<项目名称>

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=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <6/26/2017> | <1.0> | <测试计划初始版本> | <于航，李新宇，唐天成，刘瑾旭> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

1.4 项目标识 4

2. 测试需求 4

3. 测试策略 4

3.1 测试类型 4

3.1.1 数据和数据库完整性测试 4

3.1.2 功能测试 4

3.1.3 业务周期测试 4

3.1.4 用户界面测试 4

3.1.5 性能评价 4

3.1.6 负载测试 4

3.1.7 强度测试 4

3.1.8 容量测试 4

3.1.9 安全性和访问控制测试 4

3.1.10 故障转移和恢复测试 4

3.1.11 配置测试 4

3.1.12 安装测试 4

3.2 工具 4

4. 资源 4

4.1 角色 4

4.2 系统 4

5. 项目里程碑 4

6. 可交付工件 4

6.1 测试模型 4

6.2 测试日志 4

6.3 缺陷报告 4

7. 附录 A：项目任务 4

测试计划

# 简介

## 目的

<边走边拍> 的这一“测试计划”文档有助于实现以下目标:

为软件可靠性与安全性的评估提供依据

提供系统化、规范化、工程化、实用化的测试技术规范，尽早发现故障。

检查软件对误操作的处理能力

此计划适用于系统中各组成部分的软件测试工作。

验证软件是否满足软件设计所规定的技术要求

测试工作包括：

数据和数据库完整性测试 4

功能测试 4

业务周期测试 4

用户界面测试 4

性能评价 4

负载测试 4

强度测试 4

容量测试 4

安全性和访问控制测试 4

故障转移和恢复测试 4

配置测试 4

安装测试 可交付元素：

测试计划

测试用例

测试报告

## 背景

主要功能特性：

注册与登录

拍摄照片

照片分享

在地图上标注照片

在地图上动态展示行走轨迹和照片

评论与交流

转发时撰写文字内容

不同时间上同一位置的照片对比

用户管理

对照片和用户进行简单的分析。

测试对象的架构：前端采用MVP三层架构，model层建立路线和照片信息的模型，view层展示路线和照片等功能性信息以及好友管理、个性设置等界面展示，present层协助activity对view和model进行处理；后端采用SSH架构，后端web子系统采用SSH的框架进行开发，使用Struts实现MVC，用Spring完成IoC，由hibernate实现ORM。其中，Controller层包含了Struts Action；Model层包含了Service，Service.impl，Dao，Dao.impl和Entity；View层包含了JSP，CSS，JS，Image和Fonts。

项目的简史：无

## 范围

功能测试：

目标：确保测试的功能正常，其中包括注册登录、拍摄照片、分享照片等。

范围：测试各个模块以及窗口所完成的功能是否准确、数据是否正确

技术：边界值、等价类划分法。

集成测试：

目标：确保模块连接起来能够正常工作，暴露出一些局部反映不出的问题。

范围：将所有模块按照设计要求组成系统，进行集成测试。

技术：自底向上集成测试

性能测试：

目标：确定在各种工作负载下系统的性能。

范围：整个系统。

技术：基准测试、容量规划测试。

验收测试：

目标：确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

范围：完成了单元测试、集成测试、系统测试之后的软件产品。

技术：配置审核、程序测试。

影响测试设计、开发或实施的所有风险或意外事件:

1. 进度风险:时间紧迫，开发任务较为繁重，且小组成员初次进行项目开发，进度上存在一定程度风险。
2. 技术风险:由于初次进行android项目开发，小组成员没有相关经验，技术上存在不足

影响测试设计、开发或实施的所有约束：

1. 服务器操作系统为windows server 2007, 内存为1GB， 过大的并发访问量会使系统崩溃
2. 系统使用Java编写，数据库使用Mysql和MongoDB
3. 在Android studio/Eclipse环境下进行Java，JS，HTML代码编写
4. 首先使用Java完成app和web子系统的基本代码编写，然后从外部导入数据，最后进行整合调试
5. 测试在Windows 环境下进行完后，与 MySQL 和MongoDB结合后必须在服务器上进行调试，测试

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

[注：可以视情况删除或添加项目。]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 功能性规约 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 用例报告 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 项目计划 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 设计规约 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 原型 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 用户手册 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 业务模型或业务流程 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 数据模型或数据流 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 业务功能和业务规则 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 项目或业务风险评估 | 🞏 是 🞏 否 | 🞏 是 🞏 否 |  |  |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

|  |  |  |
| --- | --- | --- |
| **用例** | **功能性需求** | **非功能性需求** |
| 拍摄照片 | 用户在app中拍摄照片，并添加到我的路线中，或不加入路线只记录地理位置信息 | app打开系统相机的响应时间小于1s |
| 美化照片 | 用户在拍摄照片后对照片进行裁剪等处理 | app响应客户时间不超过2s |
| 录制路线 | 用户使用本app录制一条完整的可以分享的路线 | app的响应时间小于2s |
| 查看路线及图片 | 用户查看路线以及标注在上面的图片 | app响应时间小于2秒 |
| 分享路线及图片到微信群 | 用户可以分享自己保存的路线及照片到微信群 | app的响应时间小于3秒 |
| 查看其他人的路线及图片 | 用户在app上查看其他人分享的路线以及照片 | 系统响应时间小于3秒 |
| 评论其他人的路线及照片 | 用户可以评论或点赞其他人分享的路线及照片，用户通过评论区进行交流 | app的响应时间小于1秒 |
| 好友信息管理 | 用户进行添加好友、修改信息、删除好友的操作 | app的响应时间小于3秒 |
| 撰写美文或小诗 | 系统管理员查看被浏览次数，被评论次数前几名的分享，以及当前系统默认的最佳分享，从中更改最佳的分享 | 系统的响应时间小于3s |
| 在本app中分享路线和照片 | 用户将正在查看的路线和照片分享到本app内，有好友分享，大众分享和私信分享三种方式 | 系统的响应时间小于3s |
| 用户信息管理 | 系统管理员进行信息用户管理，对用户信息进行增删改查的操作。 | 系统每次的响应时间不超过5秒 |
| 查看统计分析 | 系统管理员查看用户分享的情况，并且获得系统对于用户分享详细情况的统计分析 | 系统管理员点击“进行深度统计分析”后，响应时间小于5分钟 |
| 管理最佳的分享 | 系统管理员查看被浏览次数，被评论次数前几名的分享，以及当前系统默认的最佳分享，从中更改最佳的分享 | 系统的响应时间小于5s |
| 登录 | 用户登录“边走边拍”app | 系统的响应时间小于2s |
| 用户信息管理 | 系统管理员进行信息用户管理，对用户信息进行增删改查的操作 | 系统每次的响应时间不超过5秒 |

# 测试策略

[测试策略提供了推荐用于测试对象的方法。上一节“测试需求”中说明了将要测试哪些对象，而本节则要说明如何对测试对象进行测试。

对于每种测试，都应提供测试说明，并解释其实施和执行的原因。

如果不实施和执行某种测试，则应该用一句话加以说明，并陈述这样做的理由。例如，“将不实施和执行该测试。。该测试不合适。”

制定测试策略时所考虑的主要事项有：将要使用的方法以及判断测试何时完成的标准。

下面列出了在进行每项测试时需考虑的事项，除此之外，测试还只应在安全的环境中使用已知的、受控的数据库来执行。 ]

## 测试类型

### 数据和数据库完整性测试

[数据库和数据库进程应作为<项目**名称**>中的子系统来进行测试。

在测试这些子系统时，不应将测试对象的用户界面用作数据的接口。对于数据库管理系统 (DBMS)，还需要进行深入的研究，以确定可以支持以下测试的工具和方法。]

|  |  |
| --- | --- |
| 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。 |
| 方法： | • [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。  • 检查数据库，确保数据已按预期的方式填充，并且所有 数据库事件都按正常方式出现；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： | • [测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接 输入或修改数据。  • 进程应该以手工方式调用。  • 应使用小型或最小的数据库（其中的记录数很有限）来 使所有无法接受的事件具有更大的可见性。] |

### 功能测试

[测试对象的功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面 (GUI) 与应用程序交互并分析输出结果来验证应用程序及其内部进程。以下列出的是每个应用程序推荐的测试方法概要：]

|  |  |
| --- | --- |
| 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。] |
| 方法： | [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。] |
| 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： | [确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）] |

### 业务周期测试

[业务周期测试应模拟在一段时间内对 <项目名称> 执行的活动。应先确定一段时间（例如一年），然后执行将在该时段内发生的事务和活动。这种测试包括所有的每日、每周和每月的周期，以及所有与日期相关的事件（如备忘录）。]

|  |  |
| --- | --- |
| 测试目标 | [确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。] |
| 方法： | [通过执行以下活动，测试将模拟若干个业务周期：  • 将修改或增强对测试对象进行的功能测试，以增加每项功能的执 行次数，从而在指定的时段内模拟若干个不同的用户。  • 将使用有效的和无效的日期或时段来执行所有与时间或日期相关 的功能。  • 将在适当的时候执行或启动所有周期性出现的功能。  • 在测试中还将使用有效的和无效的数据，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。 |
| 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： | • [系统日期和事件可能需要特殊的支持活动  • 需要通过业务模型来确定相应的测试需求和测试过程。] |

### 用户界面测试

[通过用户界面 (UI) 测试来核实用户与软件的交互。UI 测试的目标在于确保用户界面向用户提供了适当的访问和浏览测试对象功能的操作。除此之外，UI 测试还要确保 UI 功能内部的对象符合预期要求，并遵循公司或行业的标准。]

|  |  |
| --- | --- |
| 测试目标： | [核实以下内容：  • 通过浏览测试对象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  • 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。] |
| 方法： | [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： | [证实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。] |

### 性能评价

[性能评价是一种性能测试，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。性能评价的目标是核实性能需求是否都已满足。实施和执行性能评价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价和微调。

注：以下事务均指“逻辑业务事务”。这种事务被定义为将由系统的某个主角通过使用测试对象来执行的特定用例，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 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：  • 正常的预期工作量  • 预期的最繁重工作量] |
| 方法： | • [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 事务的迭代次数。  • 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），并在多台客户机（虚拟的或实际的客户机，请参见下面 的“需考虑的特殊事项”）上重复。] |
| 完成标准： | • [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。]  • [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： | [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“结构化查询语 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与实际大小相同或等比例缩放的数据库。] |

### 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：以下事务均指“逻辑业务事务”。这些事务被定义为将由系统的最终用户通过使用应用程序来执行的具体功能，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 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。] |
| 方法： | • [使用为功能或业务周期测试制定的测试。  • 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： | • [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。  • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数 据库。] |

### 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： | [核实测试对象能够在以下强度条件下正常运行，不会出现任何错误：  • 服务器上几乎没有或根本没有可用的内存（RAM 和 DASD）  • 连接或模拟了最大实际（或实际可承受）数量的客户机  • 多个用户对相同的数据/账户执行相同的事务  • 最繁重的事务量或最差的事务组合（请参见上面的“性能测 试”）。  注： 强度测试的目标还可表述为确定和记录那些使系统无法 继续正常运行的情况或条件。  客户机的强度测试在“配置测试”的第 3.1.11 节中进 行了说明。] |
| 方法： | • [使用为性能评价或负载测试制定的测试。  • 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。  • 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： | • [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。  • 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。  • 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。] |

### 容量测试

[容量测试使测试对象处理大量的数据，以确定是否达到了将使软件发生故障的极限。容量测试还将确定测试对象在给定时间内是否能够持续处理的最大负载或工作量。例如，如果测试对象正在为生成一份报表而处理一组数据库记录，那么容量测试就会使用一个大型的测试数据库，检验该软件是否正常运行并生成了正确的报表。]

|  |  |
| --- | --- |
| 测试目标： | [核实测试对象在以下大容量条件下能否正常运行：  • 连接（或模拟了）最大（实际或实际可承受）数量的客户机，所 有客户机在长时间内执行相同的、且情况（性能）最差的业务功 能。  • 已达到最大的数据库大小（实际的或按比例缩放的），而且同时 执行了多个查询或报表事务。] |
| 方法： | • [使用为性能评价或负载测试制定的测试。  • 应该使用多台客户机来运行相同的测试或互补的测试，以便在长 时间内产生最繁重的事务量或最差的事务组合（请参见上面的 “强度测试”）。  • 创建最大的数据库大小（实际的、按比例缩放的、或输入了代表 性数据的数据库），并使用多台客户机在长时间内同时运行查询 和报表事务。] |
| 完成标准： | • [所计划的测试已全部执行，而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。] |
| 需考虑的特殊事项： | [对于上述的大容量条件，哪个时段是可以接受的时间？] |

### 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，主角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，但只有经理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”能够看到所有客户信息（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： | * 应用程序级别的安全性：[*核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。]* * 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的主角才能访问系统和应用程序。]* |
| 方法： | * 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使用的功能或数据。*]*   • [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：*[请参见下面的“需考虑的特殊事项”]* |
| 完成标准： | [各种已知的主角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： | [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。由于此测试可能是网络管理或系统管理的职能，可能不需要执行此测试。] |

### 故障转移和恢复测试

[故障转移和恢复测试可确保测试对象能成功完成故障转移，并从硬件、软件或网络等方面的各种故障中进行恢复，这些故障导致数据意外丢失或破坏了数据的完整性。

故障转移测试可确保：对于必须始终保持运行状态的系统来说，如果发生了故障，那么备选或备份的系统就适当地将发生故障的系统“接管”过来，而且不会丢失任何数据或事务。

恢复测试是一种相反的测试流程。其中，将应用程序或系统置于极端的条件下（或者是模仿的极端条件下），以产生故障，例如设备输入/输出 (I/O) 故障或无效的数据库指针和关健字。启用恢复流程后，将监测和检查应用程序和系统，以核实应用程序或系统是正确无误的，或数据已得到了恢复。]

|  |  |
| --- | --- |
| 测试目标： | [确保恢复进程（手工或自动）将数据库、应用程序和系统正确地恢复到了预期的已知状态。测试中将包括以下各种情况：  • 客户机断电  • 服务器断电  • 通过网络服务器产生的通信中断  • DASD 和/或 DASD 控制器被中断、断电或与 DASD 和/ 或DASD 控制器的通信中断  • 周期未完成（数据过滤进程被中断，数据同步进程被中 断）。  • 数据库指针或关键字无效  • 数据库中的数据元素无效或遭到破坏] |
| 方法： | [应该使用为功能和业务周期测试创建的测试来创建一系列的事务。一旦达到预期的测试起点，就应该分别执行或模拟以下操作：  • 客户机断电：关闭 PC 的电源。  • 服务器断电：模拟或启动服务器的断电过程。  • 通过网络服务器产生的中断：模拟或启动网络的通信中 断（实际断开通信线路的连接或关闭网络服务器或路由 器的电源）。  • DASD 和 DASD 控制器被中断、断电或与 DASD 和 DASD 控制器的通信中断：模拟与一个或多个 DASD 控 制器或设备的通信，或实际取消这种通信。  一旦实现了上述情况（或模拟情况），就应该执行其他事务。而且一旦达到第二个测试点状态，就应调用恢复过程。  在测试不完整的周期时，所使用的方法与上述方法相同，只不过应异常终止或提前终止数据库进程本身。  对以下情况的测试需要达到一个已知的数据库状态。当破坏若干个数据库字段、指针和关键字时，应该以手工方式在数据库中（通过数据库工具）直接进行。其他事务应该通过使用“应用程序功能测试”和“业务周期测试”中的测试来执行，并且应执行完整的周期。] |
| 完成标准： | [在所有上述情况中，应用程序、数据库和系统应该在恢复过程完成时立即返回到一个已知的预期状态。此状态包括仅限于已知损坏的字段、指针或关键字范围内的数据损坏，以及表明进程或事务因中断而未被完成的报表。] |
| 需考虑的特殊事项： | • [恢复测试会给其他操作带来许多的麻烦。断开缆线连接 的方法（模拟断电或通信中断）可能并不可取或不可 行。所以，可能会需要采用其他方法，例如诊断性软件 工具。  • 需要系统（或计算机操作）、数据库和网络组中的资 源。  • 这些测试应该在工作时间之外或在一台独立的计算机上 运行。] |

### 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： | [核实测试对象可在要求的硬件和软件配置中正常运行。] |
| 方法： | • [使用功能测试脚本。  • 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（例如 Microsoft 应用程序：Excel 和 Word），然后将其关闭。  • 执行所选的事务，以模拟主角与测试对象软件和非测试 对象软件之间的交互。  • 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： | • [需要、可以使用并可以通过桌面访问哪种非测试对象软 件？  • 通常使用的是哪些应用程序？  • 应用程序正在运行什么数据？例如，在Excel 中打开的 大型电子表格，或是在 Word 中打开的 100 页文档。  • 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。] |

### 安装测试

[安装测试有两个目的。第一个目的是确保该软件能够在所有可能的配置下进行安装，例如，进行首次安装、升级、完整的或自定义的安装，以及在正常和异常情况下安装。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：  • 首次安装。以前从未安装过 <项目名称> 的新计算机  • 更新。以前安装过相同版本的 <项目名称> 的计算机  • 更新。以前安装过较早版本的 <项目名称> 的计算机 |
| 方法： | • [手工开发脚本或开发自动脚本，以验证目标计算机的状 况⎯ 新 - <项目名称> 从未安装过；已安装 <项目名称> 相同或较早版本）。  • 启动或执行安装。  • 使用预先确定的功能测试脚本子集来运行事务。] |
| 完成标准： | <项目名称> 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 <项目名称> 的哪些事务才能准确地测试出 <项目名称> 应用程序已经成功安装，而且没有遗漏主要的软件构件？] |

## 工具

此项目将使用以下工具：

[注：可以视情况删除或添加项目。]

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 测试管理 | Astra QuickTest | Mercury Interactive Winrunner | Mercury Interactive WinrunnerPro E19.1 |
| 缺陷跟踪 | TestDirector | Compuware | Compuware WIDGET ENGINE 3.8.2 ibuild 7.91 |
| 用于功能性测试的 ASQ 工具 | Rational Quantify | Rational | Rational ver 0.8.3 2002-2003 MYamaguchi |
| 用于性能测试的 ASQ 工具 | Rational Quantify Purify | Rational | Rational er 0.8.1 2002-2003 is freeware |
| 测试覆盖监测器或评价器 | Edition, EcoScope, TrackRecord | TestFactory | Editino Stable Release for Windowns XP Pro 0.8.3 |
| 项目管理 | Git/SourceTree | Atlassian | 2.1.2.5 |
| DBMS 工具 | Mysql | oracle | Mysql 5.1 |

# 资源

[本节列出推荐 <项目名称> 项目使用的资源，及其主要职责、知识或技能。]

## 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：可视情况删除或添加项目。]

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 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 | 1 | 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 | 4 | 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 | 2 | 确保测试环境和资产得到管理和维护。  职责：   * 管理测试系统 * 授予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 | 4 | 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 |

## 系统

下表列出了测试项目所需的系统资源。

[此时并不完全了解测试系统的具体元素。建议让系统模拟生产环境，并在适当的情况下减小访问量和数据库大小。]

[注：可以视情况删除或添加项目。]

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 | Windows Server 2008 R2 企业版 SP1 64位 |
| —网络或子网 | 115.159.74.217 |
| —服务器名服务器名 | tomcat |
| —数据库名 | bzbp |
| 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 | 115.159.74.217 |
| —服务器名服务器名 | TBD |
| 测试开发 PC | TBD |

# 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** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 48 | 2017/6/26 | 2017/7/5 |
| 设计测试 | 48 | 2017/7/6 | 2017/7/15 |
| 实施测试 | 16 | 2017/7/16 | 2017/7/20 |
| 执行测试 | 56 | 2017/7/21 | 2017/8/31 |
| 评估测试 | 56 | 2017/9/1 | 2017/9/10 |

# 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。]

## 测试模型

[本节确定将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用 ASQ 工具来创建或引用。]

V模型

## 测试日志

[说明用来记录和报告测试结果和测试状态的方法和工具。]

BareTail（工具）

## 缺陷报告

[本节确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。]

组织、重现、隔离、归纳、对比、总结、精简、消除歧义、中立、检查

# 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

- 确定并结构化测试过程

- 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

• 评估测试

- 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