软件测试标准规范

# 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。

# 适用范围

本文档适用于项目开发过程中的需求测试、单元测试、集成测试、系统测试、关键业务流程测试、验收测试以及一些专项测试（如：性能测试、安全性测试等）。

# 职责

* 项目测试负责人组织编制《测试计划》、《测试方案》，指导和督促测试人员完成各阶段的测试工作。
* 项目组测试人员按照《测试计划》、《测试方案》完成所承担的测试任务，并按要求填写《问题报告及维护记录》。
* 测试经理依照确认规程和准则对工作产出物进行确认，提出对确认规程和准则的修改意见。
* 项目经理人组织相关人员进行测试环境的安装与实施。
* 项目经理审核负责控制整个项目的时间和质量（开发与测试的质量）。
* 开发人员确认与修改测试人员提交的Bug。测试人员对已修Bug进行回归验证。修复无效则重新打回至相应的开发人员。

# 工作流程

## 测试依据

项目经理应向测试人员提供《客户需求规格说明书》、《概要设计》、《详细设计》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。

## 需求确认（需求测试）

需求确认（需求测试）方法如下：

1、需求描述是否具备完整性；（没有遗漏内容；或描述片面）  
2、需求描述是否有二义性；（没有让不同的人有不同的理解结论）  
3、需求描述是否是正确的；（需求之间没有冲突等）  
4、是否包含有非功能属性的需求；（性能，安全性，可靠性，易用性等）  
5、是否需求是可以验证的；（需求描述具备可测试性）  
6、需求是否可实现。

**特别需要对需求文档中的文字描述质量进行把关。**设想如果我们测试人员无法在需求的有无上进行判断或需求的价值上进行建议，那么测试人员至少可以在需求描述的质量上进行一定的把关。有一个很简单的道理：如果开发人员在需求描述上都存在含混不清，或是描述不准确，不完整，那么一定会对需求理解产生偏差，在开发实现时与最初需求编写者的要求产生偏差与歧义。测试人员在依据需求进行测试用例编写时，也会出现需求理解困难，导致测试用例编写的困难，最终测试未能覆盖到需求编写者的最初想法。

**建议：需求评审最好选取项目测试经验最丰富的同事来实施。**

## 制订《测试方案》

在测试之前，由项目经理根据《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：

1、测试目的；

2、所需人员及相关业务培训要求；

3、测试环境搭建；

4、测试工具；

5、应用代码版本控制工具；

6、测试用例，包括测试数据和预期结果。

## 单元测试（由开发人员自己完成，工具有junit、pmd等）

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，从模块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般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。
* **特别重要的是，代码必须有注释，以便经后跟踪。**

## 集成测试（由测试人员配合开发人员共同完成）

编码开发完成，**项目组内部应进行模块与模块之间的组装测试。可能需要针对不同模块，开发驱动模块或桩模块。**

集成测试由项目经理组织策划（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。

**集成测试过程应填写《问题报告及维护记录》，测试结果应形成《测试报告》。**

## 系统测试（全部由测试人员完成，分手工与自动化测试，系统整体的功能测试（业务流程、用户界面、用户体验等）、性能测试、安全性测试）

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。

系统测试由测试负责人组织策划（编写测试计划、测试用例）并实施，系统测试过程应形成《问题报告及维护记录》。

**系统测试一般进行如下几种情况的测试：**

* 正常情况
* 非正常情况
* 破坏性测试
* 边界情况
* 非法情况
* 强度测试
* 性能测试
* 兼容性测试
* 用户友好性测试

**界面设计规范测试：**

* 光标的初始位置
* 字体是否统一
* 字号是否符合规定
* 标题颜色
* 按钮的名称是否规范
* 界面布局是否合理，整体效果如何

**输入值测试：**

* 数据类型
* 数据长度
* 约束条件是否满足，是否完整
* TAB和Enter键是否起作用
* 键盘操作能否全部代替鼠标操作
* 输入（光标）是否按照顺序前进

**按钮测试：**

* 将按钮放开和封闭(置为不可用)是否严格、准确，不能使用的按钮必须封闭(置为不可用)。
* 检查“退出”、“取消”等具有共性按钮的功能

**异常情况测试：**

在完成正常功能测试后，安正常处理的相同操作顺序，执行与正常处理不同的动作例如

* 正常处理中要求输入日期的字段，这时输入字符或数字
* 正常处理中输入字段有范围要求，这时输入超过范围的值
* 正常处理中用两个值限定范围，这时用一个值或不限定
* 正常处理中要求用“Tab”键，这时安“Enter”键或其他键
* 正常处理中单选框、多选框、下拉框等，十一偶那个非指定键操作
* 使用不同于指定的按钮操作

## 业务流程测试

在系统测试结束后，均可由最终用户或测试人员对系统进行测试。业务测试着重测试业务流程，功能、用户界面等方面。

项目、测试负责人负责组织相关人员制定测试方案和测试用例，并进行测试。

测试的结果应形成《问题报告及维护记录》。

## 性能测试与安全性测试

需在稳定的版本上进行，由其是要求一个独立的测试环境，与开发环境、功能测试环境区分。

## 验收测试

### 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

### 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

### 版本发布的准则

* 软件产品通过了单元测试、集成测试、业务测试、系统测试、性能测试
* 测试部提交文档：测试计划、测试方案、测试用例、测试分析报告
* 所有测试项必须符合以下标准
* 致命错误：无
* 功能错误：无
* 功能缺陷：项目经理、技术经理、测试负责人审核通过
* 界面缺陷：项目经理、技术经理、测试负责人审核通过
* 建议：项目经理、技术经理、测试负责人审核通过
* 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。

验收测试过程应形成《问题报告及维护记录》。

## 用户现场测试

将软件部署到用户实际生产环境后，由于环境差异，需要在用户现场进行确认测试，保证系统功能、性能完备，可正常运行。测试内容：

* 根据软件系统规模，准备现场测试用例，涵盖所有重要功能点，若规模小，需要将全部功能点全部测试一遍
* 对于后台已定义好的工作流、功能栏目路径以及用户信息等数据，不可进行修改和删除操作，新增的测试数据也需要在测试完成后给予清楚
* 重点检查上传、下载的数据是否可以正常的打开或保存
* 确认界面美观，基本信息和链接无错误
* 考虑用户实际的软件环境和网络环境，以客户端最为复杂的软硬件环境作为测试机器，检查有无异常情况出现
* 针对前期发现的bug进行回归测试，以保证发布版本为最新版本

## 编写测试文档

### 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，也涵盖了正常测试和异常测试。

### 输入数据

输入数据包括界面输入数据、数据库的初始数据及其他外部输入数据。特别是数据库的初始所需属性一一列出，全面是指：数据能达到模块所涉及的全部功能，典型是指这个数据能充分反映功能特点。

### 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。

### 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、输出到数据库的数据、输出到其他外部介质上的数据，并指出断点结果或最终结果。

### 实际输出

填写本测试点程序运行后的实际输出。

### 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。

### 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，可以让修改者一目了然。

# 缺陷管理

## 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 |
| 缺陷所处的模块或子系统 | 缺陷分步的模块或子系统 |
| 缺陷出现几率 | 指发现错误的几率，分必现、非必现（存在隐患） |
| 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 |

## 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、性能测试等
* 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。过程缺陷的发现者一般是测试人员、项目经理等

## 文档缺陷分类

|  |  |
| --- | --- |
| 缺陷分类 | 描述 |
| 描述不完整 | 文档内容缺失，或文档应该包括的范围没有涵盖 |
| 不一致 | 一致性问题有两类：  一是与源头说明书不一致，比如需求和客户业务需求不一致、设计与需求不一致等  二是上下文或者与前提不一致 |
| 描述错误 | 文档描述是错误的，不可实现或导致错误的输出或结果 |
| 功能问题 | 该缺陷将会导致用户功能的错误、不满足、不可用 |
| 不清楚或有歧义 | 内容的描述不清楚、不能准确表达、或表达的意思有歧义 |
| 逻辑错误 | 内容组织逻辑不清楚、逻辑错误 |
| 接口问题 | 与最终用户接口问题、与外部系统的接口问题、内部子系统或模块的接口问题 |
| 输入输出问题 | 输入输出不完整、不正确、不可测试或验证 |
| 不细化 | 内容还需要进一步细化 |
| 性能问题 | 文档的设计或实现方式存在性能问题 |
| 安全性问题 | 文档的设计或实现方式存在安全性问题 |

## 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 |  |

## 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 | Web应用程序结构化页面无法显示，或者显示错误 |
| 脚本错误 | Web应用程序当中出现脚本错误，包括客户端对数据进行校验和运算的各种情况下产生的错误 |
| 页面链接错误 | Web应用程序页面出现空链接、错误链接、死链接 |
| 页面文字错误 | Web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 | Web应用程序页面出现图片内容使用不当，或者无法显示 |
| ALT错误 | Web应用程序页面当中超文本标识语言、文本标签解释错误 |
| 排版错误 | Web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 |

## 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（可能无法按需求规定实现）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 |

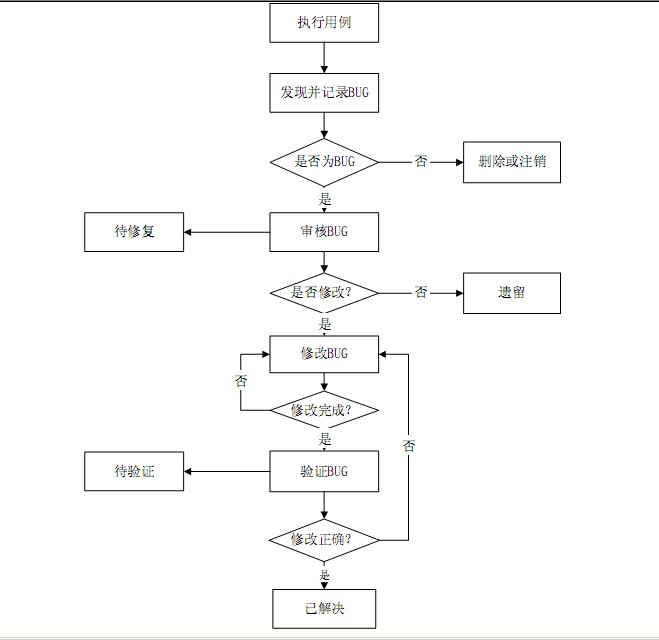
## 缺陷优先级定义

|  |  |
| --- | --- |
| 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 |

## 缺陷状态定义

|  |  |
| --- | --- |
| 缺陷状态 | 描述 |
| 初始状态 | 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 |
| 确认缺陷 | 与开发确定是否为缺陷。非缺陷打回；是缺陷进行分配 |
| 已解决 | 缺陷被属主修改，等待测试人员验证 |
| 验证缺陷 | 验证成功则关闭缺陷；验证未成功则重新打开缺陷 |
| 关闭缺陷 | 测试人员验证缺陷已经修复 |
| 重新打开缺陷 | 测试人员验证，缺陷没有修改正确 |
| 遗留缺陷 | 经项目经理和技术经理验证此缺陷在本版本中不用修改 |

## 缺陷管理流程



# 处理机制

## 退回机制

若在测试过程中发生如下情况，将系统退回到申请部门：

* 经过测试后，发现与需求说明规格说明书中定义的功能项存在较大的差异
* 单一模块，测试过程中发现缺陷输了较多或者无法继续进行系统其它功能模块的测试，继续测试无意义
* 测试过程中，频繁死机或系统崩溃
* 主业务流程出现断点

## 异常情况处理机制

非正常情况下，需要进行特别处理的情形，此情况需要主管领导签字确认：

* 上线时间紧急的情况下，未经测试部充分测试就需要部署到用户现场
* 作为总包时，子商进度明显延迟，尚未进行验收测试就需要上线

## 报告机制

若出现以下情况，需要及时向部门领导和项目经理汇报的情况：

* 测试后期出现重大逻辑错误，修改测试影响上线时间
* 测试过程中用户需求出现重大变更
* 测试负责人定期汇报测试情况

# 测试完成的标准

## 被测试出的、在软件错误级别分类中定义的：

* 一级缺陷，致命错误，100%得到修改并且复测通过
* 二级缺陷，严重错误，100%得到修改并且复测通过
* 三级缺陷，一般错误，95%得到修改并且复测通过
* 四级缺陷，轻微错误，95%得到修改并且复测通过

## 用户可以接受未修改的软件错误

## 测试超过了预定时间表，由项目经理决定是否停止测试

## 测试结论及评价标准

|  |  |
| --- | --- |
| 测试结论 | 评价标准 |
| 拒绝发布 | 遗留了一级、二级缺陷 |
| 测试通过版本 | 不能遗留以一、二类缺陷  三类 一般缺陷95%得到修改并且通过复测  四类轻微缺陷85%得到修改并且通过复测 |
| 推荐使用版本 | 不能遗留以一、二类缺陷  三类 一般缺陷95%得到修改并且通过复测  四类轻微缺陷90%得到修改并且通过复测 |
| 可以证实发布版本 | 不能遗留以一、二类缺陷  三类 一般缺陷97%得到修改并且通过复测  四类轻微缺陷90%得到修改并且通过复测 |

## 输出

《阶段性测试报告》

《性能测试报告》

《测试总结报告》

《测试问题列表》

# 其他约束

# 9 记录

|  |  |  |
| --- | --- | --- |
| 序 号 | 名 称 | 编 号 |
| 1 | 测试计划 |  |
| 2 | 测试方案 |  |
| 3 | 问题报告及维护记录 |  |
| 4 | 测试总结报告 |  |