**WordPress项目测试计划书**

目录

[1.引言 3](#_Toc530475898)

[1.1测试目的 3](#_Toc530475899)

[1.2测试项目背景 3](#_Toc530475900)

[1.3有关项目人员组成以及联系方式 4](#_Toc530475901)

[2.任务概述 4](#_Toc530475902)

[2.1测试范围 4](#_Toc530475903)

[2.1.1Workpress项目测试范围的依据为： 4](#_Toc530475904)

[2.1.2测试范围 4](#_Toc530475905)

[2.2测试目标 6](#_Toc530475906)

[2.3测试交付物 6](#_Toc530475907)

[3.测试策略 8](#_Toc530475908)

[3.1测试方法 8](#_Toc530475909)

[3.1.1单元测试 8](#_Toc530475910)

[3.1.2集成测试 9](#_Toc530475911)

[3.1.3 系统测试 9](#_Toc530475912)

[3.1.4 性能测试 10](#_Toc530475913)

[3.2测试阶段性计划 10](#_Toc530475914)

[3.3测试停止及恢复条件 11](#_Toc530475915)

[4.测试资源 12](#_Toc530475916)

[4.1人力资源 12](#_Toc530475917)

[4.2测试环境： 12](#_Toc530475918)

[4.3测试工具 12](#_Toc530475919)

[５.风险评估 13](#_Toc530475920)

[5.1 人力方面 13](#_Toc530475921)

[5.2 时间方面 13](#_Toc530475922)

[5.3 资源方面 13](#_Toc530475923)

[5.4 部门合作方面 13](#_Toc530475924)

[5.5功能的实现情况 13](#_Toc530475925)

# 1.引言

## 1.1测试目的

“WordPress”的这一测试计划文档作为测试工作的指导性文件，为了实现资源的合理利用，避免资源浪费和风险。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软件构件。

2.推荐可采用的测试策略，并对这些策略加以说明。

3.确定所需的资源，并对测试的工作量进行估计。

4.列出测试项目的可交付元素，包括用例以及测试报告等。

## 1.2测试项目背景

WordPress是使用PHP语言开发的博客平台，用户可以在支持PHP和MySQL数据库的服务器上架设属于自己的网站。也可以把WordPress当作一个内容管理系统（CMS）来使用，用户可以在支持 PHP 和 MySQL数据库的服务器上使用自己的博客。

WordPress功能的实现主要由用户登录界面，注册用户操作（支持会员注册登录，后台管理功能），内容发布（提供文章、评论、分类等多种形式的RSS聚合，还具有评论的管理，垃圾信息过滤的功能），文档类别管理（提供链接的添加、归类功能，支持多样式CSS和PHP程序的直接编辑、修改），菜单定义组成（通过选择不同主题，方便地改变页面的显示效果，还可以通过添加插件，可提供多种特殊的功能）。

测试对象的框架：被测程序按照主要功能分为以下五大模块：用户登录模块，用户注册模块，内容发布模块，文档类别管理模块，菜单定义模块。针对此系统的模块性比较强，测试过程以模块为单元进行测试，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有关项目人员组成以及测试任务

|  |  |  |
| --- | --- | --- |
| **测试人员** | **第一阶段测试任务** | **第二阶段测试任务** |
| 卢丹琦 | 用户登录界面 | 系统测试 |
| 沈梦琪 | 注册用户操作 | 集成测试 |
| 史素佳 | 内容发布模块 | 性能测试 |
| 韦士杰 | 文档类别管理模块 | 系统测试 |
| 杨天莹 | 内容发布模块 | 集成测试 |
| 岳哲伟 | 文档类别管理模块 | 性能测试 |
| 刘 江 | 菜单定义模块 | 集成测试 |

# 2.任务概述

## 2.1测试范围

### 2.1.1Workpress项目测试范围的依据为：

1. 各子系统所包含的功能
2. 项目需求文档特别确定的测试范围

### 2.1.2测试范围

1. 功能测试：功能模块
2. 性能测试：检查快速载入和导出数据、检查系统的响应，多个功能同时使用时检查系统的切换响应速度。

|  |  |
| --- | --- |
| 测试内容 | 测试范围 |
| 单元测试 | 分模块进行单元测试  消除场内逻辑、功能上的缺陷  单元性能的评价：1.单元接口、2.局部数据结构、3.重要的执行路径测试、4.出错路径的处理、5.边界条件测试 |
| 集成测试 | 将各个模块进行集中和装配  检查和排除子系统结构或相应程序结构上的错误  系统单元配合是否合适  整体性能和功能是否完善  自底向上集成 |
| 功能测试 | 管理站点  登出  文章RSS、  评论RSS  WordPress.org  管理帖子、文档、档案、类别，管理屏幕， |
| 性能测试 | 一、数据量 以WordPress数据库中存在的记录为标准，测试如下性能数据： 1.修改文章、媒体、评论、用户等数据 2.同时在线评论文章的并发量 3.多少同时添加用户，用户信息是否一致冲突 4.用户的评论是否能跟实时更新 5.频繁提交评论，文章，是否可以及时更新 二、硬件配置 不同硬件配置对系统性能的影响 1.一般配置的性能（CPU：Intel core i5、内存4G） 2.在一般配置的基础上增加内存后的性能（CPU: Intel core i5、内存8G） 3.在一般配置的基础上升级CPU后的性能 |

3.客户模块：新增用户的管理，批量新增用户的处理。

4.安全测试：用户的个人隐私如密码是否会泄露。

## 2.2测试目标

1. 测试覆盖率：90%
2. 测试用例执行率：100%
3. Bug修复率：100%

## 2.3测试交付物

|  |  |
| --- | --- |
| **项目阶段** | **交付物名称** |
| 启动阶段 | 测试计划 |
| 测试用例 |
| 执行阶段 | 测试源代码 |
| 测试数据（保留为sql文本） |
| 监控阶段 | 错误分析报告 |
| 性能测试报告 |
| 项目阶段汇报 |
| 会议记录 |
| 会议签到表 |
| 收尾阶段 | 确认测试分析报告 |
| 遗留问题清单 |
| 测试总结报告 |
| 验收测试报告 |

# 3.测试策略

## 3.1测试方法

### 3.1.1单元测试

|  |  |
| --- | --- |
| 测试目标 | 使用java对页面中每一个单元模块编写测试类，逐一进行测试 |
| 测试范围 | 用户登录界面、注册用户操作、内容发布、文档类别管理、菜单定义五个模块 |
| 技术 | 白盒测试 |
| 开始标准 | 需求测试和代码开发完成 |
| 完成标准 | 所有单元测试用例都被执行并通过；所有发现的缺陷都被修正并回归测试过；单元测试报告被权签人批准 |
| 测试重点和优先级 | 每个类的逻辑错误 |
| 需要考虑的特殊事项 | 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理由 |

### 3.1.2集成测试

|  |  |
| --- | --- |
| 测试目标 | 使用java将单元测试构件按逻辑顺序依次执行 |
| 测试范围 | 每个接口参数的个数、输入输出属性、顺序、等价类和边界值，函数的返回值 |
| 技术 | 灰盒测试 |
| 开始标准 | 单元测试完成之后 |
| 完成标准 | 所有集成测试用例都被执行并通过；所有发现的缺陷都被修正并回归测试过；集成测试报告被权签人批准 |
| 测试重点和优先级 | 与数据库相关的测试 |
| 需要考虑的特殊事项 | 涉及到用户名密码，需要保证数据的保密性；集成测试属于灰盒测试，过分使用白盒测试会出现与单元测试重复的问题，过分使用黑盒测试会出现与系统测试重复的问题 |

### 3.1.3系统测试

|  |  |
| --- | --- |
| 测试目标 | 在浏览器端进行系统化测试 |
| 测试范围 | 从输入条件和输出结果来进行判断是否满足程序设计要求 |
| 技术 | 黑盒测试：等价类划分、边界值分析、因果图分析、错误猜测的方法 |
| 开始标准 | 集成测试完成之后 |
| 完成标准 | 所有系统测试用例都被执行并通过；所有发现的缺陷都被修正并回归测试过；功能要求符合标准；系统测试报告被权签人批准 |
| 测试重点和优先级 | WordPress程序符合规格标准 |
| 需要考虑的特殊事项 | 无 |

### 3.1.4性能测试

|  |  |
| --- | --- |
| 测试目标 | 验证软件系统是否能够达到用户提出的性能指标，优化软件进而优化系统 |
| 测试范围 | 在特定负载和配置环境下WordPress每个功能执行的响应时间和吞吐率 |
| 技术 | 使用特定工具模拟超常数据量、负载 |
| 开始标准 | 概要设计完成后 |
| 完成标准 | 所有性能测试用例都被执行并通过；所有发现的缺陷都被修正并回归测试过；功能要求符合标准；性能测试报告被权签人批准 |
| 测试重点和优先级 | 无 |
| 需要考虑的特殊事项 | 监测系统的各项性能指标，搭建苛刻的硬件和软件测试设备环境 |

## 3.2测试阶段性计划

|  |  |  |
| --- | --- | --- |
| 测试范围 | 所需课时（单位：课时） | 具体时间 |
| **前期准备** | **5** | **11.16-11.23** |
| 完善测试框架 | 2 | 11.19-11.20 |
| 美化测试报告 | 2 | 11.21-11.23 |
| 通过javamail实现发送邮件 | 1 | 11.16 |
| **单元测试** | **2** | **11.26-11.27** |
| 测试各单元模块功能 | 1 | 11.26 |
| 完善代码 | 0.5 | 11.27 |
| 生成测试报告 | 0.5 | 11.27 |
| **集成测试** | **3** | **11.28-12.3** |
| 测试各单元模块集成之后的功能 | 1.5 | 11.28-11.29 |
| 完善代码 | 1 | 11.29-11.3 |
| 生成测试报告 | 0.5 | 11.3 |
| **功能测试** | **5** | **11.4-11.11** |
| 编写测试用例 | 1 | 11.4 |
| 编写测试代码 | 2 | 11.5-11.7 |
| 完善代码 | 1.5 | 11.10-11.11 |
| 生成测试报告 | 0.5 | 11.11 |
| **性能测试** | **3** | **11.12-11.15** |
| 手工测试 | 2 | 11.12-11.13 |
| 生成测试报告 | 1 | 11.15 |

## 3.3测试停止及恢复条件

1. 遇到测试人员缺失等调配人力等问题时，可以停止测试，直到找到弥补空位的人员或缺失人员回归岗位；
2. 遇到网络因各种原因不稳定、甚至中断的网络问题时，可以中断，直至网络恢复或者找到可以代替的新网络；
3. 当测试过程遇到硬件适配问题，如硬件资源不足，测试所需要的硬件设施缺失问题时，可以停止测试，直至找到所需的硬件资源；
4. 出现外界不可抗故障时可以停止测试，如：停电、网络中断等；
5. 当测试出现功能性错误无法停止时，强制性退出；
6. 遇到不可继续执行的漏洞，比如无法登录，页面无法打开等问题时可以停止，问题解决时恢复测试

# 4.测试资源

## 4.1人力资源

|  |  |  |
| --- | --- | --- |
| 人员 | 角色 | 具体职责 |
| 史素佳 | 测试经理 | 制定并维护测试计划 |
| 沈梦琪 | 测试工程师 | 设计测试用例以及测试进程，搭建测试环境，完成单元测试，系统测试，集成测试，提交测试报告 |
| 卢丹琦 |
| 杨天莹 |
| 韦士杰 |
| 刘 江 |
| 岳哲伟 |

## 4.2测试环境：

服务器测试环境配置：

服务器：本地服务器

地址：http://localhost:8032/wordpress/

硬件环境：CPU：Intel core i5

内存：4G

硬盘：500G

软件环境：火狐浏览器56.0版本，UPUP本地服务器

操作系统：windows10

搭建LAMP平台（Windows + Apache + MySQL + PHP），将CMS系统（WordPress）发布可以通过客户端浏览器访问并测试

## 4.3测试工具

|  |  |
| --- | --- |
| 测试工具 | 用途 |
| 禅 道 | 记录测试用例的执行和bug返回 |
| eclipse | 编写测试框架以及测试用例 |
| Postman | 执行并验证测试用例 |

# ５.风险评估

## 5.1 人力方面

风险：（1）人力资源不足

（2）人员经验以及对软件的熟悉度不够

（3）人员调整导致研发周期延迟

风险规避方法：（1）保障稳定的人员安排，提前做好人员离职后备措施

（2）进行项目培训，确保工作人员对软件的熟悉度

（3）一旦发生周期延迟，应马上调整进度安排，确保工作的完成

## 5.2 时间方面

风险：测试时间不足，缩短导致某些测试计划无法执行解决：

风险规避方法：做好时间和任务规划，鼓励测试人员在规定时间之前完成测试计划。

## 5.3 资源方面

风险：硬件资源不合适

风险规避方法：事先分析测试所需硬件资源，保证测试工作顺利进行

## 5.4 部门合作方面

风险：开发人员、测试人员关于项目的执行引起纠纷

风险规避方法：团结内部，使大家共同协作来解决问题，实行每日15分钟立会

## 5.5功能的实现情况

风险：模块功能不健全，未能实现理想效果

风险规避方法：（1）前期进行严格的说明，并对进度安排严格监督

（2）进行每日立会，对软件的功能实现进行实时沟通

（3）保证代码编写质量以及Bug的及时修复