**蜗牛Boss系统**

**软件测试计划**

**第一组**

**组长：薛博城**

**组员：杜越新、邓寓、王琪**

目录

[1前言 1](#_Toc34923358)

[1.1 项目背景 1](#_Toc34923359)

[1.2 编写目的 1](#_Toc34923360)

[1.3 功能实现 1](#_Toc34923361)

[1.4 名次解释 2](#_Toc34923362)

[1.5 参考资料 2](#_Toc34923363)

[1.6 测试摘要 2](#_Toc34923364)

[2. 资源需求 4](#_Toc34923365)

[2.1 硬件资源 4](#_Toc34923366)

[2.2 软件资源 4](#_Toc34923367)

[2.3 人力资源 4](#_Toc34923368)

[3. 测试详述 6](#_Toc34923369)

[3.1 测试范围 6](#_Toc34923370)

[3.2 测试目标 6](#_Toc34923371)

[3.3 组织形式 6](#_Toc34923372)

[3.4 风险和约束 6](#_Toc34923373)

[3.4.1 主要风险和应对措施 6](#_Toc34923374)

[3.4.2 风险评估 7](#_Toc34923375)

[3.4.3 风险管理 7](#_Toc34923376)

[3.5 测试通过/失败标准 8](#_Toc34923377)

[3.6 测试挂起/恢复条件 8](#_Toc34923378)

[3.7 测试进度 8](#_Toc34923379)

[4. 测试策略 10](#_Toc34923380)

[4.1 整体策略 10](#_Toc34923381)

[4.2 测试类型 10](#_Toc34923382)

[4.3 测试技术 10](#_Toc34923383)

[5. 测试提交文档 11](#_Toc34923384)

[6. 质量目标 11](#_Toc34923385)

[7. 计划审核记录 12](#_Toc34923386)

# 1前言

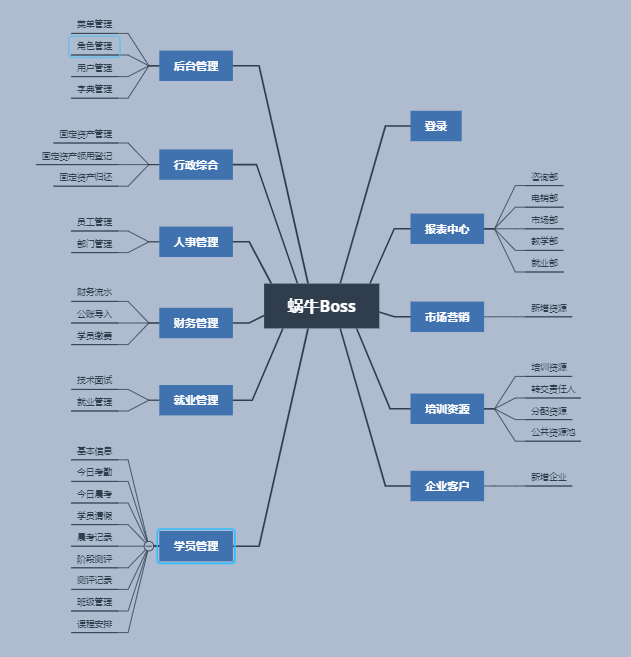
## 项目背景

蜗牛boss管理系统是由蜗牛学院针对教学资源管理、员工管理、学员管理、学员就业管理、财务管理及整个系统正常运转而设计的内部管理系统，是一个可以运行在Windows 或者 Linux 系统服务器上的 B/S 架构的应用软件。

## 编写目的

此测试计划目的是在系统完成编码前，让测试组相关人员对项目功能结构进行了解。对测试组成员人力配置、软硬件环境配置、测试质量、测试进度以及测试人员完成项目的方法和质量评估做出明确的指导！

## 功能实现



## 名次解释

|  |  |  |  |
| --- | --- | --- | --- |
| **术语或缩写词** | **英文解释** | **中文解释** | |
| 缺陷级别 | Defect level | 一级缺陷：遗漏需求，系统崩溃，数据错误  二级缺陷：正向验证未通过，导致接口错误。导致需要重新安装或重新启动该软件  三级缺陷：反向的验证未通过，导致接口错误  四级缺陷：存在某些细微的缺陷，但不影响程序正常应用。 |
| 功能测试 | Functional testing | 根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。 | |
| 兼容性测试 | Compatibility Testing | 测试软件是否和系统的其它与之交互的元素之间兼容，如：浏览器、操作系统、硬件等。验证测试对象在不同的软件和硬件配置中的运行情况。 | |
| 可用性测试 | **usability testing** | 检验其是否达到可用性标准。 | |

## 参考资料

|  |  |  |
| --- | --- | --- |
| **资料名称** | **作者** | **说明** |
| 蜗牛boss2.5版本操作手册 | 蜗牛学院 | 系统操作的说明和指导 |
| 软件需求列表 | 薛博城、杜越新、  邓寓、王琪 | 客户需求的整理和分类，是设计测试用例的基础 |
| 蜗牛boss需求规格说明书 | 蜗牛学院 | 隐性需求的说明和分析 |
| 角色账号信息参考文档 | 蜗牛学院 | 不同角色的登录信息 |

## 测试摘要

**重点事项：**

1、B/S架构的软件系统，完成对该系统的功能及兼容性测试，编写最后的测试总结报告以提供对该系统的质量验证

2、不同主流浏览器是否都能够兼容

3、一般影响系统要求或基本功能的实现，反向验证未通过

4、时间进度：测试在3月9日开始，4月18日结束。

5、测试目标：完成对该系统的功能及可用性及兼容性测试，确保系统功能正常运行，支持各种软硬件对系统的支持！

**争议事项：**

对任务的评估时间远大于给定的测试时间以及需求不明确

**风险评估：**

1、质量需求或产品的特性理解不准确，造成测试范围分析的误差，结果某些地方始终测试不到或验证的标准不对;

2、测试用例没有得到百分之百的执行，如有些测试用例被有意或无意的遗漏;

3、需求的临时/突然变化，导致设计的修改和代码的重写，测试时间不够;

4、质量标准不都是很清晰的，如适用性的测试，仁者见仁、智者见智;

5、测试用例设计不到位，忽视了一些边界条件、深层次的逻辑、用户场景等;

6、测试环境，一般不可能和实际运行环境完全一致，造成测试结果的误差;

7、有些缺陷出现频率不是百分之百，不容易被发现;如果代码质量差，软件缺陷很多，被漏检的缺陷可能性就大;

8、回归测试一般不运行全部测试用例，是有选择性的执行，必然带来风险。

9、测试人员出现人员变动，带来的风险。

# 资源需求

## 硬件资源

|  |  |  |  |
| --- | --- | --- | --- |
| **IP/机型** | **操作系统** | **用途说明** | **预计空间** |
| 192.168.0.102 | Win8 | 服务端 | 1G |
| 192.168. 6.181 | Win10 | 客户端 | 4G |
| 192.168.6.266 | Win7 | 4G |
| 192.168.6.125 | Win7 | 4G |
| 网络 |  | 局域网 |  |

## 软件资源

|  |  |  |
| --- | --- | --- |
| **软件类型** | **软件名称** | **用途说明** |
| 服务器端 | VMware Workstation Pro | 用于安装SentOS6.5 |
| CentOS6.5 | 软件运行环境搭建 |
| Java Runtime Environment（JRE）运行环境：1.8 以上 | 软件运行环境搭建 |
| Tomcat 9.0.8 以上 | 软件运行环境搭建 |
| MySQL 数据库：5.1 以上 | 软件运行环境搭建 |
| 客户端 | IE 11.0 | 运行蜗牛boss系统 |
| Firefox 71.0（32 位） | 运行蜗牛boss系统 |
| Chrome 78.0.3904（64 位） | 运行蜗牛boss系统 |
| Git-2.25.1-64位 | 版本控制管理 |
| 禅道10.1 以上 | 缺陷管理 |
| Win8中文版8.1 Service Pack 1 64位 | 个人用户版本windows系统 |

## 人力资源

|  |  |  |
| --- | --- | --- |
| **角色** | **姓名** | **职责** |
| 测试组长 | 薛博城 | 负责协调和分配，测试培训资源模块,主导测试计划、测试报告和文档评审，书写对应模块测试需求列表和测试用例 |
| 组员 | 杜越新 | 负责测试财务管理、人事管理、行政综合和后台管理模块，参与测试计划、测试报告和文档评审，书写对应模块测试需求列表和测试用例 |
| 组员 | 邓寓 | 负责测试登录、报表中心、市场营销企业客户和就业管理模块，参与测试计划、测试报告和文档评审，书写对应模块测试需求列表和测试用例 |
| 组员 | 王琪 | 负责测试销售报表模块，参与测试计划、测试报告和文档评审，书写对应模块测试需求列表和测试用例 |

# 测试详述

## 测试范围

在 Windows系统下，对系统各模块进行功能测试，可用性测试以及兼容性测试，在主流浏览器（包括ie11.0、chrome和firefox的最新版本）下是否可以正常使用。

## 测试目标

保证系统交付用户前无一、二级别BUG,遗留的三级缺陷不超过所有缺陷的5%，尽可能减少影响用户使用的BUG。

## 组织形式

以小组的形式完成测试活动，由小组组长组织、协调、分配组员工作，在完成相应模块测试工作后，分配新版本的分配任务。

组织小组完成项目测试评审工作！

## 风险和约束

### 主要风险和应对措施

在测试工作中，主要的风险表现有以下几点：

（1）需求风险。需求人员提供的SRS没有严格的输入输出条件，导致对需求的理解可能有偏差。应对措施是及时与需求人员沟通。

（2）测试用例风险。由于测试人员进行用例设计的经验不足，导致可能会忽略一些边界条件，或者遗漏常见缺陷，应对措施是对测试人员进行培训学习，强化个人技能。

（3）缺陷风险。某些缺陷偶发，难以重现，容易被遗漏。应对措施是及时记录频率缺陷赋以图片或视频证明。

（4）代码质量风险。软件代码质量差，导致缺陷较多，容易出现测试的遗漏。应对措施是提前进行代码走读并对代码进行评审。

（5）测试环境风险。有些情况下测试环境与生产环境不能完全一致，导致测试结果存在误差。应对措施是设计的测试场景接近真实场景。

（6）测试技术风险。某些项目存在技术难度，测试能力和水平导致测试进展缓慢，项目延期。应对措施是对测试人员进行技术培训，开展项目研讨会。

（7）回归测试风险。回归测试一般不运行全部测试用例，可能存在测试不完全。应对措施是在缺陷常发模块以及缺陷等级高的部分重点测试。

（8）沟通协调风险。测试过程中存在不同人员之间的沟通、协作，难免存在误解、沟通不畅的情况，导致项目延期。应对措施是秉承有借有还，再借不难的原则。

（9）进度风险：时间安排和个人进度会影响项目的整体进度。应对措施是加班、借人，与上级汇报情况。

（10）其它不可预计风险。一些突发状况、不可抗力等也构成风险因素，且难以预估和避免。

以上是测试过程中可能发生的风险，其中有的风险是难以避免的，如缺陷风险等。有的风险从理论上可以避免，但实际操作过程中出于时间和成本的考虑，也难以完全回避，如回归测试风险等。对于难以避免的风险，我们的目标是将风险降到最低水平。

### 风险评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 风险 | 严重程度（高3中2低1） | 发生可能性（极高4、高3、一般2、很小1） | 分数=严重程度\*可能性 | 风险级别 |
| 需求的理解偏差风险 | 2 | 3 | 6 | 二级 |
| 测试用例设计遗漏风险 | 3 | 3 | 9 | 一级 |
| 缺陷风险 | 2 | 2 | 4 | 三级 |
| 回归测试风险 | 2 | 3 | 6 | 二级 |
| 沟通协调风险 | 1 | 2 | 2 | 四级 |

### 风险管理

一级风险：测试用例设计遗漏风险需要每日进行检查：

二级风险：需求的理解偏差风险，回归测试风险每周进行检查；

三级风险：缺陷风险、进度风险、人员风险等其他风险不定时进行检查等

四级风险：沟通协调风险不定时进行汇报检查

## 测试通过/失败标准

通过标准：

1、基本流程能够通畅的完成，核心功能以及辅助功能可以体现。

2、主要应该体现业务功能和其辅助功能可以实现

3、基本界面符合术语规范，不存在错误或明显歧义；所有可使用的流程中的界面 设计工作必须完成；

4、按照标准流程没有出现各种内部错误，包括数据库异常和代码异常；

5、所有报表能够在基本数据的基础上正确生成；

6、计划测试时间用完了；

7、领导或用户授权允许存在已知部分问题上线。

失败标准：

冒烟测试失败，业务功能无法完成，标准流程内出现各种错误，超出测试工作完成的时间限制。

## 测试挂起/恢复条件

常见的挂起（正常执行时有突发情况不能正常使用）准则有：

1、冒烟测试不通过，导致无法进行后续测试，

2、人员缺员严重导致无法执行测试过程

3、人员被调用至其他项目

4、技术无法达到测试要求

恢复条件：测试恢复的条件是当将引起挂起的测试用例重新测试通过之后，可以恢复测试。

## 测试进度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试阶段** | **开始时间** | **结束时间** | **资源** | **是否里程碑** |
| 生成测试需求文档 | 3月9日 | 3月11日 | 薛博城、杜越新、  邓寓、王琪 | 是 |
| 制定测试计划 | 3月11日 | 3月12日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 对测试计划进行评审 | 3月12日 | 3月12日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 设计测试方案 | 3月13日 | 3月16日 | 薛博城、杜越新、  邓寓、王琪 | 是 |
| 对测试方案进行评审 | 3月16日 | 3月17日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 设计测试用例 | 3月17日 | 3月20日 | 薛博城、杜越新、  邓寓、王琪 | 是 |
| 1. 对测试用例进行评审 2. 配置测试软件的环境 | 3月21日 | 3月21日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 1.执行手工测试用例并提交bug至禅道  2.编写系统测试总结报告并评审 | 3月21日 | 3月25日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 1.整理好系列文档并提交至git | 3月25日 | 3月25日 | 薛博城、杜越新、  邓寓、王琪 | 是 |
| 1.编写GUI测试框架  2.编写接口测试框架 | 3月25日 | 3月29日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 完善2.5版本测试用例、测试方案、测试报告 | 3月30日 | 3月30日 | 薛博城、杜越新、  邓寓、王琪 | 是 |
| 1.执行2.5版本GUI测试2. 编写GUI测试报告 | 3月31日 | 4月5日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 1. 执行2.5版本接口测试2.编写接口测试报告 | 4月6日 | 4月9日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 1.编写2.5版本性能测试方案2.执行2.5版本性能测试 | 4月10日 | 4月13日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 编写性能测试报告 | 4月14日 | 4月14日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 编写蜗牛性能测试总结书 | 4月15日 | 4月15日 | 薛博城、杜越新、  邓寓、王琪 |  |
| 整理好系列文档并提交至git | 4月16日 | 4月17日 | 薛博城、杜越新、  邓寓、王琪 |  |

# 测试策略

## 整体策略

1、采用边界值法，必要时使用等价类划分方法补充一定数量的测试用例;

2、对照程序逻辑，检查已设计出的测试用例的逻辑覆盖程度，看是否达到了要求;

3、如果程序功能规格说明中含有输入条件的组合情况，则可以选择正交试验

## 测试类型

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **测试类型** | **说明** | **是否采用** |
| **1** | 功能测试 | 根据需求文档、设计文档等检查产品是否正确实现了功能。 | ***√*** |
| **2** | 流程测试 | 按操作流程进行的测试，主要有业务流程、数据流程、逻辑流程、正反流程，检查软件在按流程操作时是否能够正确处理 | ***√*** |
| **3** | 界面测试 | 检查界面是否符合公司界面规范，是否美观合理 | ***√*** |
| **4** | 易用性测试 | 检查系统是否易用友好，是否符合通用的操作习惯 |  |
| **5** | 接口测试 | 检查系统能否与外部系统或外部设备等是否接口正常 | ***√*** |
| **6** | 安装测试 | 检查系统能否正确安装、配置基础数据是否正确 |  |
| **7** | 性能测试 | 提取系统性能数据，检查系统是否满足在需求中所规定达到的性能。 | ***√*** |
| **8** | 安全性测试 | 检查系统安全，是否达到安全需求，是否存安全隐患 |  |
| **9** | 兼容性测试 | 对于 B/S 架构的系统来说需要考虑用户端浏览器的版本。主要对IE，Firefox，Chrome这三个主流浏览器进行兼容性测试 | ***√*** |

## 测试技术

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **测试技术** | **说明** | **是否采用** |
| **1** | 测试用例设计 | 在产品需求评审通过后编写测试用例 | ***√*** |
| **2** | 黑盒测试 | 边界值法，等价类法，错误猜测法，流程分析法 | ***√*** |
| **3** | UI自动化测试 | 使用webdriver与unittest框架进行测试 | ***√*** |
| **4** | 性能测试 | 使用JMeter工具进行性能测试 | ***√*** |
| **5** | 接口测试 | 使用requests库和unittest框架进行测试 | ***√*** |

# 测试提交文档

|  |  |  |
| --- | --- | --- |
| **文档说明** | **作者** | **文档位置(配置库)** |
| 系统测试计划 | 薛博城、杜越新、  邓寓、王琪 |  |
| 测试需求列表 |  |
| 测试用例 |  |
| 测试缺陷报告 | 禅道 |
| 系统测试报告 |  |
| 工作日报 |  |

# 质量目标

|  |  |  |
| --- | --- | --- |
| **编写** | **测试质量目标** | **确认人以及特殊说明** |
| **1** | 测试已实现的产品是否达到设计的要求，包括：各个功能点是否已实现，业务流程是否正确 | QA（王老师） |
| **2** | 所有的测试用例已经执行过 | QA（王老师） |
| **3** | 不允许存在一级和二级缺陷 | QA（王老师） |
| **4** | 缺陷的发现速率正在下降并接近0 | QA（王老师） |
| **5** | 在最后的半天没有发现一级和二级的缺陷 | QA（王老师） |

# 计划审核记录

|  |
| --- |
| QA&CM审核意见：  签名： 日期： |
| 质管部经理审核意见：  签名： 日期： |
| 项目经理审核意见：  签名： 日期： |