错误。
2. 应用软件的名称或识别符及版本
3. 错误出现时的情况：功能、模块、特征、目标、屏幕等
4. 环境说明、系统、平台、有关硬件说明
5. 测试用例的名称/编号/识别符
6. 简短的一句话的错误描述
7. 详细的错误描述
8. 如果所产生的错误不在测试用例之内，或者开发人员不太容易了解测试用例/测试场景/测试工具，就要描述使错误重现的操作步骤。
9. 在测试中所使用的文件名/或文件的说明/数据/消息/等等
10. 文件引用/错误消息/日志文件的引用/屏幕情况/测试工具日志，它们有助于发现问题的原因
11. 问题严重的程度 (通常分为“致命”、“严重”、“一般”、“提示/建议”)
12. 错误是否可再现?
13. 测试人员的名字
14. 测试日期
15. 报告错误的日期
16. 与问题有关的开发人员/开发组/部门的名字
17. 描述问题的起因
18. 修改过的代码段/文件/模块/类别/方法
19. 修改日期
20. 包含修改内容的应用软件的版本
21. 回归测试需求
22. 负责回归测试的测试人员
23. 回归测试的结果

在各个阶段，都应把问题跟踪过程的报告送给有关人员。例如，测试人员应当知道何时应进行再测试，开发人员应当知道错误何时发现并知道到何处能找到有关信息，而管理人员则应具备生成报告/总结的能力。

### 什么是测试用例？

测试用例是一份文档，它描述了一个输入、输出或者是与其相应的预期的响应，以便来判断应用软件的工作是否正常。测试用例应当包括测试标识、测试用例的名称、预置条件、输入数据要求、步骤以及预期的结果。

注：开发一个应用软件的测试用例的过程，需要全面、深入地考虑该软件的操作，所以有助于发现在其需求或设计里面的问题。因此，如果有可能，在开发周期中应当尽早准备测试用例。

### 什么是测试计划？

一个软件项目的测试计划是一份文件，它说明了在进行软件测试过程中的目标、范围、方法、和重点。在准备测试计划的过程中，要通盘考虑在接受一个软件产品之前所必需进行的一切验证工作。完整的文档将帮助测试组以外的人们了解“为什么”和“如何”验证产品。文档应当是十分完全的，但注意不要太繁杂，以至测试组以外的人不愿去读它。以下是一些应当包括在测试计划中的内容，它们依赖于特定的项目：

1. 标题
2. 软件标识，包括版本号
3. 文档修改历史，包括作者、日期、批准人
4. 内容目录
5. 文档的用途、所面向的读者
6. 测试目标
7. 软件产品概况
8. 有关的文件清单，例如需求、设计文档、其他测试计划等
9. 有关的协议名称和协议的标识
10. 全部软件项目组织情况和人员/联系所需的信息/责任
11. 测试组织的情况和人员/联系所需的信息/责任
12. 假定和相关
13. 项目风险分析
14. 测试优先级和重点
15. 测试范围和限制
16. 测试环境──硬件、操作系统、必需的其它软件、数据结构、与其他系统之间的接口
17. 对在测试中用于跟踪错误原因或源代码的专用软件和硬件工具进行讨论
18. 测试自动化──判断和回顾
19. 所使用的测试工具，包括版本、路径等
20. 测试脚本/测试代码的维护过程和版本控制
21. 问题跟踪和解决──工具和过程
22. 项目中所用的测试方法
23. 软件的进入和退出的条件
24. 初始的健全测试周期和条件
25. 测试中止和重新启动的条件
26. 人员分配
27. 人员预先培训的需求
28. 测试地点
29. 外部测试机构的利用及其目的、责任、可交付使用、联系人、及配合
30. 附录──术语表、缩写等

### 软件测试包括哪些内容？

以下是一些需要考虑的步骤：

1. 得到需求、功能设计、内部设计说明书和其他必要的文档
2. 得到预算和进度要求
3. 确定与项目有关的人员和他们的责任、对报告的要求、所需的标准和过程 (例如发行过程、变更过程、等等)
4. 确定应用软件的高风险范围，建立优先级、确定测试所涉及的范围和限制
5. 确定测试的步骤和方法：单元、集成、功能、系统、负载、可用性、兼容性等各种测试
6. 确定对测试环境的要求 (硬件、软件、通信等)
7. 确定所需的测试工具，包括记录/回放工具、覆盖分析、测试跟踪、问题/错误跟踪等等
8. 确定对测试的输入数据的要求
9. 分配任务和任务负责人，以及所需的测试人力
10. 设立大致的时间表、期限和里程碑
11. 确定输入环境的类别、边界值分析、错误类别
12. 准备测试计划文档和对计划进行必要的Review
13. 对测试用例进行必要的Review
14. 准备测试环境和测试工具，得到必需的用户手册/参考文件/结构指南/安装指南，建立测试跟踪过程，建立日志和档案、建立或得到测试输入数据
15. 得到并安装软件版本
16. 进行测试
17. 评估和报告结果
18. 跟踪问题/错误，并解决它
19. 如果有必要，重新进行测试
20. 在整个生命周期维护和修改测试计划、测试用例、测试环境和测试工具

### 测试也有版本控制吗？

这里的版本主要是指测试对象的版本控制，也就是指对开发部提交的产品进行版本控制。在开发小组版本管理不规范的情况下，测试小组进行版本控制十分重要，要保证测试对象是可以控制的。建议开发和测试双方进行明确的约定，可以各自指定专门的测试版本负责人，制定提交原则，对提交情况进行详细的记录，这样基本避免了版本失控导致的测试失误或无效。

### 其它注意事项

* 系统测试活动启动后，需要输出进度周报，包括本周活动开展情况和下周工作计划、对测试活动开展的四点分析（难点、关键点、风险点、问题点）和二点总结（亮点、不足点），发送给项目相关干系人，项目干系人包括但不仅于产品经理、项目经理、测试项目经理及测试项目组成员；
* 项目立项后，测试负责人需要全程参与项目各活动阶段，包括项目过程交付件的评审，参与里程碑的决策，同时督促提醒将结论形成纪要发送人项目干系人，同时将会签后的结论归档入库；
* 系统规格说明书基线化后是进行测试需求分析与设计的前提条件之一，测试活动中的所有交付件需要组织项目干系人进行评审，项目干系人包括但不限于产品经理、项目经理、测试项目经理及开发、测试骨干成员；
* 测试项目经理及测试骨干成员需要参加项目例会及各类讨论会，并督促提醒将结论形成纪要发送人项目干系人，同时将会签后的结论归档入库；
* 测试过程中遇到技术方面的短板需要及时向直属主管、项目经理、测试项目经理反馈，以寻求外部的帮助；
* 测试活动过程中的风险与问题，需要指定专人跟进，直至问题得到闭环处理。