<自动售货机（ASM）模拟软件>

测试计划

版本 <1.0>

[注：以下提供的模板用于 Rational Unified Process。 其中包括用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=正文）。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2022.10.10 | 1.0 | 第一版 | 牟战 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 简介](" \l "1.     Introduction)

[1.1 目的](" \l "1.1     Purpose)

[1.2 背景](" \l "1.2     Background)

[1.3 范围](" \l "1.3     Scope)

[1.4 项目核实](" \l "1.4               Project Identification)

[2. 测试需求](" \l "2.     Requirements for Test)

[3. 测试策略](" \l "3.     Test Strategy)

[3.1 测试类型](" \l "3.1               Testing Types)

[3.1.1 数据和数据库完整性测试](" \l "3.1.1     Data and Database Integrity Testing)

[3.1.2 功能测试](" \l "3.1.2     Function Testing)

[3.1.3 用户界面测试](" \l "3.1.4     User Interface Testing)

[3.1.4 性能评测](" \l "3.1.5     Performance Profiling)

[3.1.5 负载测试](" \l "3.1.6     Load Testing)

[3.1.6 强度测试](" \l "3.1.7     Stress Testing)

[3.1.7 安全性和访问控制测试](" \l "3.1.9     Security and Access Control Testing)

[3.1.8 配置测试](" \l "3.1.11     Configuration Testing)

[3.1.9 安装测试](" \l "3.1.12     Installation Testing)

[3.2 工具](" \l "3.2     Tools)

[4. 资源](" \l "4.     Resources)

[4.1 角色](" \l "4.1     Workers)

[4.2 系统](" \l "4.2     System)

[5. 项目里程碑](" \l "5.     Project Milestones)

[6. 可交付工件](" \l "6.     Deliverables)

[6.1 测试模型](" \l "6.1     Test Model)

[6.2 测试记录](" \l "6.2      Test Logs)

[6.3 缺陷报告](" \l "6.3     Defect Reports)

[7. 附录 A： 项目任务](" \l "7.     Appendix A: Project Tasks)

测试计划

# 1.     简介

## 1.1     目的

①确定现有项目的信息和应测试的软件构件。

②列出推荐的测试需求（高级需求）。

③推荐可采用的测试策略，并对这些策略加以说明。

④确定所需的资源，并对测试的工作量进行估计。

⑤列出测试项目的可交付元素。

## 1.2     背景

经讨论，我们的ASM模拟软件的功能大致已确定，包括商品管理、管理员登陆、商品显示和支付方式四个部分。

具体的参与者包括管理员和顾客，管理员系统需要完成的具体功能有身份验证、取走钱币、投放钱币、投放货物、设置价格和查看订单，顾客系统包括投放钱币、查看货物价格和购买货物。

在撰写软件测试计划前，项目经理完成了软件开发计划，系统分析员完成了软件需求规约，界面设计师完成了界面设计文档，我们已经对此项目有了初步的构思与设计。

## 1.3     范围

在软件测试的初期，我们的测试主要采用单元测试，对完成的每一个小的功能模块进行测试，确保每一个模块都能实现其功能。接着对软件进行集成测试，在单元测试的基础上，将所有的模块根据软件设计师的要求组装成为一个个子系统，分别对每个子系统进行测试。最后将一个个子系统组装成一个系统进行测试。

我们所给出的所有功能（具体见1.2 背景）都将接受测试，除此外，还会对响应时间、事务处理速率等性能方面进行测试。

在软件测试过程中可能会遇到以下几方面的风险：对需求理解的不充分导致测试结果不具有代表性；采用不擅长的技术对测试结果的可信度以及测试进度造成影响的风险；测试环境不具备影响测试进度的风险。

## 1.4      项目核实

下表列出了制定*测试计划*时所使用的文档，并标明了各文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档**  **（版本/日期）** | **已创建或可用** | **已被接收或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 🗹 是  否 | 🗹 是  否 | 高轩 |  |
| 功能性规约 | 🗹 是  否 | 🗹 是  否 | 宣傲 |  |
| 用例报告 | 🗹 是  否 | 🗹 是  否 | 吴锦超 |  |
| 项目计划 | 🗹 是  否 | 🗹 是  否 | 高轩 |  |
| 设计规约 | 🗹 是  否 | 🗹 是  否 | 吴锦超 |  |

# 2.     测试需求

# **2.1 用例**

# 购买商品、调整商品数量、调整商品种类、调整商品价格和登陆ASM系统

# **2.2 功能性需求**

本 ASM 模拟软件面向普通用户及管理员，因此要求符合一般的操作交互方式。 使用该 ASM 软件，可以支持用户购买商品，管理员调整商品价格、类别、数量，维护该 ASM 内部的商品信息。

# **2.3 非功能性需求**

# 可用性、可靠性、性能以及可支持性等。3.     测试策略

## 3.1               测试类型

### 3.1.1     数据和数据库完整性测试

|  |  |
| --- | --- |
| 测试目标： | 确保数据库访问方法和进程正常运行，数据不会遭到损坏。 |
| 技术： | 1、调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据（或对数据的请求）。  2、检查数据库，确保数据已按预期的方式填充， 并且所有的数据库事件都已正常发生；或者检查所返回的数据，确保为正当的理由检索到了正确的数据。 |
| 完成标准： | 所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏 |
| 需考虑的特殊事项： | 1、测试可能需要 DBMS 开发环境或驱动程序在数据库中直接输入或修改数据。  2、进程应该以手工方式调用。  3、应使用小型或最小的数据库（记录的数量有限） 来使所有无法接受的事件具有更大的可视度。 |

### 3.1.2     功能测试

|  |  |
| --- | --- |
| 测试目标： | 确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等功能。 |
| 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：  1、在使用有效数据时得到预期的结果。  2、在使用无效数据时显示相应的错误消息或警告消息。  3、各业务规则都得到了正确的应用。 |
| 完成标准： | 所计划的测试已全部执行，  所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | 1. 对需求理解不充分可能导致测试结果的不可信。   2、GUI设计的不合理会影响软件测试的进程。 |

### 3.1.3     用户界面测试

[用户界面 (UI) 测试用于核实用户与软件之间的交互。 UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。 另外，UI 测试还可确保 UI 中的对象按照预期的方式运行，并符合公司或行业的标准。]

|  |  |
| --- | --- |
| 测试目标： | 核实以下内容：  1、通过测试对象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 健、鼠标移动、 和快捷键）的使用。  2、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。 |
| 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。 |
| 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。 |
| 需考虑的特殊事项： | 并不是所有定制或第三方对象的特征都可访问。 |

### 3.1.4     性能评测

[性能评测是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。 性能评测的目标是核实性能需求是否都已满足。 实施和执行性能评测的目的是将测试对象的性能行为当作条件（例如工作量或硬件配置）的一种函数来进行评测和微调。

注： 以下所说的事务是指“逻辑业务事务”。 这种事务被定义为将由系统的某个 Actor 通过使用测试对象来执行的特定用例，例如，添加或修改给定的合同。]

|  |  |
| --- | --- |
| 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：  1、正常的预期工作量  2、预期的最繁重工作量 |
| 技术： | 1、使用为功能或业务周期测试制定的测试过程。  2、通过修改数据文件来增加事务数量。  3、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），并在多个客户机上重复。 |
| 完成标准： | 1、单个事务或单个用户：在每个事务所预期或要求的时间范围内成功地完 成测试脚本，没有发生任何故障。  2、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。 |
| 需考虑的特殊事项： | 1、综合的性能测试还包括在服务器上添加后台工作量。  2、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。  3、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。 |

### 3.1.5     负载测试

[负载测试是一种性能测试。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。 负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。 此外，负载测试还要评估性能特征，例如，响应时间、事务处理速率和其他与时间相关的方面。]

[注： 以下所说的事务是指“逻辑业务事务”。 这种事务被定义为将由系统的某个最终用户通过使用应用程序来执行的特定功能，例如，添加或修改给定的合同。]

|  |  |
| --- | --- |
| 测试目标： | 核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。 |
| 技术： | 1、使用为功能或业务周期测试制定的测试。  2、通过修改数据文件来增加事务数量， 或通过修改测试来增加每项事务发生的次数。 |
| 完成标准： | 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，没有发生任何故障。 |
| 需考虑的特殊事项： | 1、负载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。  2、负载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。 |

### 3.1.6     强度测试

[强度测试是一种性能测试，实施和执行此类测试的目的是找出因资源不足或资源争用而导致的错误。如果内存或磁盘空间不足，测试对象就可能会表现出一些在正常条件下并不明显的缺陷。而其他缺陷则可能由于争用共享资源（如数据库锁或网络带宽）而造成的。强度测试还可用于确定测试对象能够处理的最大工作量。]

[注： 以下提到的事务都是指逻辑业务事务。]

|  |  |
| --- | --- |
| 测试目标： | 确定和记录那些使系统无法继续正常运行的情况或条件。 |
| 技术： | 1、使用为性能评测或负载测试制定的测试。  2、要对有限的资源进行测试，就应该在一台计算机上运行测试，而且应该减少或限制服务器上的RAM 和DASD。  3、对于其他强度测试，应该使用多台客户机来运行相同的测试或互补的测试，以产生最繁重的事务量或最差的事务组合。 |
| 完成标准： | 所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。 |
| 需考虑的特殊事项： | 1、如果要增加网络工作强度，可能会需要使用网络工具来给网络加载消息或信息包。  2、应该暂时减少用于系统的DASD，以限制数据库可用空间的增长。  3、使多个客户机对相同的记录或数据账户 同时进行的访问达到同步。 |

### 3.1.7     安全性和访问控制测试

[安全性和访问控制测试侧重于安全性的两个关键方面：

 应用程序级别的安全性，包括对数据或业务功能的访问

 系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性可确保：在预期的安全性情况下，Actor 只能访问特定的功能或用例，或者只能访问有限的数据。 例如，可能会允许所有人输入数据，创建新账户，但只有管理员才能删除这些数据或账户。 如果具有数据级别的安全性，测试就可确保“用户类型一”能够看到所有客户消息（包括财务数据），而“用户二”只能看见同一客户的统计数据。

系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。]

|  |  |
| --- | --- |
| 测试目标： | 1、应用程序级别的安全性：核实参与者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。  2、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的参与者才能访问系统和应用程序。 |
| 技术： | 应用程序级别的安全性：  1、确定并列出各用户类型及其被授权访问的功能或数据。  2、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。  3、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：  必须与相应的网络或系统管理员一起对系统访问权进行检查和讨论。 |
| 完成标准： | 各种已知的Actor 类型都可访问相应的功能或数据，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。 |
| 需考虑的特殊事项： | 由于此测试可能是网络管理或系统管理的职能，可能会不需要执行此测试。 |

### 3.1.8    配置测试

[配置测试核实测试对象在不同的软件和硬件配置中的运行情况。 在大多数生产环境中，客户机工作站、网络连接和数据库服务器的具体硬件规格会有所不同。 客户机工作站可能会安装不同的软件例如，应用程序、驱动程序等而且在任何时候，都可能运行许多不同的软件组合，从而占用不同的资源。]

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象可在所需的硬件和软件配置中正常运行。 |
| 技术： | 1、使用功能测试脚本。  2、在测试过程中或在测试开始之前，打开各种与非测试对象相关的软件，然后将其关闭。  3、执行所选的事务，以模拟参与者与测试对象软件和非测试对象软件之间的交互。  4、重复上述步骤，尽量减少客户机工作站上的常规可用内存。 |
| 完成标准： | 对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。 |
| 需考虑的特殊事项： | 1、需要、可以使用并可以通过桌面访问哪种非测试对象软件。  2、通常使用的是哪些应用程序。  3、应用程序正在运行什么数据。  4、作为此测试的一部分，应将整个系统、Netware、网络服务器、 数据库等都记录下来. |

### 3.1.9     安装测试

[安装测试有两个目的。第一个目的是确保该软件在正常情况和异常情况的不同条件下例如，进行首次安装、升级、完整的或自定义的安装都能进行安装。异常情况包括磁盘空间不足、缺少目录创建权限等。第二个目的是核实软件在安装后可立即正常运行。这通常是指运行大量为功能测试制定的测试。]

|  |  |
| --- | --- |
| 测试目标： | 核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中：  1、首次安装。以前从未安装过ASM模拟软件的新计算机。  2、更新。以前安装过相同版本的ASM模拟软件的计算机。  3、更新。以前安装过ASM模拟软件的较早版本的计算机。 |
| 技术： | 1、手工开发脚本或开发自动脚本，以验证目标计算机的状况。  2、启动或执行安装。  3、使用预先确定的功能测试脚本子集来运行事务。 |
| 完成标准： | ASM模拟软件事务成功执行，没有出现任何故障。 |
| 需考虑的特殊事项： | 应该选择ASM模拟软件的哪些事务才能准确地测试出ASM模拟软件应用程序已经成功安装，而且没有遗漏主要的软件构件。 |

## 3.2     工具（暂定）

此项目将使用以下工具：

|  |  |  |
| --- | --- | --- |
|  | 工具 | 产商/自产 |
| 测试管理 | RequisitePro | Rational |
| 缺陷跟踪 | ClearQuest | Rational |
| 用于功能性测试的 ASQ 工具 | Rational Robot | Rational |
| 用于性能测试的 ASQ 工具 | Visual Quantify | Rational |
| 测试覆盖监测器或评测器 | Visual Pure Coverage | Rational |
| 项目管理 | Project Word Excel | miscrosoft |

# 4.     资源

## 4.1     角色

下表列出了在此项目的人员配备方面所作的各种假定。

[注： 可适当地删除或添加角色项。]

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 所推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理，  测试项目经理 | 2 | 进行管理监督。  职责：           提供技术指导           获取适当的资源           提供管理报告 |
| 测试设计员 | 1 | 确定测试用例、确定测试用例的优先级并实施测试用例。  职责：           生成测试计划           生成测试模型           评估测试工作的有效性 |
| 测试员 | 2 | 执行测试。  职责：           执行测试           记录结果           从错误中恢复           记录变更请求 |
| 测试系统管理员 | 1 | 确保测试环境和资产得到管理和维护。  职责：           管理测试系统           分配和管理角色对测试系统的访问权 |
| 数据库管理员 | 1 | 确保测试数据（数据库）环境和资产得到管理和维护。  职责：           管理测试数据（数据库） |
| 设计员 | 1 | 确定并定义测试类的操作、属性和关联。  职责：           确定并定义测试类           确定并定义测试包 |
| 实施员 | 2 | 实施测试类和测试包，并对它们进行单元测试。  职责：           创建在测试模型中实施的测试类和测试包 |

## 4.2     系统

下表列出了测试项目所需的系统资源。

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 |  |
| -网络或子网 | TBD |
| -服务器名称 | TBD |
| -数据库名称 | TBD |
| 客户端测试 PC |  |
| -包括特殊的配置需求 | TBD |
| 测试存储库 |  |
| -网络或子网 | TBD |
| -服务器名称 | TBD |
| 测试开发 PC | TBD |

# 5.     项目里程碑

[对 <项目名称> 的测试应包括上面各节所述的各项测试的测试活动。 应该为这些测试确定单独的项目里程碑，以通知项目的状态和成果。]

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作** | **开始日期** | **结束日期** |
| 制定测试计划 | 7天 | 2022.10.10 | 2022.10.16 |
| 设计测试 | 7天 | 2022.10.17 | 2022.10.23 |
| 实施测试 | 8天 | 2022.10.24 | 2022.10.31 |
| 执行测试 | 14天 | 2022.11.01 | 2022.10.14 |
| 对测试进行评估 | 14天 | 2022.11.15 | 2022.11.28 |

# 6.     可交付工件

[本节列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。]

## 6.1     测试模型

## 由测试设计员创建，预计于2022.10.23交付给测试员

## 6.2     测试记录

## 由测试员创建，预计于2022.10.31交付给测试经理

## 6.3     缺陷报告

# 由测试经理创建，预计于2022.11.20交付给项目经理7.     附录 A：项目任务

以下是一些与测试有关的任务：

 制定测试计划

-          确定测试需求

-          评估风险

-          制定测试策略

-          确定测试资源

-          创建时间表

-          生成测试计划

 设计测试

- 准备工作量分析文档

- 确定并说明测试用例

- 确定测试过程，并建立测试过程的结构

- 复审和评估测试覆盖

 实施测试

- 记录或通过编程创建测试脚本

- 确定设计与实施模型中的测试专用功能

- 建立外部数据集

 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

 对测试进行评估

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准