虚拟宠物医院学习系统 性能测试计划

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ 、、] 正式发布  [ ] 正在修改 | 文件标识： | 性能测试计划\_G01\_A\_v1.0 |
| 当前版本： | 1.0 |
| 作 者： | 卞炜奕 |
| 完成日期： | 2018-04-12 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 0.1 | 卞炜奕 | 崔宇璇、谢斌、  杨潇睿 | 2018/3/16 | 初稿 |
| 0.2 | 卞炜奕 | 崔宇璇、谢斌、  杨潇睿 | 2018/3/22 | 修改 |
| 0.3 | 卞炜奕 | 崔宇璇、谢斌、  杨潇睿 | 2018/3/29 | 新增混合场景 |
| 0.4 | 卞炜奕 | 崔宇璇、谢斌、  杨潇睿 | 2018/4/9 | 新增混合场景与独立场景 |

目录

[1 项目概要介绍 4](#_Toc509511835)

[1.1 项目简介 4](#_Toc509511836)

[1.2 项目成员 4](#_Toc509511837)

[1.3 测试范围 4](#_Toc509511838)

[1.3.1 范围内 4](#_Toc509511839)

[1.3.2 范围外 4](#_Toc509511840)

[1.4 前提假设 4](#_Toc509511841)

[1.5 测试目标 5](#_Toc509511842)

[2 性能测试策略 5](#_Toc509511843)

[2.1 性能测试模型 5](#_Toc509511844)

[2.2 性能测试场景 6](#_Toc509511845)

[2.3 重点测试策略 6](#_Toc509511846)

[2.3.1 重点测试原则 6](#_Toc509511847)

[2.3.2 重点测试功能操作 7](#_Toc509511848)

[3 测试案例设计 7](#_Toc509511849)

[3.1 生产压力分析 7](#_Toc509511850)

[3.2 场景通过标准 7](#_Toc509511851)

[3.3 测试场景设计 7](#_Toc509511852)

[3.3.1 独立场景 7](#_Toc509511853)

[具体的独立场景测试用例如下所示： 8](#_Toc509511854)

[3.3.2 混合场景 10](#_Toc509511855)

[3.3.3 峰值场景 11](#_Toc509511856)

[4 测试实施安排 12](#_Toc509511857)

[4.1 测试进度 12](#_Toc509511858)

[4.2 测试流程 12](#_Toc509511859)

[4.3 测试报告需求 12](#_Toc509511860)

[4.4 性能缺陷管理 13](#_Toc509511861)

[5 性能测试标准 13](#_Toc509511862)

[5.1 启动标准 13](#_Toc509511863)

[5.2 中止标准 13](#_Toc509511864)

[5.3 通过标准 13](#_Toc509511865)

[6 测试环境规划 13](#_Toc509511866)

[6.1 部署环境 13](#_Toc509511867)

[6.2 执行环境 14](#_Toc509511868)

[7 测试风险分析 14](#_Toc509511869)

[8 角色与职责 15](#_Toc509511870)

# 项目概要介绍

## 项目简介

宠物医院近年来在国内逐步兴起。缺乏符合资质的宠物医生是宠物医疗产业的短板。然而，由于过强的分散性和地域经济相关性使得在各地建造实体宠物医生教学和培训机构并不现实。因此，基于互联网的宠物医疗方案应用而生，催生了宠物医院的信息化建设和相关教学培训的大规模发展。

虚拟宠物医院学习系统是一个虚拟宠物医院教学软件，可以使得宠物工作者不去实体医院就能系统地学习各种宠物诊疗专业知识。该软件主要针对相关专业毕业实习医生，能够使得毕业实习生了解宠物医院结构、科室、进行病例学习等。本软件设置不同岗位角色并配备约200个左右的真实病例。使用者可以通过选择如前台、医助等不同角色进行在线学习及考核等。通过宠物医院虚拟学习系统，宠物工作者能够充分熟悉宠物医院的工作环境、岗位责任及工作流程等，积累临床经验，为成为一名合格的宠物医生和建设符合资质的宠物医院奠定基础。

## 项目成员

谢斌，杨潇睿，崔宇璇、卞炜奕

## 测试范围

### 范围内

虚拟宠物医院学习系统的前端业务，包括3D导览、角色职能学习、病例学习、测试、登录注册的业务功能。

### 范围外

虚拟宠物医院学习系统的后台业务，包括用户管理、病例管理、数据管理、测试管理的业务功能。

## 前提假设

被测系统完成功能测试，测试报告显示达到要求。

## 测试目标

1) 系统在单步操作响应时间方面，能满足用户当前及未来1到3年的发展需求；

2) 系统的批量作业运行稳定，处理能力能满足用户当前及未来1到3年的发展需求；

3) 在业务处理能力方面，能满足当前及未来1到3年的业务增长需求；

4) 发现并解决宕机，内存泄漏等严重问题，使系统具备良好的稳定性、健壮性；

5) 系统在资源使用方面比较合理，各项资源平均利用率在30%左右。

# 性能测试策略

与其它测试类型一样，性能测试周期分为：性能测试需求分析、性能测试设计、性能测试实现和性能测试报告四个存在迭代的阶段。其中，性能测试需求和性能测试设计由测试人员手工完成，而性能测试实现需借助性能测试工具达成，通过对性能测试报告的分析判定系统是否存在性能缺陷。

## 性能测试模型

性能测试模型定义了性能测试涉及的各个待测方面，是性能测试设计的指南。

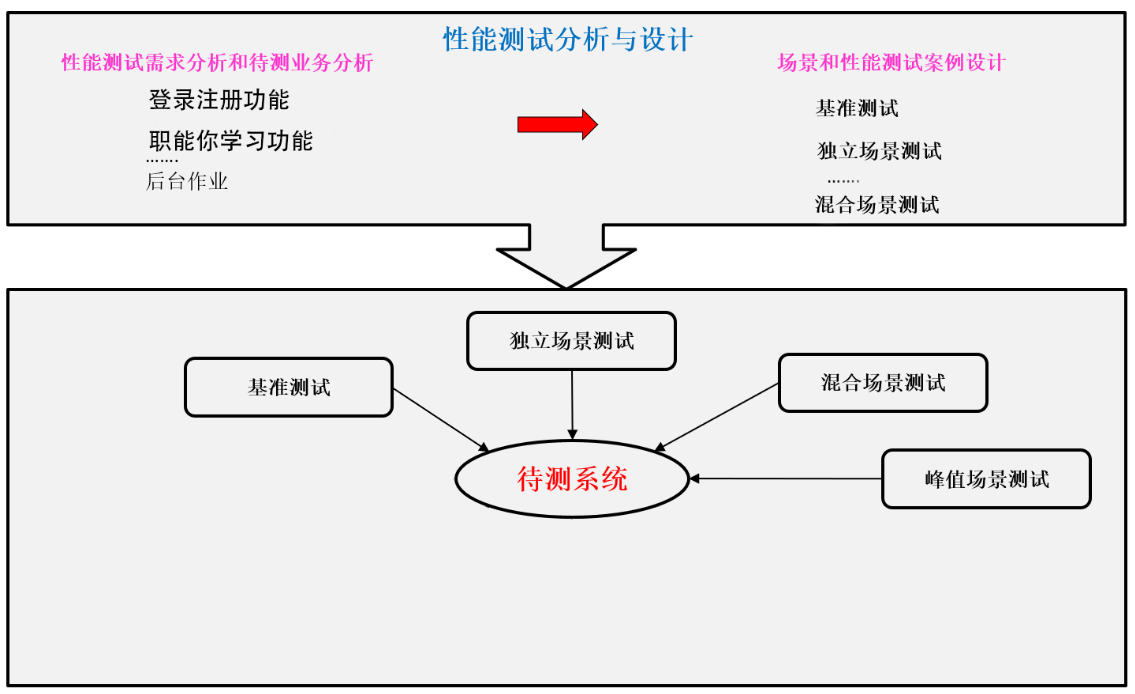


Figure 性能测试模型

根据Figure 1所示的性能测试模型，可以设计如Table 1所示的性能测试类型

Table 1性能测试类型

|  |  |  |  |
| --- | --- | --- | --- |
| **测试种类** | **测试方法** | **测试内容** | **场景类型** |
| 性能测试 | 以系统最大并发用户数的1-2倍作为上限对关键功能操作进行梯度加压测试，在压力时间内通过的功能操作量应接近峰值时段的功能操作量，甚至超过系统全天的功能操作量 | 核心业务，需求分析中发现的容易出现性能问题的业务 | 独立场景 |
| 压力测试 | 以系统预期最大并发用户数的1-2倍作为上限对常用/关键功能操作进行混合场景梯度加压测试，在压力时间内通过的功能操作量应接近峰值时段的功能操作量，甚至超过系统全天的功能操作量 | 存在一定关联关系的常用/核心业务 | 混合场景 |
| 峰值测试 | 按照系统峰值时预期最大并发用户数的2-3倍对峰值时段涉及的核心功能操作进行的混合场景梯度加压测试，在测试时间内通过的功能操作数量应接近或超过系统全天的功能操作量 | 生产上峰值场景中的核心业务 | 峰值场景 |

## 性能测试场景

根据项目实际，确定本次性能测试场景，例如可包含以下场景的性能测试：

独立场景

混合场景

峰值场景

## 重点测试策略

### 重点测试原则

为了全面评估系统性能，本次重点测试策略安排如下：

1. 对关键功能操作开展全面的独立场景测试；
2. 对关键功能操作开展全面的峰值测试；
3. 在集成性能测试方面，对功能操作量大、容易出现性能瓶颈的模块，重点进行测试并安排专人在系统发布前一直跟进开发组的需求变更以便进行及时的回归性能测试。
4. 在渠道性能测试方面，对渠道场景按照业务量及重要性来划分优先级，按照优先级先后来执行，保证系统上线后关键业务的稳定性。

### 重点测试功能操作

# 测试案例设计

## 生产压力分析

对于本系统的开发

## 场景通过标准

对于不同的测试场景，我们需要制定不同的通过标准才能顺利完成测试工作。具体要求如下表所示：

Table 场景通过标准

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景类型** | **单步操作响应时间** | **加压时间** | **操作账户数据** | **处理业务笔数** | **事务成功率** | **各后台利用率** | **并发/在线上限** | **在线思考时间设置** |
| 独立场景 | <1秒 | 30分钟 | >500 | >500 | 100% | <50% | 100 | 无 |
| 混合场景 | <1秒 | 1小时 | >500 | >1000 | 100% | <80% | 200 | 无 |
| 峰值场景 | <2秒 | 1小时 | >500 | >1000 | 100% | <80% | 200 | 无 |

## 测试场景设计

### 独立场景

对于关键场景，将分别对其采用渐进式加压的方式来进行独立场景测试。每个场景测试多组并发、并发数从1逐步增加到200。对于具体的场景，测试几组并发依据案例执行。案例需要按照业务量设计。对于使用频度较低的场景，可以采用独立/混合场景的方式进行测试，即在背景压力下以小并发的方式来进行长时间疲劳测试。

Table 独立场景设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 1 | <1秒 | N/A | 3分钟 | 直接加压 |
| 10 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 20 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100 次/秒 | 20分钟 | 1个用户/5秒 |
| 100 | <1秒 | >100 次/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <1秒 | >100 次/秒 | 10分钟 | 1个用户/1秒 |

### 具体的独立场景测试用例如下所示：

1、注册部分：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1 | 场景1:成功注册 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 邮箱名:example@qq.com  密码:12345687abc | 弹出注册成功框 |
| 2、 | 场景2:邮箱名或密码不符合规定格式 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 邮箱名:email@123.com  密码:1111 | 弹出格式异常提示框 |
| 3、 | 场景3:邮箱已被注册 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 邮箱名:example@qq.com  密码:12345687abc | 弹出邮箱名已被注册提示框 |
| 4、 | 场景4:用户未在正确输入验证码 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 邮箱名:example@qq.com  密码:12345687abc | 弹出验证码错误提示框 |

2、登录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:成功登录 | 1、进入登录页面  2、输入邮箱名  3、输入密码  4、点击“登录” | 邮箱名:example@qq.com  密码:12345687abc | 弹出登录成功框 |
| 2、 | 场景2:邮箱名或密码错误 | 1、进入登录页面  2、输入邮箱名  3、输入密码  4、点击“登录” | 邮箱名:email@123.com  密码:1111 | 弹出邮箱名或密码错误提示框 |
| 3、 | 场景3:用户未注册 | 1、进入登陆页面  2、用户发现自己未注册账号  3、点击“注册”进入注册页面进行注册  4、点击“登陆”  5、进入登陆页面  6、输入邮箱名  7、输入密码  8、点击“登陆” | 邮箱名:example@qq.com  密码:12345687abc | 弹出登录成功框 |

3、测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:完成测试 | 1、进入测试页面  2、选择测试分类  3、选择测试名称  4、完成答题  5、提交试卷 | 试题选项：abc等 | 弹出试卷提交成功框 |
| 2、 | 场景2:用户超时完成试卷 | 1、进入测试页面  2、选择测试分类  3、选择测试名称  4、选择答案  5、测试时间超时  系统自动提交试卷 | 试题选项：abc等 | 弹出试卷已经自动提交 |
| 3、 | 场景3：用户添加题目至搜藏夹 | 1、进入练习页面  2、选择要加入的题目  3、点击收藏题目按钮 | 暂无 | 弹出提示框提示 |
| 4、 | 场景3：用户将题目从搜藏夹中删除 | 1、进入练习页面  2、选择要取消的题目  3、点击取消收藏按钮 | 暂无 | 弹出提示框提示 |

4、病例学习

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:查看病例 | 1、进入病例学习页面  2、选择病种  3、选择病例点击查看 | 暂无 | 页面上显示所选的病例信息 |

### 混合场景

对于相关联的关键功能操作，对其采用渐进式加压的方式来进行混合场景测试，以测试这些关联功能操作是否存在性能问题。每个场景测试多组并发，并发数逐步增加，对于不同的场景，增加的数量与峰值有所不同。对于具体的场景，测试几组并发依据案例来执行（案例主要依据业务量来设计）。对于使用频度较低的场景，可以采用独立/混合场景的方式进行测试，即在背景压力下以小并发的方式来进行长时间疲劳测试。

具体场景内的操作流程包括：

1、用户注册—>用户登录—>用户退出

2、用户登录—>用户点击查看3D导览图—>用户点击图上科室查看具体信息—>用户退出

3、用户登录—>用户查看病例信息—>用户查看病种信息—>用户退出

4、用户登录—>用户选择医师为角色进行角色扮演—>用户退出

5、用户登录—>用户选择医助为角色进行角色扮演—>用户退出

6、用户登录—>用户选择前台为角色进行角色扮演—>用户退出

7、用户登录—>用户进行答题—>用户提交试卷—>用户退出

8、管理员登录—>管理员对病例信息做出修改—>管理员退出

9、管理员登录—>管理员对病种信息做出修改—>管理员退出

10、管理员登录—>管理员选择上传病例图片—>管理员查看病例图片

11、管理员登录—>管理员新增考试信息—>管理员退出

12、用户登录—>用户查看视频—>用户退出

13、用户登录—>用户查看病例图片—>用户退出

14、用户登录—>用户查看错题本—>用户退出

15、用户登录—>用户查看收藏夹—>用户退出

16、管理员登录—>管理员添加新的题目—>管理员退出

对于操作流程1，7，11，该流程在系统中出现率非常高，采用从20到250逐渐加压的方式。以得到该场景通过测试的成功率。而对于4，5，6等操作流程，用户使用频率较高，因此采用20到150的加压方式。最后对于操作8，9等，此类操作在系统中出现次数较少，所以采用20到100的加压方式。这样可以保证测试的顺利进行并且有效利用资源。

Table 混合场景设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 20 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100 /秒 | 20分钟 | 1个用户/5秒 |
| 100 | <2秒 | >100/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <2秒 | >100/秒 | 10分钟 | 1个用户/1秒 |

### 峰值场景

对于该系统，大批量的操作多数集中于上午8:00-12:00，期间考虑到多是安排考试以及用户进行测试练习以及病例学习，期间会有大批量的登录，考试操作以及查看病例信息。因此，峰值最长持续时间3-4小时。考虑到系统的扩展性及实际情况，峰值测试共测试三组：第一组测试正常峰值压力的处理能力；第二组测试2倍、4倍峰值压力下的系统处理能力，以满足未来的业务发展需求，测试时各组中的功能操作按照业务比例同时加载。

第一组：先用生成峰值时压力的70%加压1小时作为基准压力，之后采用当前生产上的实际峰值压力并发进行加压。

Table 峰值场景设计1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 生产压力的70% | <1秒 | >100/秒 | 1小时 | 1个用户/5秒 |
| 生产压力 | <1秒 | >100/秒 | 4小时 | 1个用户/5秒 |

第二、三组：分别采用生产压力的2到4倍来进行加压，第二组达标后再进行后面的测试。

Table 峰值测试设计2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 生产压力的2倍 | <1秒 | >100/秒 | 5小时 | 1个用户/5秒 |
| 生产压力的4倍 | <1秒 | >100/秒 | 5小时 | 1个用户/5秒 |

对于性能较好的系统，可以直接加压到最大并发数，持续加压5小时来测试系统的峰值处理能力。

# 测试实施安排

## 测试进度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **任务** | **开始日期** | **结束日期** | **参与人员** |
| 1 | 熟悉被测试系统，确定典型事务 | 2018.03.15 | 2018.03.22 | 测试人员  开发人员  项目经理 |
| 2 | 搭建测试环境，  录制典型事务的脚本，增强脚本 | 2018.03.23 | 2018.04.01 | 测试人员 |
| 3 | 执行测试并收集相关数据 | 2018.04.02 | 2018.04.16 | 测试人员 |
| 4 | 数据分析，编写测试报告 | 2018.04.17 | 2018.04.25 | 测试人员 |

## 测试流程

流程分为需求分析、设计、实现和测试报告四个阶段。

性能测试需求分析与设计：

熟悉被测试系统，定义性能测试要求，例如并发用户的数量、典型业务流程和所需响应时间。

性能测试实现：

创建虚拟用户脚本：将最终用户活动捕获到自动脚本中。

定义场景：使用 LoadRunner Controller 设置负载测试环境。

运行场景：通过 LoadRunner Controller 驱动、管理和监控负载测试。

生成性能测试报告：

使用 LoadRunner Analysis 创建图和报告并评估性能。

## 测试报告需求

Table 测试报告需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **报告类型** | **响应时间要求** | **报告者** | **接受者** | **报告内容** |
| 邮件 | 每日 | 各渠道测试组负责人 | 测试中心，项目经理 | 每日结果总结报告 |
| 邮件 | 每日 | 各渠道测试组负责人 | 测试中心，项目经理 | 每日测试结果记录 |
| 邮件 | 每日 | 性能测试负责人 | 测试中心，项目经理 | 周报 |
| 文档 | 各测试阶段结束 | 各渠道测试组负责人 | 测试中心，项目经理 | 性能测试报告 |

## 性能缺陷管理

测试过程采用Quality Center进行缺陷管理

# 性能测试标准

## 启动标准

1. 测试环境满足计划需求
2. 基准参数配置完成校验
3. 关键功能操作通过冒烟测试

## 中止标准

1. 测试环境或关键系统不可用
2. 测试环境距生产标准差距太大
3. 缺陷周转周期不符合规定的时间
4. 出现宕机、不响应等严重的性能问题
5. 系统的功能操作成功率低于95%

## 通过标准

系统上线至少满足下面标准：

1. 系统无宕机、不响应类的严重性能问题
2. 系统响应时间80%达到系统的期望值
3. 系统的业务吞吐量达到预期目标，即当前生产需求的3倍以上
4. 柜台类系统要求通过7\*12小时以上的疲劳强度测试
5. 电子渠道类系统要求通过7\*24小时以上的疲劳强度测试

# 测试环境规划

## 部署环境

测试环境要求尽量和真实环境相同，至少要求服务器配置和网络带宽和拓扑结构应该相似。主要内容：服务器数量和配置，操作系统和数据库版本，软硬件部署等。

|  |  |  |  |
| --- | --- | --- | --- |
| **软件环境（相关软件、操作系统等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| MAC OSX | 10.11 |  |  |
| mysql | 5.7 |  |  |
| Chrome/FireFox/IE浏览器 | 最新版本 |  |  |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 |  |  |  |
| 数据库服务器 |  |  |  |
| 前台客户机 |  |  |  |

## 执行环境

|  |  |  |  |
| --- | --- | --- | --- |
| **软件环境（相关软件、操作系统等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| MAC OSX | 10.11 |  |  |
| mysql | 5.7 |  |  |
| Chrome/FireFox/IE浏览器 | 最新版本 |  |  |
| LoadRunner | 最新版本 |  |  |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 |  |  |  |
| 数据库服务器 |  |  |  |
| 前台客户机 |  |  |  |

# 测试风险分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **风险因素** | **可能结果** | **可能发生时间** | **风险**  **级别** | **应对措施** |
| 工具缺陷 | 测试工具和监控工具无法全部支持信贷业务系统的测试和监控 | 随时 | 中 | 评估被测系统，分析所有需求。  通过其它工具实现对需求的支持程度。 |
| 测试数据的准备备份及恢复无法正常完成 | 测试过程中数据用尽或不满足测试需求，将导致测试无法实施。 | 测试执行时 | 高 | 运维方配合完成数据的准备、备份和恢复 |
| 测试环境有其他用户连接进行操作，服务器产生性能缺陷 | * 1. 测试方获得最大负载压力与实际最大负载有差距   2. 服务器出现性能缺陷的现象，运维方定位性能缺陷模块并非真正性能缺陷的模块 | 测试执行时 | 高 | 测试方进行负载测试时，保证测试环境无其他连接和用户操作 |
| 测试服务器访问状态不稳定 | 测试准备和测试执行中断，测试计划时间延后 | 随时 | 高 | 保证测试期间测试环境访问畅通 |

# 角色与职责

|  |  |  |  |
| --- | --- | --- | --- |
| **角色** | **姓名** | **具体职责** | **技能要求** |
| 性能测试经理 | 谢斌 | 性能测试项目管理和协调  性能测试需求分析  测试计划和测试报告编写 | 1. 组织协调能力 2. 熟悉性能测试架构 3. 熟悉应用架构 |
| 性能测试设计人员 | 卞炜奕 | 编制测试案例，开发测试脚本，设计测试场景，执行性能测试 | 1. 熟悉性能测试体系 2. 熟悉性能测试方法和工具 3. 脚本编写和调试能力 |
| 测试环境和测试数据准备人员 | 崔宇璇 | 准备性能测试环境  准备基础数据和测试数据 | 1. LoadRunner部署能力 2. 熟悉核心系统、数据库 |
| 性能调优人员 | 杨潇睿 | 核心主机系统优化 | 1. 熟悉数据库优化 2. 熟悉应用架构 |