“I Ask”爱问论坛软件设计与开发

**软件测试计划**

**版本：1.0**

编写： 吕炳旭

校对： 曲卓涵

审核： 马嘉伟

批准： 孙启星

**404 NOT FOUND工作室**

**2019年3月**

**目 录**

[1引言 4](#_Toc4691984)

[1.1文档标识 4](#_Toc4691985)

[1.2项目概述 4](#_Toc4691986)

[1.3文档概述 5](#_Toc4691987)

[1.4参考文档 5](#_Toc4691988)

[2 测试准备 6](#_Toc4691989)

[2.1 产品需求确认 6](#_Toc4691990)

[2.2 测试用例准备 7](#_Toc4691991)

[2.3 测试任务分配 7](#_Toc4691992)

[3 测试资源 7](#_Toc4691993)

[3.1 人力资源 7](#_Toc4691994)

[3.2 测试环境 8](#_Toc4691995)

[3.3 测试工具 9](#_Toc4691996)

[4 测试策略 10](#_Toc4691997)

[4.1 数据和数据库完整性测试 10](#_Toc4691998)

[4.2 接口测试 11](#_Toc4691999)

[4.3 集成测试 11](#_Toc4692000)

[4.4 功能测试 12](#_Toc4692001)

[4.5 用户界面测试 13](#_Toc4692002)

[**4.6 性能测试** 14](#_Toc4692003)

[**4.7 负载测试** 16](#_Toc4692004)

[**4.8 安全性和访问控制测试** 17](#_Toc4692005)

[**4.9 安装测试** 18](#_Toc4692006)

[**问题严重度描述** 19](#_Toc4692007)

[**测试结束标准** 19](#_Toc4692008)

[**单元测试结束标准** 19](#_Toc4692009)

[**集成测试停止标准** 19](#_Toc4692010)

[**系统测试结束标准** 20](#_Toc4692011)

[测试用例设计 20](#_Toc4692012)

# 1引言

## 1.1文档标识

中文名称：《软件测试计划》。

英文名称：“Software Testing Plan”。

文档版本：“1.0”。

文档编号：“4NF-IAsk-STP-1.0”。

## 1.2项目概述

本文档适用于“‘I Ask’爱问论坛软件设计与开发”项目（以下简称“IAsk项目”）的开发过程。IAsk项目由404 NOT FOUND工作室（以下简称“工作室”）提出并负责实施，该项目标识号为“4NF-IAsk”，其软件产品版本号为“1.0”，包括三个内部版本，分别是0.1版、0.2版和0.3版。

I Ask爱问论坛是一个基于WEB的论坛系统，它连接各个学校、不同专业的学生用户，主要为在校学生提供一个网络问答社区。在该论坛中，用户可以根据自己的需求去搜索相应的问答，或者发布自己的提问；亦可凭借自己的知识、见解对其他人的提问进行解答。该平台旨在促进用户分享彼此的知识、经验和见解，为中文互联网源源不断地提供多种多样的信息。

工作室将依据上述设想，深度定制IAsk项目的内部逻辑架构及人机界面，完成“I Ask”爱问论坛软件的设计与开发。

## 1.3文档概述

本文档依据《国标GB/T 8567-2006计算机软件文档编制规范》制定，属于技术文档，仅限于工作室的项目相关人员阅读。

软件项目的测试计划是描述测试目的、范围、方法和软件测试的重点等的文档。对于验证软件产品的可接受程度编写测试计划文档是一种有用的方式。此文档是为I Ask爱问论坛项目制定的软件测试计划，以保证测试工作和整个开发工作融合起来，且为资源和变更事件提供一个前期的可控制的风险计划。

## 1.4参考文档

* 《GB/T 8567-2006计算机软件文档编制规范》，国家标准
* 《4NF-IAsk-SDS-1.0 软件文档规范》，工作室编写

# 2 测试准备

## 2.1 产品需求确认

仔细阅读产品原型、需求文档、或者UI，反复和产品经理确认需求的细节，最好把需求拆分成一个个子需求。需求分析的越详细，对业务的理解程度就越高，对设计测试用例的帮助就越大，测试的过程中就更有目的性。

**（1）明确需求范围**

了解该需求是为了解决用户的什么问题，是功能性需求还是非功能性需求，明确需求背后所隐藏的需求，将问题在需求阶段暴露的成本最小。

**（2）画业务流程图**

根据需求中规定的业务流程，各业务流程分支的确定，并以流程图呈现出来。

**（3）功能点整理**

根据产品需求整理出有哪些功能点，包括业务功能、数据约束、易用性需求、编辑约束、权限需求等等。

**（4）提取测试点**

根据整理的思维导图，去提取每一个功能点中的细节需求，例如新增账户，在思维导图中，最小的颗粒度就到新增账户了，但是新增账户这个功能仍然有很多的需求点，账户姓名唯一性判定，手机号码是否必填等，这些更细的需求点组合起来就形成了测试需求文档。

**（5）确定测试范围**

需求的确定，并不代表测试范围就是该需求的范围，很有可能一个需求分多个软件版本来实现，最后确定哪些需求是需要测试的，以及测试目标的优先级。

## 2.2 测试用例准备

熟悉完产品需求后，需要准备测试用例。用例是测试工作的基础，用例设计的好坏直接会决定测试的质量。测试用例常见的设计方法有：等价类划分法、边界值分析法、错误推测法、判定表法、正交实验法。

## 2.3 测试任务分配

分配测试任务前，需对测试团队每个成员都非常了解，然后根据每个成员的情况进行测试任务分配。分配测试任务需考虑以下情况。

（1）测试人员水平不同，分配任务也应不同。

（2）重点模块需要重点关注，着重测试。

（3）试任务注意在测试人员中间互换，增加测试新鲜感。

（4）测试任务分配之前一定要与测试人员沟通。

（5）分配测试任务需明确每个测试任务的优先级。

# 3 测试资源

## 3.1 人力资源

下表列出了在此项目的测试人员配备方面所作的各种假定。

|  |  |  |
| --- | --- | --- |
| 角色 | 所推荐的最少资源数量 | 具体职责或注释 |
| Leader | 1人 | 负责测试项目的整体框架合和流程运作，同时负责为团队成员解决技术问题 |
| Tester | 3人 | 面向Windows平台，Mac平台和Android平台分别做浏览器的兼容性测试，功能测试，模块测试等 |
| Liaisonor | 1人 | 负责为团队解决非技术问题，负责和开发团队做好需求和功能沟通等 |

## 3.2 测试环境

|  |  |
| --- | --- |
| 软件环境（相关软件，操作系统等） | |
| Windows10 (IE8即以上IE版本浏览器 ， chrome浏览器 ， opera浏览器 ， Firefox浏览器 )  Mac (safari浏览器 ，chrome Mac版浏览器 ， opera for Mac浏览器)  手机浏览器（webkit内核）浏览器 | |
| 硬件环境（网络、设备等） | |
| 服务器环境 | CPU： Xeon X5570 2.9GHz四核双CPU配置  内存：DDRIII 1333MHz 12G(4G\*3)  硬盘：300G\*3以上（SAS,Raid5）  网卡：千兆网卡\*2  操作系统 ：Windows Server 2003 |
| 客户端环境 | CPU：Intel 奔腾4 3.0GHz ;Intel core 以上系列；AMD 64CPU以上  内存：>=1G Bytes  显示屏分辨率：1024\*768  网卡：100M |
| 网络环境 | >10M带宽 支持[1000-2000并发] |

## 3.3 测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| 用途 | 工具 | 生产厂商/自产 | 版本 |
| 控制测试过程，创建测试工作流的框架和基础 | TestDirector | Mercury Interactive | TestDirector 8.0 |
| Web应用开发测试系统 | Selenium | 开源软件 | Selenium 3.0 |
| 预测系统行为和性能工业标准级负载 | LoadRunner | Mercury Interactive(后被HP收购) | LoadRunner12 |

# 4 测试策略

## 4.1 数据和数据库完整性测试

在I Ask爱问论坛中，数据库和数据库进程应作为一个子系统来进行测试。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，而应该单独进行数据库测试。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技术。

|  |  |
| --- | --- |
| 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 |
| 测试范围： | 数据库，数据库管理系统 |
| 技术： | 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，并且所有的数据库事件已正常发生；或者检查所返回的数据，确保正当的理由检索到了正确的数据 |
| 开始标准： | 数据集与数据库搭建完成，进入上层开发时 |
| 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。 |
| 测试重点和优先级： | 数据结构是否合理，数据高并发访问是否可支撑，数据集合是否完整 |
| 需考虑的特殊事项： | 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。  进程应该以手工方式调用。  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。 |

## 4.2 接口测试

|  |  |
| --- | --- |
| 测试目标： | 确保接口调用的正确性 |
| 测试范围： | 所有软件、硬件接口，记录输入输出数据 |
| 技术： | 打桩测试技术，集成测试技术 |
| 开始标准： | 模块开发中期，已经具备接口测试条件时 |
| 完成标准： | 集成测试完成，接口描述与实际情况相符 |
| 测试重点和优先级： | 接口命名是否规范合理，接口输入输出数据是否符合描述 |
| 需考虑的特殊事项： | 接口冗余时记录冗余接口，并在测试阶段完成去掉冗余接口 |

## 4.3 集成测试

|  |  |
| --- | --- |
| 测试目标： | 检测需求中业务流程，数据流的正确性 |
| 测试范围： | 需求中明确的业务流程，或组合不同功能模块而形成一个大的功能。 |
| 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：  在使用有效数据时得到预期的结果。 |
| 开始标准： | 在完成某个集成测试时必须达到标准 |
| 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。 |
| 测试重点和优先级： | 测试重点指在测试过程中需着重测试的地方，优先级可以根据需求及严重来定 |
| 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） |

## 4.4 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： | 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。 |
| 技术： | Selenium自动化脚本测试技术 |
| 开始标准： | 系统集成阶段，模块测试完成后 |
| 完成标准： | 系统已经达到预期的所有功能 |
| 测试重点和优先级： | 以系统功能为重点，地毯式测试系统是否已到达预期的所有功能 |
| 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） |

## 4.5 用户界面测试

|  |  |
| --- | --- |
| 测试目标： | 确保窗口与窗口之间、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： | 系统的前端各个页面 |
| 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： | 功能测试全面完成后。 |
| 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。系统已经达到预期的所有访问浏览功能。 |
| 测试重点和优先级： | 以前端页面各组件交互、跳转方式为重点。 |
| 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。 |

**4.6 性能测试**

|  |  |
| --- | --- |
| 测试目标： | 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。 |
| 测试范围： | 完整系统的各部分功能。 |
| 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： | 功能测试，用户界面测试全面完成后。 |
| 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： | 侧重对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估 |
| 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 |

**4.7 负载测试**

|  |  |
| --- | --- |
| 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。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。 |
| 测试范围： | 完整系统的各部分功能。 |
| 技术： | 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，或通过修改脚本来增加每项事务发生的次数。 |
| 开始标准： | 性能测试全面完成后。 |
| 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。 |
| 测试重点和优先级： | 响应时间、事务处理速率和其他与时间相关的方面。 |
| 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实际大小或相同缩放比例的数据库。 |

**4.8 安全性和访问控制测试**

|  |  |
| --- | --- |
| 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的Actor才能访问系统和应用程序。 |
| 测试范围： | 应用程序级别和系统级别。 |
| 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。 |
| 开始标准： | 负载测试全面完成后。 |
| 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。 |
| 测试重点和优先级： | 应用程序级别的安全性和系统级别的安全性 |
| 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。 |

**4.9 安装测试**

将不实施该测试。由于该系统为web网页开发系统，无需安装，故该测试本项目不适用。

**问题严重度描述**

|  |  |  |
| --- | --- | --- |
| 问题严重度 | 描述 | 响应时间 |
| 高 | 数据库设计错误 | 2天 |
| 中 | 前端页面控件排列错乱 | 5小时 |
| 中 | 页面跳转逻辑错误 | 1天 |
| 低 | 用户修改个人信息无效 | 1天 |
| 低 | 管理员无法删除用户 | 3小时 |

**测试结束标准**

**单元测试结束标准**

1. 单元测试用例设计已经通过评审。
2. 按照单元测试计划完成了所有规定单元的测试。
3. 达到了测试计划中关于单元测试所规定的覆盖率的要求。
4. 被测试的单元每千行代码必须发现至少3 个错误（不含五级错误）。
5. 软件单元功能与设计一致。
6. 在单元测试中发现的错误已经得到修改，各级缺陷修复率达到标准。

**集成测试停止标准**

1. 集成测试用例设计已经通过评审。
2. 按照集成构件计划及增量集成策略完成了整个系统的集成测试。
3. 达到了测试计划中关于集成测试所规定的覆盖率的要求。
4. 被测试的集成工作版本每千行代码必须发现至少2个错误（不含五级错误）。
5. 集成工作版本满足设计定义的各项功能、性能要求。
6. 在集成测试中发现的错误已经得到修改，各级缺陷修复率达到标准。

**系统测试结束标准**

1) 系统测试用例设计已经通过评审；

2) 按照系统测试计划完成了系统测试；

3) 达到了测试计划中关于系统测试所规定的覆盖率的要求（新的系统，要求按测试用例来测试，覆盖所有用例后提交测试报告；新的版本，按开发的《测试申请单》来验证修复BUG和新增功能，然后进行回归测试，提交测试日简报）；

4) 软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求；

5) 在系统测试中发现的错误已经得到修改，各级缺陷修复率达到标准。