**圣剑纹章**

**测试规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号： | v1.0 | 文档编号： | doc2\_20180108 |
| 文档密级： | 0 | 项目名： | 圣剑纹章 |
| 编写人： | 开发项目组 | 编写日期： | 2018-01-08 |

目录

[一、引言 2](#_Toc503296473)

[**1.1编写目的：** 2](#_Toc503296474)

[**1.2范围：** 3](#_Toc503296475)

[**1.3职责：** 3](#_Toc503296476)

[二、工作流程 3](#_Toc503296477)

[**2.1测试依据：** 3](#_Toc503296478)

[**2.2制订《测试规范》：** 3](#_Toc503296479)

[**2.3单元测试：** 3](#_Toc503296480)

[**2.4集成测试：** 4](#_Toc503296481)

[**2.5系统测试：** 4](#_Toc503296482)

[**2.6验收测试：** 4](#_Toc503296483)

[2.6.1验收测试的条件 4](#_Toc503296484)

[2.6.2交付版本的要求 5](#_Toc503296485)

[2.6.3版本发布的准则 5](#_Toc503296486)

[三、编写测试文档 5](#_Toc503296487)

[**3.1测试点：** 5](#_Toc503296488)

[**3.2输入数据：** 6](#_Toc503296489)

[**3.3测试描述：** 6](#_Toc503296490)

[**3.4预期输出数据：** 6](#_Toc503296491)

[**3.5实际输出：** 6](#_Toc503296492)

[**3.6正确与否：** 6](#_Toc503296493)

[**3.7测试结论：** 6](#_Toc503296494)

[四、缺陷管理 6](#_Toc503296495)

[**4.1缺陷的定义及其基本属性：** 6](#_Toc503296496)

[**4.2缺陷分类：** 7](#_Toc503296497)

[**4.3代码缺陷分类：** 7](#_Toc503296498)

[**4.4系统测试缺陷分类：** 8](#_Toc503296499)

[**4.5缺陷等级定义：** 8](#_Toc503296500)

[**4.6缺陷优先级定义** 9](#_Toc503296501)

[**4.7缺陷状态定义：** 9](#_Toc503296502)

[**4.8缺陷完成度：** 10](#_Toc503296503)

[**4.9缺陷管理流程：** 11](#_Toc503296504)

# 一：引言

**1.1编写目的：**

为了确保软件产品质量，使产品能够顺利交付和通过验收，特编写本文档，以作参考。

**1.2范围：**

本文档适用于项目开发过程中的单元测试、集成测试、系统测试、业务测试、验收测试以及一些专项测试。

**1.3职责：**

项目测试负责人组织编制《测试规范》，指导和督促人员完成各阶段的设计测试工作。

项目组测试人员由开发人员兼任，按本文档规范撰写测试大纲及测试报告。

# 

# 二：工作流程

**2.1测试依据：**

详细设计是模块测试的依据。因此设计人员应向测试人员提供《需求规格说明书》、《详细设计》、《概要设计》等有关资料。测试人员必须认真阅读，真正弄懂系统需求和详细设计。

**2.2制订《测试规范》：**

在测试之前，由项目负责人根据《测试计划》的要求，组织人员编制相应的《测试方案》，《测试方案》应包括以下内容：

测试目的；

所需人员及相应培训要求；

测试环境、工具和测试软件；

测试用例、测试数据和预期的结果。

**2.3单元测试：**

项目开发实现过程中，每个程序单元（程序单元的划分视具体开发工具而定，一般定为函数或子程序级）编码调试通过后，要及时进行单元测试。

单元测试由单元开发者自己进行，使用白盒测试方法，根据程序单元的控制流程，争取达到分支覆盖。对于交互式运行的产品，不便于进行自动测试的，可以采用功能测试的方法进行。

单元测试针对程序模块，从程序的内部结构出发设计测试用例。多个模块可以独立进行单元测试。

单元测试内容包括模块接口测试、局部数据结构测试、路径测试、错误处理测试等；

单元测试组织原则一遍根据开发进度安排对已开发完成的单一模块进行测试；

单元测试停止标准：完成了所有规定单元的测试，单元测试中发现的bug已经得到修改。

**2.4集成测试：**

编码开发完成，项目组内部应进行组装测试。

集成测试由项目负责人组织策划（编写测试计划、测试用例）并实施。集成测试着重对各功能模块之间的接口进行测试，验证各功能模块是否能协调工作、参数传递及功能调用是否正常。测试采用交叉方法，即个人开发的软件应由其他的项目组成员进行测试。

**2.5系统测试：**

在项目开发完成之后，应对整个系统软件和硬件进行系统测试。对性能、可靠性、健壮性、压力承受力等方面分别进行评价，以验证系统是否满足规定的需要。

**2.6验收测试：**

### 2.6.1验收测试的条件

按照项目计划规定的验收测试进度安排进行测试准备。

在验收测试前，各项内部的测试活动都受到监控并争取执行。

### 2.6.2交付版本的要求

* 按照集成测试用例完成了整个系统的集成测试
* 集成版本满足设计定义的各项功能、性能要求
* 提交的数据库脚本样本需要完整，没有冗余数据
* 在集成测试中发现的bug已经得到解决，各级缺陷修改率达到标准
* 软件需求分析说明书中定义的所有功能都已经实现，性能指标全部达到性能需求指标
* 提交阶段性测试报告，包括功能和性能测试报告
* 所有文档齐备完整

### 2.6.3版本发布的准则

软件产品通过了单元测试、集成测试、系统测试。

所有测试项必须符合以下标准：

致命错误：无

功能错误：无

以上几项其中之一不满足要求，视为不合格。

在产品交付和用户验收之前，通过验收测试来确认在规定的使用环境下整个产品的运行情况是否满足规定的要求。

在产品交付之前，由指定的验收负责人组织制定测试方案和测试用例，主持验收。

# 三：编写测试文档

**3.1测试点：**

将测试模块分解成多个功能点，测试点应涵盖功能点，也涵盖了正常测试。

**3.2输入数据：**

输入数据包括界面输入数据及其他外部输入数据。

**3.3测试描述：**

描述测试步骤，包括：操作员所执行的动作（包括鼠标、键盘、加载外部数据等操作）；系统的反应，包括：光标定位、光标聚焦、显示字段值、按钮的封闭和放开、功能键的封闭和放开、系统提示和系统消息等。

**3.4预期输出数据：**

按准备的输入数据和设计要求的处理过程，模块应输出的数据。

输出数据包括：屏幕输出数据。

**3.5实际输出：**

填写本测试点程序运行后的实际输出。

**3.6正确与否：**

程序运行后，实际输出结果和预期输出结果一致时，为正常，否则为不正常。

**3.7测试结论：**

填写本次测试的结论，是合格或不合格。若不合格时，应总结存在的问题。

# 四：缺陷管理

**4.1缺陷的定义及其基本属性：**

缺陷是指在软件开发过程中的针对软件产品和开发过程中的问题，这些问题已经影响或可能会影响软件产品的质量。缺陷应该具备以下属性，也就是往缺陷管理库或者缺陷列表中提交的缺陷应该具备以下属性：

|  |  |
| --- | --- |
| 属性名称 | 描述 |
| 缺陷标识 | 标记某个缺陷的一组符号，每个缺陷必须有一个唯一的标识 |
| 缺陷类型 | 根据缺陷的自然属性划分的缺陷种类 |
| 缺陷验证程度 | 因缺陷引起的故障对软件产品的影响程度 |
| 缺陷所处的模块 | 缺陷分步的模块 |
| 缺陷的重现步骤 | 详细的缺陷重现步骤 |
| 附件 | 与缺陷相关的附件（截图、附件、用例等） |
| 备注 | 对缺陷的其他描述 |

**4.2缺陷分类：**

根据缺陷的定义，将缺陷分为如下列：

文档缺陷：是指对文档的静态检查过程中发现的缺陷。检查活动包括同行评审、产品审计等。评审的缺陷要根据被评审对象的类型来确定，被评审的对象包括最终出产物和中间过程产出物，比如需求文档、设计文档、计划、报告、用例等。

代码缺陷：是指对代码进行同行评审、审计或代码走查过程中发现的缺陷。

测试缺陷：是指由测试活动发现的测试对象（被测对象一般是指可运行的代码、系统，不包括静态测试发现的问题）的缺陷，测试活动包括单元测试、集成测试、系统测试等。

过程缺陷：有称为不符合项问题，是指通过过程审计、过程分析、管理评审、质量评估、质量审核等活动发现的关于过程的缺陷和问题。

**4.3代码缺陷分类：**

|  |  |
| --- | --- |
| 缺陷分类 | 描述 |
| 常量变量定义问题 |  |
| 不满足设计或需求 |  |
| 编写代码不符合规范 |  |
| 条件判断处理 |  |
| 循环处理错误 |  |
| 异常处理 |  |
| 算法逻辑问题 |  |
| 注释问题 |  |
| 代码冗余 |  |
| 性能问题 |  |

**4.4系统测试缺陷分类：**

|  |  |
| --- | --- |
| 缺陷类型 | 描述 |
| 功能错误 | 影响了重要的特性、用户界面、产品接口或全局数据结构，并且设计文档需要争取的变更。如逻辑、循环、递归、功能等缺陷 |
| 结构错误 | 应用程序结构化页面无法显示，或者显示错误 |
| 页面文字错误 | 应用程序页面出现的中外文拼写、使用、以及不同语种页面的编码错误 |
| 页面图形错误 | 应用程序页面出现图片内容使用不当，或者无法显示 |
| 排版错误 | 应用程序页面排版不符合要求或者不符合使用习惯 |
| 业务逻辑不合理 | 应用程序的实现流程和规定业务流程不一致，或者实现流程无法正确完成。包括流程数据的部分并行、争用、同步等操作，引起的流程断裂、死锁、以及其他异常情况 |
| 业务逻辑不方便 | 应用程序实现流程在实际情况下虽然可以完成，但是存在不必要的反复、等待、冗余等影响使用效率的情况 |
| 其他错误 | 其他未分类错误 |
| 建议 | 系统改进建议 |

**4.5缺陷等级定义：**

缺陷的严重程度对以上所述的缺陷类型都是适合的，缺陷的严重程度反映的是对缺陷的发现对象可能造成的影响或后果来定义的。

|  |  |  |  |
| --- | --- | --- | --- |
| 缺陷等级 | 缺陷性质 | 系统中对应的错误分类 | 描述 |
| 一级 | 致命错误 | 系统崩溃  系统死锁 | 导致对被描述的主要对象的理解错误、不可行、不可运转、对业务和整个系统造成重大损失或损害；对使用、维护或保管人员有危险或不安全，以及对产品的基本功能有致命影响的缺陷 |
| 二级 | 严重缺陷 | 严重错误 | 对被描述的部分对象的理解或实现错误，部分的模块或系统不可行或不能运转或部分模块和系统缺失，对整个系统有重大影响或可能造成部分的损失或损害；严重影响使用安全 |
| 三级 | 一般缺陷 | 次要错误  布局不合理  文字错误 | 系统中部分单元模块或单个功能描述和实现有错误、有偏差、不一致或有缺失，不影响模块的正常运行，或有影响，但可以有替代的办法或避免办法 |
| 四级 | 微小缺陷 | 微不足道 | 基本不影响系统的运行和功能的实现。但是与标准、规范和定义不一致 |
| 五级 | 建议缺陷 | 新特性 | 不在定义、标准、范围的定义和约束之内，但是从提出者来看是需要完善的建议 |

**4.6缺陷优先级定义**

|  |  |
| --- | --- |
| 缺陷优先级 | 描述 |
| 特急 | 需要立刻进行修改 |
| 加急 | 一天到两天之内必须修改 |
| 高 | 介于中和加急之间 |
| 中 | 缺陷需要正常排队等待修复或列入软件发布清单 |
| 低 | 留到组后解决，如果项目的进度跟紧张可以在产品发布以前不解决 |

**4.7缺陷状态定义：**

|  |  |
| --- | --- |
| 缺陷状态 | 描述 |
| 初始状态（New） | 测试或开发人员提交一个新的缺陷，等待开发人员或项目经理分配修改负责人 |
| 打回（FeedBack） | 要求缺陷的报告者再次对缺陷进行说明 |
| 已分配（Assigned） | 是指已经分配给属主，等待修改。 |
| 已解决（Resolved） | 缺陷被属主修改，等待测试人员验证 |
| 关闭（Closed） | 测试人员验证缺陷已经修复 |
| 重新打开（Reopen） | 测试人员验证，缺陷没有修改正确 |
| 遗留（Later） | 经项目经理和技术经理验证此缺陷在本版本中不用修改 |

**4.8缺陷完成度：**

|  |  |
| --- | --- |
| 缺陷完成度 | 描述 |
| 打开 | 缺陷没有被解决 |
| 已解决 | 缺陷已经修改 |
| 遗留 | 此缺陷步骤本阶段解决 |
| 重新打开 | 重新打开某个缺陷 |
| 不做修改 | 不对这个缺陷进行修改 |
| 重复 | 与某个缺陷重复 |

**4.9缺陷管理流程：**

