单元测试报告

北京XX技术有限公司

2005.6

本文档所包含的内容属于北京XX技术有限公司成都研发中心的内部文档，未经授权或特别许可，不得擅自复制或散布本文档的部分或全部内容。

版权所有（C）2000-2005 北京XX技术有限公司。保留所有权利。

目录与索引

术语清单 5

第一章 基本测试信息 5

第二章 环境描述 5

第三章 静态检查 5

3.1 XXX模块 5

3.2 YYY模块 7

第四章 动态执行测试 7

4.1 测试说明 7

4.2 测试单元划分 8

4.3 测试桩和测试代码设计 8

4.4 测试用例和结果 8

4.4.1 XXX模块 8

4.4.1.1 测试单元1 8

第五章 总结和评价 8

5.1 测试结果统计 8

5.2 测试评估 9

5.3 测试总结和改进建议 9

第六章 遗留问题 9

第七章 附件 10

7.1 附件1：交付的测试工作产品 10

7.2 附件2：修改、添加的测试方案或测试用例 11

7.3 附件3：其它附件 11

参考资料清单 11

表目录

表１　　ＸＸ表 5

表２　　ＸＸ表 5

图目录

图１　　ＸＸ图 5

图２　　ＸＸ图 5

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期Date | 修订版Revision version | 描述Description | 作者Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 术语清单

对本文所用术语进行说明，要求提供每个术语的英文全名和中文注释。

# 第一章 基本测试信息

|  |  |
| --- | --- |
| 被测试产品名称 |  |
| 被测试产品版本 |  |
| 测试时间 |  |
| 测试地点 |  |
| 测试人员及任务分配 |  |

# 第二章 环境描述

描述本次测试的软硬件测试环境和采用的单元测试工具以及测试方法等。

# 第三章 静态检查

## 3.1 XXX模块

通过人工检查方法对XXX模块进行检查，并填写下表。

|  |  |  |
| --- | --- | --- |
| 活动 | 活动说明 | 检查结果 |
| 检查算法的逻辑正确性 | 确定所编写的代码算法、数据结构定义（如：队列、堆栈等）是否实现了模块或方法所要求的功能。 |  |
| 确定模块接口的正确性 | 确定形参个数、数据类型、顺序是否正确；确定返回值类型及返回值的正确性。 |  |
| 输入参数有没有作正确性检查 | 如果没有作正确性检查，确定该参数是否的确无需做参数正确性检查，否则请添加上参数的正确性检查。 |  |
| 调用其他方法接口的正确性 | 检查实参类型正确与否、传入的参数值正确与否、个数正确与否。返回值正确与否，有没有误解返回值所表示的意思。 |  |
| 出错处理 | 模块代码要求能预见出错的条件，并设置适当的出错处理，以便在一旦程序出错时，能对出错程序重做安排，保证其逻辑的正确性，这种出错处理应当是模块功能的一部分。若出现下列情况之一，则表明模块的错误处理功能包含有错误或缺陷：出错的描述难以理解；出错的描述不足以对错误定位，不足以确定出错的原因；显示的错误信息与实际的错误原因不符；对错误条件的处理不正确；在对错误进行处理之前，错误条件已经引起系统的干预等。 |  |
| 保证表达式和语句的正确性； | 检查所编写的语句的语法、逻辑的正确性。对表达式应该保证不含二义性，对于容易产生歧义的表达式或运算符优先级（如：《 、=、 》、 &&、||、++、 --等）可以采用扩号“（）”运算符避免二义性，这样一方面能够保证代码的正确可靠，同时也能够提高代码的可读性。 |  |
| 检查常量或全局变量使用的正确性 | 确定所使用的常量或全局变量的取值和数值、数据类型；保证常量每次引用同它的取值、数值和类型的一致性。注意对全局变量使用的互斥问题。 |  |
| 表示符定义的规范一致性； | 保证变量命名能够见名知意，并且简洁但不宜过长或过短、规范、容易记忆、最好能够拼读。 |  |
| 程序风格的一致性、规范性 | 代码必须能保证符合企业规范，保证所有成员的代码风格一致、规范、工整。 |  |
| 检查程序中使用到的神秘数字是否采用了表示符定义 | 神秘的数字包括各种常数、数组的大小、字符位置、变换因子以及程序中出现的其他以文字形式写出的数值。在程序源代码里，一个具有原本形式的数对其本身的重要性或作用没提供任何指示性信息，它们也导致程序难以理解和修改。对于这类神秘数字必须采用相应的标量来表示；如果该数字在整个系统中都可能使用到务必将它定义为全局常量；如果该神秘数字在一个类中使用可将其定义为类的属性（Attribute）,如果该神秘数字只在一个方法中出现务必将其定义为局部变量或常量。 |  |
| 检查代码是否可以优化、算法效率是否最高。 | 如：语句是否可以优化，是否可以用1条语句代替程序中的多条语句的功能，循环是否必要，循环中的语句是否可以抽出到循环之外等。 |  |
| 检查您的程序是否清晰简洁容易理解。 | 注意：冗长的程序并不一定不是清晰的。 |  |
| 检查方法内部注释是否完整；是否清晰简洁 | 是否正确的反映了代码的功能，错误的注释比没有注释更糟；是否做了多余的注释；对于简单的一看就懂的代码没有必要注释。 |  |
| 检查注释文档是否完整 | 对包、类、属性、方法功能、参数、返回值的注释是否正确且容易理解；是否会落了或多了某个参数的注释，参数类型是否正确，参数的限定值是否正确。特别是对于形式参数与返回值中关于神秘数值的注释，如：类型参数 应该指出 1.代表什么，2.代表什么，3.代表什么等。对于返回结果集(Result Set)的注释，应该注释结果集中包含那些字段及字段类型、字段顺序等。 |  |

## 3.2 YYY模块

# 第四章 动态执行测试

## 4.1 测试说明

对动态执行测试的相关问题进行说明，包括：

a．测试包括的内容：执行路径，逻辑判定，边界，数据有效性

b．采用了哪些单元测试方法

c．使用了何种单元测试工具

d．覆盖测试使用的方法

e．测试要达到何种目标

f．是否要测试不同优化选项下的代码执行情况

## 4.2 测试单元划分

描述各模块的测试单元划分（包括单元结构，单元之间的依赖关系，各单元功能及涉及的目录或文件）。（注意：将每个模块划分为多个测试单元，后面的测试用例都是针对测试单元的）

## 4.3 测试桩和测试代码设计

描述用于进行单元测试的测试驱动和测试桩以及相关公用软件设计。

## 4.4 测试用例和结果

### 4.4.1 XXX模块

#### 4.4.1.1 测试单元1

列表给出单元测试用例。

|  |  |
| --- | --- |
| 用例编号 | TST1\_001 注：编号一般分为两个部分，前面是测试单元名缩写，后面是3位的编号，从001-999 |
| 测试目的 | 如：对错误逻辑输入检验。注：指出要测试的目的 |
| 测试内容描述 | 如：对于int fun3（char \*p1, int p2 ）的输入检验，如果 p1 == null ,程序中检验到，应该记录到系统 logfile, return –1; |
| 输入期望 | p1 == null |
| 功能处理期望描述 | Logfile 多一条历史记录，方法return -1； |
| 输出期望 | Return –1 |
| 单元测试结果 | |
| 实际输入数据 | p1 = null |
| 实际处理情况描述 | 程序没有进行p1 == null 的验证，没有及时return –1，而是继续运行，导致系统异常。 |
| 实际输出 | 没有写 logfile 文件； |
| 测试结论 | 通过/不通过 |

4.4.2 YYY模块

# 第五章 总结和评价

## 5.1 测试结果统计

对本次测试的项目进行统计，包括总项数，通过项数，失败项数，部分通过项数以及百分比等，一般用以下表格进行描述，其中的字段可根据实际情况进行设计和裁剪：

表1 测试结果统计表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 总测试用例数 | 实际测试用例数 | 通过 | 不通过 | 待定 | 未测试 | 无需测试 |
| 数目 |  |  |  |  |  |  |  |
| 百分比 |  |  |  |  |  |  |  |

## 5.2 测试评估

对被测试对象以及测试活动给出总结性的评估，包括稳定性、测试充分性等。

对被测试对象和测试活动的评估必须参照软件需求规格说明的要求，分析被测试对象与软件需求规格的偏离程度、偏离点，同时需要对结果偏离以及测试不充分引起的失败风险进行评估。

注意：评估的标准必须基于测试计划中确定的被测试对象通过/失败准则。总结测试结果时，要确定测试过程中的所有问题，要对问题的解决情况进行确认。

## 5.3 测试总结和改进建议

总结本次测试活动的经验教训，总结主要的测试活动和事件。总结资源消耗数据，如总人员，总机时，每个主要测试活动花费的时间。

提供对本次测试过程活动的测试设计和操作的改进建议。每一条建议的分析及其对软件测试的影响也应提供。在测试过程中形成的对测试设计、测试用例的修改和补充的具体改进内容可列在本测试总结报告文档的附录中。

# 第六章 遗留问题

遗留问题是指测试过程中发生的并且在测试报告时仍没有得到解决的测试问题。测试报告时已经解决的，并已经通过回归测试的测试问题不记入其中。

在详细的遗留问题报告前可以先建立一个遗留问题统计表格，以便对遗留问题的相关分布信息有整体的了解，如果遗留问题数比较少，可以将此表格省去，因此此表格根据实际情况可选。建立遗留问题统计表格，可对遗留问题数和级别进行统计，包括问题总数，致命，严重，一般和提示问题的数目以及百分比等，遗留问题统计一般可用以下表格描述，其中的字段可根据实际情况进行设计和裁剪：

遗留问题统计表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 问题总数 | 致命问题 | 严重问题 | 一般问题 | 提示问题 | 其它统计项 |
| 数目 |  |  |  |  |  |  |
| 百分比 |  |  |  |  |  |  |

以下部分详细记录每一个遗留问题，也可以视时间情况只详细记录问题级别比较高的遗留问题，低级别的遗留问题采用简单列表进行罗列。所有进行详细记录的遗留问题都统一采用表格的形式来描述，每个问题一个表格，表格的形式如下：

遗留问题1

|  |  |
| --- | --- |
| 问题简述 | 对问题的简短概要描述 |
| 问题详述 | 给出问题的详细描述，可以从如下几个方面考虑：  1） 环境和设置  2） 版本配套情况  3） 输入  4） 测试的步骤  5） 预期输出结果  6） 实际输出结果  7） 问题重现的方法  应包括对定位和修正有帮助的活动和现象，例如：描述任何对定位问题有帮助的测试用例执行情况，以及任何与规定的测试步骤之间的差异等。 |
| 问题级别 | 级别定义如下：  致命：引起系统死机或者系统崩溃的问题  严重：引起系统某一个功能失效且不能简单恢复的问题  一般：引起系统某一功能失效但可简单恢复或较难重现的问题  提示：从操作或者维护的角度发现的问题或者建议 |
| 问题分析与对策 | 针对此问题提出影响程度分析，应对策略，例如应急措施、更新产品资料或者纳入下一版本需求等。  指出该问题对项目设计文档，测试文档等可能带来的影响。 |
| 避免措施 | 针对此问题提出的可预防或避免发生的策略。如果没有措施可避免问题的发生，则填“无”。 |
| 备注 | 其它补充内容。 |

# 第七章 附件

本部分中，交付的测试工作产品和测试项目通过情况清单为必须的附件，其余可根据实际情况进行裁减，不同的测试报告根据需要可以给出不同类型的附件。附件的目的是帮助本报告的使用者理解报告，记录修改情况和有用的数据等。

## 7.1 附件1：交付的测试工作产品

指明本测试完成后交付的测试文档、测试代码、测试工具等测试工作产品，一般包括但不限于如下工作产品：

1） 测试计划

2） 测试设计

3） 测试用例

4） 测试规程

5） 测试日志

6） 测试事件报告

7） 测试总结报告

8） 测试输入和输出数据

9） 测试工具

10）测试代码以及设计文档

## 7.2 附件2：修改、添加的测试方案或测试用例

对“4.3 测试总结和改进建议”中描述的任何需要修改、添加的测试方案或测试用例进行描述。

## 7.3 附件3：其它附件

# 参考资料清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 作者 | 编号 | 发布日期 | 查阅地点或渠道 | 出版单位（若不为本公司发布的文献，请填写此列） |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |