# 然之协同办公系统

测试计划

河北师范大学软件学院

16级X 班

编写

2018年

目录

[然之协同办公系统 1](#_Toc531090243)

[测试计划 1](#_Toc531090244)

[1 简介 3](#_Toc531090245)

[1.1目的 3](#_Toc531090246)

[1.2背景 3](#_Toc531090247)

[1.3范围 3](#_Toc531090248)

[2测试参考文档和测试提交文档 5](#_Toc531090249)

[2.1测试参考文档 5](#_Toc531090250)

[2.2测试提交文档 5](#_Toc531090251)

[3测试进度 5](#_Toc531090252)

[3.1.1进度安排 5](#_Toc531090253)

[3.1.2暂停标准和恢复要求 7](#_Toc531090254)

[4测试资源 7](#_Toc531090255)

[4.1人力资源 7](#_Toc531090256)

[4.2测试环境 8](#_Toc531090257)

[4.3测试工具 8](#_Toc531090258)

[5系统风险及优先级 8](#_Toc531090259)

[6测试策略 9](#_Toc531090260)

[6.1单元测试 9](#_Toc531090261)

[6.2集成测试 9](#_Toc531090262)

[6.3功能测试 9](#_Toc531090263)

[6.4性能测试 9](#_Toc531090264)

[7与测试有关的任务 10](#_Toc531090265)

## 1 简介

### 1.1目的

本测试计划文档作为指导然之协同办公系统测试项目秩序渐进的基础，帮助测试团队安排合适的资源和进度，避免可能的风险。本文档有助于实现以下目标：

1. 确定现有项目的信息和应测试的功能模块。
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，并对这些策略加以详细说明。
4. 确定所需的资源，并对测试的工作量进行估计。
5. 列出测试目的可交付元素，包括用例以及测试报告等。

### 1.2背景

然之协同系统是一款面向中小企业的协同办公系统，它包括了客户管理、销售跟踪、日常办公等功能，专注于提供一体化、精简的解决方案。

然之协同办公系统的主要功能：此系统主要是将客户管理系统、日常办公系统、现金记账功能、团队分享功能和应用导航组合成一个一体化的、精简的企业内部管理系统。其由五个主要功能模块组成，1. 客户管理（crm），主要为销售团队使用，包括客户资料的维护、订单的跟踪维护、合同的跟踪维护以及联系人的跟踪和维护；2. 日常办公(oa)，包含项目管理、公告管理和文档管理三大功能。3. 现金记账(cash)，用来记录公司日常的收入和支出，以掌握公司的运营情况。4. 团队分享(team)，包括论坛、博客、通讯录等功能，满足团队内部的交流和分享的目的。5. 应用导航(ips)，将企业自己的管理系统和工具添加为然之系统的应用并统一进行管理。

系统的性能：系统的模块操作过程比较方便，易用性高，存储过程为主要的操作方法，实现代码的简洁化和通用性，进一步提高系统的运行速率，减少等待和出错率。

测试对象的构架：被测程序按照主要功能分为五大模块：客户管理模块，日常办公模块，现金记账模块，团队分享模块和应用导航模块。针对此系统的模块性比较强，测试过程以模块为单位进行测试，包括各模块中的核心代码测试、界面测试和数据库测试。

### 1.3范围

#### 1.3.1总体描述

由于各种活动的相互影响和制约，系统的设计完成中可能存在某些错误，软件测试主要是对“然之”办公系统进行全面检查，及时发现程序中的逻辑错误，以保证产品的正确性和可靠性。

具体结合到系统操作，基本测试内容如下：

1. 易用性，即人机界面的交互
2. 性能，即检查快速载入和导出数据、检查系统的响应等
3. 功能，即用户在系统中可以进行的各种操作
4. 业务规则，即检查需求说明对业务流程的描述是否准确、考虑与目标用户的业务环境是否契合等
5. 事务准确性，即确保事务正确完成、确保被取消的事务回滚正确等
6. 数据有效性与完整性，即检查数据的格式是否正确、确保字符集适当等
7. 系统可靠性，即检查程序、数据库的故障处理能力，数据恢复能力等

#### 1.3.2详细描述

将测试主要分为4个阶段：

第一阶段：

完善文档。为了确保测试工作的正常运行，需先要完善需求规格说明书和概要设计两个重要的测试参考文档。

第二阶段：

将整个办公程序按功能分为5个模块（客户管理；日常办公；现金记账；团队分享；应用导航），分别对每个模块进行单元测试，在该阶段测试团队在每个单独的程序块中消除场内的逻辑、功能上的缺陷和错误，保证每个模块作为一个单元能正确执行，并为上一级测试做准备：采用自底向上的方法免去桩模块开发。高质量的程序单元是组成可靠系统的基础，对单元性能的评价有：1、单元接口，2、局部数据结果，3、重要的执行路径测试，4、出错处理路径，5、影响上述几点的边界条件测试。

第三阶段：

进行集成测试，采用自底向上的集成方法，将各个模块进行集中和装配，形成一个完整的软件后就可以进行集成测试，集成测试除了进一步检测和排除子系统结构或相应程序结构上的错误之外，还应该验证所有的系统单元配合是否合适、整体性能和功能是否完整。

第四阶段：

进行系统测试。首先对整个程序进行功能测试，验证软件的功能和性能以及其他特点是否与用户的需求一致。然后进行系统非功能测试，结合本系统特点，测试团队侧重于易用性测试，性能测试，强度测试，可靠性测试，可恢复性测试。表示层测试、业务测试，数据层测试，比较全面对系统进行测试。

#### 1.3.3相关风险

本次测试过程中，可能出现的风险如下：

1. Bug的修复情况
2. 模块功能的实现情况
3. 系统整体功能的实现情况
4. 代码的编写质量
5. 人员经验以及对测试软件的熟悉度
6. 开发人员、测试人员关于项目约定的执行情况
7. 人员调整导致研发周期延迟
8. 开发时间的缩短导致某些测试计划无法执行

## 2测试参考文档和测试提交文档

### 2.1测试参考文档

概要设计说明书

需求规格说明书

### 2.2测试提交文档

单元测试报告

集成测试报告

系统测试报告

验收测试报告

（注：其报告包括测试计划、设计规格、测试用例、用例脚本、执行日志、测试报告）

## 3测试进度

### 3.1.1进度安排

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 标注 | 3.1单元测试 | 工作日 | 开始时间 | 结束时间 | 测试人员 |
|  | 3.1.1各单元是否达到预期 | 3 | 18/11/21 | 18/11/23 | A，B |
|  | 3.1.2修改代码 | 1 | 18/11/23 | 18/11/24 | A，B |
|  | 3.1.3再次测试代码 | 2 | 18/11/25 | 18/11/26 | A，B |
|  | 3.1.4生成测试报告 | 2 | 18/11/27 | 18/11/28 | A，B |
|  | **3.2集成测试** | **5** | **18/11/29** | **18/12/3** | B，C |
|  | 3.2.1各单元集成后是否符合规范 | 1.25 | 18/11/29 | 18/11/30 | B，C |
|  | 3.2.2修改代码 | 1.25 | 18/11/30 | 18/12/1 | B，C |
|  | 3.2.3再次测试代码 | 1.25 | 18/12/1 | 18/12/2 | B，C |
|  | 3.2.4生成测试报告 | 1.25 | 18/12/2 | 18/12/3 | B，C |
|  | **3.3功能测试** | **5** | **18/12/4** | **18/12/8** | C，D |
|  | 3.3.1编写测试脚本并定义预期结果 | 1.25 | 18/12/4 | 18/12/5 | C，D |
|  | 3.3.2修改代码 | 1.25 | 18/12/5 | 18/12/6 | C，D |
|  | 3.3.3再次测试代码 | 1.25 | 18/12/6 | 18/12/7 | C，D |
|  | 3.3.4生成测试报告 | 1.25 | 18/12/7 | 18/12/8 | C，D |
|  | **3.4压力测试** | **5** | **18/12/9** | **18/12/13** | D，E |
|  | 3.4.1编写测试脚本并定义预期结果 | 1.25 | 18/12/9 | 18/12/10 | D，E |
|  | 3.4.2修改代码 | 1.25 | 18/12/10 | 18/12/11 | D，E |
|  | 3.4.3再次测试代码 | 1.25 | 18/12/11 | 18/12/12 | D，E |
|  | 3.4.4生成测试报告 | 1.25 | 18/12/12 | 18/12/13 | D，E |
|  | **3.5性能测试** | **6** | **18/12/14** | **18/12/19** | E，F |
|  | 3.5.1编写测试脚本并定义预期结果 | 2.4 | 18/12/14 | 18/12/16 | E，F |
|  | 3.5.2修改代码 | 1.2 | 18/12/16 | 18/12/17 | E，F |
|  | 3.5.3再次测试代码 | 1.2 | 18/12/17 | 18/12/18 | E，F |
|  | 3.5.4生成测试报告 | 1.2 | 18/12/18 | 18/12/19 | E，F |
|  | **3.6容错测试** | **6** | **18/12/20** | **18/12/25** | F，G |
|  | 3.6.1编写测试脚本并定义预期结果 | 2.4 | 18/12/20 | 18/12/22 | F，G |
|  | 3.6.2修改代码 | 1.2 | 18/12/22 | 18/12/23 | F，G |
|  | 3.6.3再次测试代码 | 1.2 | 18/12/23 | 18/12/24 | F，G |
|  | 3.6.4生成测试报告 | 1.2 | 18/12/24 | 18/12/25 | F，G |

### 3.1.2暂停标准和恢复要求

暂停准则：

1. 软件产品在单元、集成、确认、系统、安装、验收测试时，发现一级BUG和二级BUG的错误将暂停测试并返回开发。
2. 软件产品在单元、集成、确认、系统、安装、验收测试，分别达到软件产品在单元、集成、确认、系统、安装、验收测试停止标准。
3. 软件产品通过验收测试，并得到验收测试结论。
4. 软件产品需暂停以进行调整时，测试应随之暂停，并备份暂停点测试数据。
5. 软件产品在其开发生命周期内出现重大估计，进度偏差，需暂停或终止时，测试应随之暂停或终止，并备份暂停点测试数据。

恢复要求：

恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内修正错误并重新启动系统。恢复测试首先要采用各种办法强迫系统失败，然后验证系统是否能尽快恢复。对于自动回复需验证重新初始化、检查点、数据恢复和重新启动等机制的正确性；对于任工干预的恢复系统，还需估测平均修复时间，确定其是否在可接受的范围内。

## 4测试资源

### 4.1人力资源

#### 4.1.1项目总体人员分配

|  |  |  |
| --- | --- | --- |
| 角色 | 分配的角色 | 具体职责 |
| 测试人员 | B，C，D，E，F，G | 制定维护测试计划，设计测试用例与测试进程，搭建测试环境，执行集成测试、功能测试、系统测试、验收测试，提交测试报告 |
| 审查人员 | A | 单元测试；监督测试，并协助修改代码 |

#### 4.1.2各项测试人员分配

|  |  |  |
| --- | --- | --- |
| 任务 | 主要负责人 | 所需工具 |
| 1、需求复查 | A | 需求文档，概要设计 |
| 2、单元测试 | B |  |
| 3、集成测试 | C |  |
| 4、功能测试 | D |  |
| 5、压力测试 | E |  |
| 6、性能测试 | F |  |
| 7、容错测试 | G |  |

### 4.2测试环境

描述测试的软件环境和硬件环境，要求给出所使用的所有软件、硬件的CPU、内存、硬盘等情况，以及软硬件的部署情况

系统：Windows 10

测试工具：Eclipse ，PyCharm ，Postman ，Fiddler4，LoadRunner等。

硬件要求：CPU主频2.5GHz以上

内存4G以上

硬盘40G以上

### 4.3测试工具

|  |  |  |
| --- | --- | --- |
| **用途** | **工具** | **使用者** |
| 单元测试 | Eclipse，PostMan，Fiddler4 | A，B |
| 集成测试 | Eclipse，PostMan | B，C，D |
| 功能测试 | 手动 | D，E，F |
| 系统测试 | LoadRunner | G，A，C |

## 5系统风险及优先级

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 风险描述 | 责任人 | 优先级 | 规避措施 | 风险应急计划 |
| 1 | 人员无法及时到位 | 项目经理 | 高 | 定期督促测试成员，提前选好候补人员 | 推迟进度计划，提高工作效率。 |
| 2 | 人员技能不符合要求 | 项目经理 | 高 | 技能培训 | 提高培训质量，加强工作输出的监视。 |

## 6测试策略

### 6.1单元测试

测试目标：揭示出模块与其接口规格说明存在的矛盾。

测试范围：客户管理；日常办公；现金记账；团队分享；应用导航五个模块。

技术：白盒测试。

开始前提：需求测试完成，代码开发完成。

完成标准：所有单元测试用例都被执行过；所有发现的缺陷被修正并经过回归测试；所有被测对象的语句覆盖率达到100%，或能明确给出不需要达到的原因；单元测试报告被负责人批准。

测试重点和优先级：程序的逻辑错误。

### 6.2集成测试

测试目标：把各模块组合起来形成更大的功能组。

测试范围：针对然之办公系统的每个接口，考虑参数个数、参数的输入输出属性、参数的顺寻、参数的等价类情况、参数的边界情况、函数的返回值情况等。

技术：灰盒测试。

开始前提：单元测试完成。

完成标准：所有测试用例都被执行过；所有发现的缺陷被修正并经过回归测试；性能要求符合标准；集成测试报告被负责人批准。

测试重点和优先级：与数据库操作相关的测试。

需考虑的特殊事项：涉及客户交易等隐私的数据，需保证数据的保密性；集成测试处于灰色地带，过于白盒会出现单元测试重复，过分黑盒会出现系统测试重复的问题。

### 6.3功能测试

测试目标：保证所测试的每个独立模块的功能都是符合需要求的。

测试范围：从输入条件和输出条件结果来进行判断是否满足程序设计的需求。

技术：等价类划分，边界值分析，因果图分析，错误猜测方法。

开始前提：概要设计完成后。

完成标准：所有测试用例都被执行并通过；所有发现的缺陷都是修正并回归测试过；功能要求符合标准；功能测试报告被负责人批准。

测试重点和优先级：需求规格说明书。

### 6.4性能测试

测试目标：证明程序不能实现其目标。

测试范围：在特定负载和配置环境下程序的响应时间和吞吐量。

技术：特定工具，模拟超常的数据量、负载等，监测系统的各项性能指标。

开始前提：概要设计完成后。

完成标准：所有测试用例都被执行并通过；所有发现的缺陷都是修正并回归测试过；性能要求符合标准；功能测试报告被负责人批准。

需考虑的特殊事项：搭建苛刻的硬件和软件测试设备环境。

## 7与测试有关的任务

* 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
* 确定并说明测试用例
* 确定测试过程，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
* 核实结果，调查意外结果
* 记录缺陷、