**OTA安全管理系统**

**测试策略与测试方法**

**天津三源电力智能科技有限公司**

第 1 页

**版本修订说明**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **修改时间** | **修改人** | **审核人** | **备注** |
| V1.0 | 2022.2.10 | 王志 | 孙剑 | 新建 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1. 目的 1](#_Toc25134)

[2. 适用范围 1](#_Toc21008)

[2.1 机构： 1](#_Toc17258)

[2.2 业务： 1](#_Toc1296)

[3. 概述 1](#_Toc13550)

[4. 名词术语 1](#_Toc25871)

[5. 测试策略 1](#_Toc31470)

[5.1 功能测试 1](#_Toc22080)

[5.2 用户界面测试 2](#_Toc32363)

[5.3 性能测试 3](#_Toc25781)

[5.4 安全性测试 4](#_Toc11749)

[5.5 自动化测试 5](#_Toc21723)

[6. 测试方法 5](#_Toc13398)

[6.1 黑盒测试 5](#_Toc17661)

[6.2 白盒测试 6](#_Toc16690)

[6.3 集成测试 6](#_Toc20427)

[7. 培训需求 7](#_Toc12612)

# 

# 目的

列出可采用的测试方法、策略，并对这些方法、策略加以说明，供测试人员分析选择合适的测试方法及策略。

# 适用范围

## 机构：

技术部

## 业务：

描述测试的各种测试策略，测试人员可以根据项目的情况选择测试策略，以便于更好的适应项目的测试工作。

# 概述

对工作量，工期和成本进行估计以利于更好的进行后期工作的规划和进行

# 名词术语

# 测试策略

静态测试是依据需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行，对软件进行分析、检查和测试，不实际运行被测试的软件，约可找出30～70%的逻辑设计错误。

动态测试通过运行软件来检验软件的动态行为和运行结果的正确性。动态测试的两个基本要素：被测试程序和测试数据(测试用例)。

## 功能测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试的功能正常，其中包括导航，数据输入，处理和检索等功能。 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  在使用有效数据时得到预期的结果。在使用无效数据时显示相应的错误消息或警告消息。各业务规则都得到了正确的应用。主要实施方法为：等价类划分、边界类分析法，因果图法 |
| 开始标准 | 在单元集成测试结束后进入功能测试 |
| 完成标准 | Bug全部修复完毕，回归测试结束后完成 |
| 测试重点 和优先级 | 重点在于系统的功能方面，基本功能应置于高优先级 |
| 需考虑的 特殊事项 | 功能测试就是对产品的各功能进行验证，根据功能测试用例，逐项测试，检查产品是否达到用户要求的功能，是测试的基础，所有软件项目必须通过功能测试； |

## 用户界面测试

|  |  |
| --- | --- |
| 测试目标 | 通过测试进行的浏览可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab键、鼠标移动、和快捷键）的使用  窗口的对象和特征（例如，菜单、大小、位置、状态和中心）都符合标准*。* |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。 |
| 开始标准 | 功能测试完成 |
| 完成标准 | 成功地核实出各个窗口都与基准版本保持一致，或符合可接受标准。 |
| 测试重点 和优先级 | 兼容性、布局 |
| 需考虑的 特殊事项 | 通过测试进行的浏览可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab键、鼠标移动、和快捷键）的使用  窗口的对象和特征（例如，菜单、大小、位置、状态和中心）都符合标准*。* |

## 性能测试

|  |  |
| --- | --- |
| 测试目标 | 核实所指定的事务或业务功能在以下情况下的性能行为：  正常的预期工作量  预期的最繁重工作量（系统可承受最大并发用户数）  测试方面：负载、压力 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | [使用为功能或业务周期测试制定的测试过程。  通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代数量。脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多个客户机（虚拟的或实际的客户机，请参见下面的“需要考虑的特殊事项”）上重复。] |
| 开始标准 | 录制脚本 |
| 完成标准 | 单个事务或单个用户：在每个事务所预期时间范围内成功地完成测试脚本，没有发生任何故障。  多个事务或多个用户：在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。 |
| 测试重点 和优先级 | 对整体系统的最大可支持并发用户数和合理相应时间 |
| 需考虑的 特殊事项 | 综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：  直接将“事务强行分配到”服务器上，这通常以“结构化语言”（SQL）调用的形式来实现。  通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机。此负载可通过“远程终端仿真（Remote Terminal Emulation）工具来实现。此技术还可用于在网络中加载“流量”。  使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载。  性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  性能测试所用的数据库应该是实际大小或相同缩放比例的数据库。 |

## 安全性测试

|  |  |
| --- | --- |
| 测试目标 | 应用程序级别的安全性：[核实Actor只能访问其所属用户类型已被授权访问的那些功能或数据。]  系统级别的安全性：[核实只有具备系统和应用程序访问权限者才能访问系统和应用程序。] |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 应用程序级别的安全性：[确定并列出各用户类型及其被授权访问的功能或数据。]  [为各用户类型创建测试，并通过创建各用户类型所特有的事务来核实其权限。]  修改用户类型并为相同的用户重新运行测试。对于每种用户类型，确保正确地提供或拒绝了这些附加的功能或数据。  系统级别的访问：[请参见以下的“需考虑的特殊事项”。] |
| 开始标准 |  |
| 完成标准 | [可以相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。] |
| 测试重点 和优先级 |  |
| 需考虑的 特殊事项 | [必须与相应的网络或系统管理员一直对系统访问权进行检查和讨论。由于此测试可能是网络管理可系统管理的职能，可能会不需要执行此测试。] |

## 自动化测试

|  |  |
| --- | --- |
| 测试目标 | 自动化测试是把以人为驱动的测试行为转化为机器执行的一种过程，实现程序自动化执行回归测试，减少人力成本。 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 使用Selenium、Python编写Web自动化测试脚本，对程序执行自动化回归测试，生成测试报告，现在各个用例的测试情况。 |
| 开始标准 | 功能测试完成且所有问题关闭后，编写自动化测试用例后，进行自动化测试脚本编写； |
| 完成标准 | 完成自动化测试用例编写，完成自动化脚本编写，且正常执行后生成测试报告。 |
| 测试重点 和优先级 | 重点关注系统中优先级较高的用例，优先完成系统主流程测试脚本； |
| 需考虑的 特殊事项 | 不能取代手工测试，有很多需要人脑判断结果的测试用例无法用自动工具实现，或者代价太大，需要测试人员有编码能力，需要测试人员能力较高水平 |

# 测试方法

## 黑盒测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试的功能正常，其中包括导航，数据输入，处理和检索等功能。 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。 |
| 开始标准 | 在单元集成测试结束后进入测试 |
| 完成标准 | Bug全部修复完毕，回归测试结束后完成 |
| 测试重点 和优先级 | 重点在于系统的功能方面，基本功能应置于高优先级 |
| 需考虑的 特殊事项 |  |

## 白盒测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试的代码正常，其中包括程序内部逻辑结构，逻辑路径进行测试 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 白盒测试的测试方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法、域测试、符号测试、路径覆盖和程序变异。  白盒测试法的覆盖标准有逻辑覆盖、循环覆盖和基本路径测试。其中逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。 |
| 开始标准 | 在代码编写完成后进入测试 |
| 完成标准 | Bug全部修复完毕，回归测试结束后完成 |
| 测试重点 和优先级 | 依据程序设计说明书，按照一定规范化的方法进行软件结构划分和设计测试用例 |
| 需考虑的 特殊事项 | 白盒测试对测试人员要求较高，需要测试人员有研发编码能力，对编码进行检查，白盒测试需要花费更多时间和人力，需要评估项目时间； |

## 集成测试

|  |  |
| --- | --- |
| 测试目标 | 确保系统模块之间的正常工作，外部接口和内部接口运行数据准确性 |
| 测试范围 | 软件需求说明书所描述的功能和范围 |
| 技术 | 集成测试的目标是按照设计要求使用那些通过单元测试的构件来构造程序结构。单个模块具有高质量但不足以保证整个系统的质量。有许多隐蔽的失效是高质量模块间发生非预期交互而产生的。以下两种测试技术是用于集成测试：  1）功能性测试。使用黑盒测试技术针对被测模块的接口规格说明进行测试。  2）非功能性测试。对模块的性能或可靠性进行测试。 |
| 开始标准 | 在代码编写完成后进入测试 |
| 完成标准 | Bug全部修复完毕，回归测试结束后完成 |
| 测试重点 和优先级 | 集成测试是确保各单元组合在一起后能够按既定意图协作运行，并确保运作中数据 |
| 需考虑的 特殊事项 | 集成测试可以更好的保障项目各个模块间数据通讯，对系统测试更底层，可以更早发现暴露问题，建议项目时间允许情况下都执行集成测试。 |

# 培训需求

测试组成员接受测试方法策略培训。