****

**省人力资源市场数据采集系统**

**项目管理计划**

**姓 名：** 郑乐祺

**学 号：** 1120202073

**学 院：** 计算机学院

**专 业：** 软件工程

**指导老师：** 闫波

**目录**

[1. 引言 9](#_Toc8853)

[1.1. 编写目的 9](#_Toc165)

[1.2. 预期读者 9](#_Toc6173)

[1.3. 参考资料 9](#_Toc5677)

[2. 需求变更记录 10](#_Toc579)

[2.1. 第一次需求变更 10](#_Toc12323)

[2.2. 第二次需求变更 12](#_Toc8523)

[3. 范围管理计划 15](#_Toc10729)

[3.1. 范围管理策略 15](#_Toc22139)

[3.2. 角色和职责 15](#_Toc11636)

[3.3. 范围定义 15](#_Toc17071)

[3.4. WBS图 16](#_Toc25598)

[3.4.1. 项目总体安排WBS图 16](#_Toc3774)

[3.4.2. 详细设计WBS图 18](#_Toc3429)

[3.4.3. 前端子模块WBS图 18](#_Toc21485)

[3.4.4. 后端子模块WBS图 19](#_Toc15180)

[3.4.5. 测试任务WBS图 19](#_Toc3322)

[3.5. 验收标准 19](#_Toc5755)

[3.6. 范围基准 20](#_Toc31883)

[3.7. 范围控制 20](#_Toc2067)

[3.8. 范围验证 20](#_Toc7169)

[3.9. 变更管理 20](#_Toc11666)

[4. 进度管理计划 20](#_Toc32259)

[4.1. 进度管理概述 20](#_Toc30134)

[4.2. 各阶段安排表： 21](#_Toc10810)

[4.2.1. 需求分析安排 21](#_Toc30148)

[4.2.2. 总体设计安排 21](#_Toc25622)

[4.2.3. 详细设计安排 22](#_Toc16831)

[4.2.4. 代码实现安排 24](#_Toc7318)

[4.2.5. 软件测试安排 24](#_Toc10124)

[4.2.6. 文档审查安排 26](#_Toc3940)

[4.2.7. 项目交付安排 26](#_Toc27682)

[4.3. 项目里程碑 27](#_Toc19910)

[4.4. 各阶段任务 27](#_Toc25387)

[4.4.1. 需求分析任务 27](#_Toc20026)

[4.4.2. 总体设计任务 28](#_Toc15558)

[4.4.3. 详细设计任务 28](#_Toc3500)

[4.4.4. 代码实现任务 29](#_Toc587)

[4.4.5. 软件测试任务 30](#_Toc30635)

[4.4.6. 文档审查任务 30](#_Toc16961)

[4.4.7. 项目交付任务 30](#_Toc13583)

[5. 成本管理计划 31](#_Toc11491)

[5.1. 软件项目规模 31](#_Toc15586)

[5.2. 成本 31](#_Toc11748)

[5.2.1. 人力资源成本 32](#_Toc8594)

[5.2.2. 软硬件成本 33](#_Toc13658)

[5.3. 成本计算 35](#_Toc9398)

[5.3.1. 人力资源成本计算 35](#_Toc12535)

[5.3.2. 总成本计算 36](#_Toc11751)

[6. 质量管理计划 37](#_Toc32751)

[6.1. 质量目标和要求 37](#_Toc25983)

[6.1.1. 质量目标 37](#_Toc16197)

[6.1.2. 完成目标所需方案 38](#_Toc31296)

[6.2. 质量管理组织架构 38](#_Toc31064)

[6.3. 质量确保对象 39](#_Toc4392)

[6.3.1. 文档对象 39](#_Toc23260)

[6.3.1.1. 文档质量保证对象 39](#_Toc6406)

[6.3.1.2. 文档质量衡量标准 39](#_Toc24745)

[6.3.2. 代码对象 41](#_Toc18310)

[6.3.2.1. 代码质量保证对象 41](#_Toc19726)

[6.3.2.2. 代码质量衡量标准 41](#_Toc29646)

[6.4. 质量计划 43](#_Toc26744)

[6.4.1. 需求分析阶段的质量控制措施 43](#_Toc25320)

[6.4.2. 设计阶段的质量控制措施 44](#_Toc28859)

[6.4.3. 编码和测试阶段的质量控制措施 44](#_Toc31881)

[6.4.4. 上线和运维阶段的质量控制措施 44](#_Toc29503)

[6.5. 质量检查和审核 45](#_Toc21232)

[7. 人力管理计划 45](#_Toc29139)

[7.1. 项目团队成员 45](#_Toc32506)

[7.2. RACI模型 47](#_Toc28417)

[7.3. 人力资源管理措施 48](#_Toc6561)

[8. 沟通管理计划 48](#_Toc9750)

[8.1. 项目沟通 48](#_Toc9838)

[8.2. 会议沟通 48](#_Toc5169)

[9. 风险管理计划 55](#_Toc7864)

[9.1. 风险条目与评估 56](#_Toc25374)

[9.1.1. 技术风险 56](#_Toc2983)

[9.1.1.1. 数据安全风险 56](#_Toc3023)

[9.1.1.2. 技术实现风险 56](#_Toc19776)

[9.1.1.3. 数据读写性能风险 56](#_Toc30112)

[9.1.1.4. 系统稳定性风险 56](#_Toc24345)

[9.1.1.5. 安全性风险 57](#_Toc16596)

[9.1.1.6. 系统架构风险 57](#_Toc12129)

[9.1.1.7. 据备份和恢复风险 57](#_Toc7663)

[9.1.2. 业务风险 57](#_Toc857)

[9.1.2.1. 用户需求变更风险 57](#_Toc24784)

[9.1.2.2. 数据准确性风险 58](#_Toc31865)

[9.1.2.3. 数据完整性风险 58](#_Toc13761)

[9.1.2.4. 业务流程风险 58](#_Toc11218)

[9.1.2.5. 复杂业务流程风险 58](#_Toc20434)

[9.1.3. 法律风险 58](#_Toc19822)

[9.1.3.1. 数据合规风险 58](#_Toc25398)

[9.1.3.2. 合同法律风险 59](#_Toc21434)

[9.1.3.3. 信息安全风险 59](#_Toc5000)

[9.1.3.4. 商标法律风险 59](#_Toc27458)

[9.1.4. 人力资源风险 59](#_Toc32022)

[9.1.4.1. 人力资源不足风险 59](#_Toc8516)

[9.1.4.2. 人力资源技术欠缺风险 59](#_Toc31143)

[9.1.4.3. 项目管理经验缺失风险 60](#_Toc27623)

[9.1.5. 软件测试不足风险 60](#_Toc1114)

[9.1.6. 项目延期风险 60](#_Toc2327)

[9.1.7. 风险评估结果表 60](#_Toc32141)

[9.2. 风险定性分析 62](#_Toc4437)

[9.3. 应对措施 63](#_Toc22255)

[9.3.1. 风险应对总体策略 63](#_Toc17920)

[9.3.2. 技术风险应对策略 64](#_Toc12582)

[9.3.3. 业务风险应对策略 64](#_Toc7899)

[9.3.4. 法律风险应对策略 64](#_Toc32694)

[9.3.5. 人力资源风险应对策略 65](#_Toc30947)

[9.3.6. 软件测试风险应对策略 65](#_Toc11532)

[9.3.7. 时间风险应对策略 66](#_Toc6322)

[9.4. 风险监控和报告 66](#_Toc18383)

[10. 合同管理计划 66](#_Toc1272)

[10.1. 合同管理团队 66](#_Toc19442)

[10.2. 合同类型 67](#_Toc8664)

[10.3. 合同审查和批准 67](#_Toc27927)

[10.4. 合同执行 67](#_Toc31795)

[10.5. 合同变更管理 67](#_Toc13634)

[10.6. 合同支付 67](#_Toc27675)

[10.7. 合同终止 68](#_Toc8687)

[10.8. 记录和文档管理 68](#_Toc12292)

[11. 配置管理计划 68](#_Toc31493)

[11.1. 软件配置管理 68](#_Toc28276)

[11.1.1. 软件配置管理组织 68](#_Toc14912)

[11.1.2. 软件过程生命周期 70](#_Toc14913)

[11.2. 软件配置管理活动 70](#_Toc20228)

[11.2.1. 配置项标识 71](#_Toc21911)

[11.2.2. 配置项计划 71](#_Toc10684)

[11.2.3. 配置基线 73](#_Toc4705)

[11.2.4. 配置库结构 74](#_Toc4908)

[11.3. 软件配置审核 74](#_Toc25288)

[11.4. 软件配置控制 75](#_Toc14182)

[11.4.1. 版本管理 76](#_Toc746)

[11.4.2. 项目管理 77](#_Toc18743)

# 引言

## 编写目的

本项目管理计划的编写目的是为了确保云南省企业就业失业数据采集系统项目能够按计划、高质量地完成，满足云南省管理部门对企业就业失业数据收集和分析的需求。本计划将提供明确的项目目标、范围、时间、成本、质量、沟通、风险和相关方管理等方面的计划，以便项目团队能够有效地组织和管理项目工作，最大程度地实现项目成功交付。

## 预期读者

本项目管理计划的预期读者包括项目团队成员、项目经理、云南省管理部门的相关人员以及其他对项目管理计划感兴趣的利益相关者。项目团队成员将根据该计划指导自己的工作，项目经理将负责监督和管理整个项目的执行，云南省管理部门的相关人员将了解项目的管理方法和计划，利益相关者将了解项目的执行和控制过程，以保证项目的成功交付和利益的实现。

## 参考资料

在编写本项目管理计划时，参考了以下资料和标准：

**云南省企业就业失业数据采集系统的需求文档：**

该文档提供了项目的业务需求、功能要求和数据项定义，为项目的范围和目标的明确提供了依据。

**项目管理知识体系（PMBOK）指南：**

国际上广泛认可的项目管理标准和最佳实践，提供了项目管理的框架、方法和工具，用于指导项目的规划、执行和控制。

**组织内部项目管理方法论和模板：**

包括项目计划、风险管理、沟通管理等方面的内部指南和工具，为项目管理计划的编写提供了参考和支持。

# 需求变更记录

## 第一次需求变更

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 客户 | |  | | 项目名称 | 省人力资源市场数据采集系统 |
| 客户版本 | | BLG-YUNNANSYS-20230531-SC-v1.0 | | 报告人 | 郑乐祺 |
| 日期 | | 2023.06.02 | | 表单编号 | BLG-YUNNANSYS-01-change-0602 |
| 变更提出点：☑客户需求 □开发测试发现 □工程实施发现 □其他问题 | | | | | |
| 变更内容：  **报表次数变化：**  三到五月份每月报表四次  九到十月份每月报表四次  其余月份每月报表一次 | | | | | |
| 更改后希望达到目标：  不同月份对应不同报表次数 | | | | | |
| 建议解决方案：  考虑到春招和秋招，对报表系统进行更改。 | | | | | |
| □ 基线版本：\_\_\_\_\_\_\_\_\_\_\_\_ | | | | 变更原因：□缺陷 □功能增强 ☑新需求 | |
| 涉及模块 | |  | | | |
| 预计工作量 | |  | | 是否变更 | ☑是 □否 |
| 优先级 | | □特急 ☑急 □一般 □缓 | | 完成时限 |  |
| 确认意见 | | 予以更改  签字：郑乐祺 | | | |
| 变更影响范围 | | | | 配置提取路径 | |
| 申请配置项目 | 源代码 |  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| 文档 | 范围管理计划 | |  | |
| 进度管理计划 | |  | |
| 成本管理计划 | |  | |
| 变更实施人 | | 郑乐祺 | 申请日期 | | 2023年06月02日 |

## 第二次需求变更

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 客户 | |  | | 项目名称 | 省人力资源市场数据采集系统 |
| 客户版本 | | BLG-YUNNANSYS-20230606-SC-v3.0 | | 报告人 | 郑乐祺 |
| 日期 | | 2023.6.7 | | 表单编号 | BLG-YUNNANSYS-02-change-0607 |
| 变更提出点：☑客户需求 □开发测试发现 □工程实施发现 □其他问题 | | | | | |
| 变更内容：  部分省市的申报流程改变 | | | | | |
| 更改后希望达到目标：  从 监测点->(小)市->省  变成 监测点->区->市(大市)->省 | | | | | |
| 建议解决方案：  对部门省市的申报流程进行细化，申报流程多增加一个申报到区的环节。  收集需要添加申报流程的省市名单，对此开辟新的申报流程，更改其申报权限。 | | | | | |
| □ 基线版本：\_\_\_\_\_\_\_\_\_\_\_\_ | | | | 变更原因：□缺陷 □功能增强 ☑新需求 | |
| 涉及模块 | |  | | | |
| 预计工作量 | |  | | 是否变更 | ☑是 □否 |
| 优先级 | | ☑特急 □急 □一般 □缓 | | 完成时限 |  |
| 确认意见 | | 予以更改  签字：郑乐祺 | | | |
| 变更影响范围 | | | | 配置提取路径 | |
| 申请配置项目 | 源代码 |  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
| 文档 | 范围管理计划 | |  | |
| 进度管理计划 | |  | |
| 成本管理计划 | |  | |
| 变更实施人 | | 郑乐祺 | 申请日期 | | 2023年6月7日 |

# 范围管理计划

## 范围管理策略

范围管理将遵循以下策略：

所有的需求变更必须经过正式的变更控制过程。

项目团队将与关键利益相关者一起定义并确认项目范围。

项目范围将在项目生命周期中进行定期审查和更新。

## 角色和职责

**项目经理：**

负责管理项目范围，包括定义、确认和控制范围。

**业务分析师：**

负责收集和分析需求，以及维护需求文档。

**技术团队：**

负责实现定义的需求。

## 范围定义

项目范围包括以下功能：

企业用户：

企业信息管理、备案上报、数据填报、数据查询。

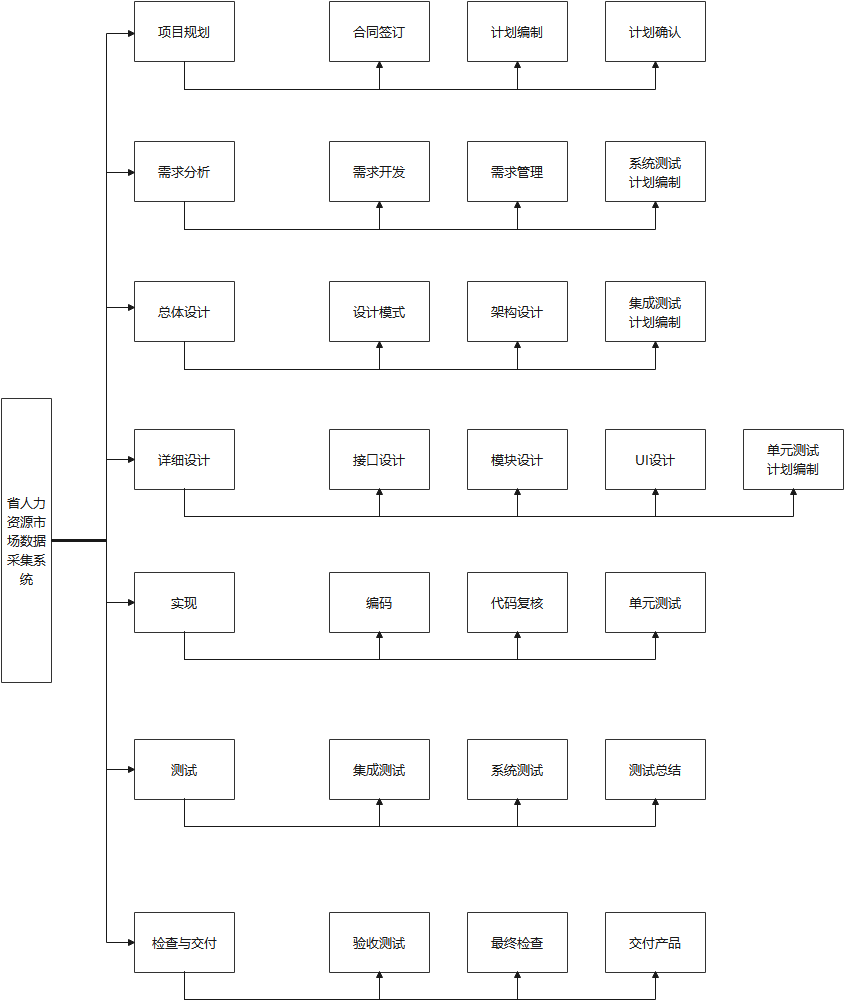
省级用户：

企业备案管理、报表管理、数据修改、数据删除、数据退回、数据汇总、数据导出、数据查询、多维分析、图表分析、发布通知、系统管理。

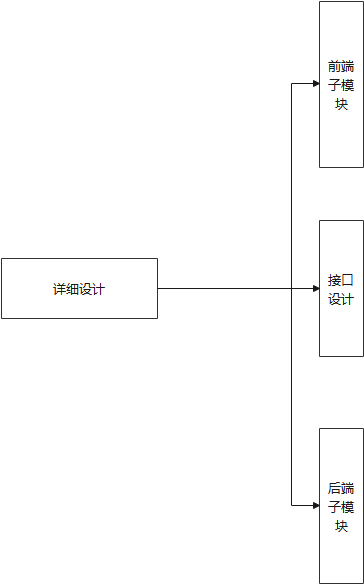
## WBS图

项目的工作分解结构将根据功能需求进行定义，每个功能将进一步分解为一系列的任务，包括需求分析、设计、开发、测试和部署。

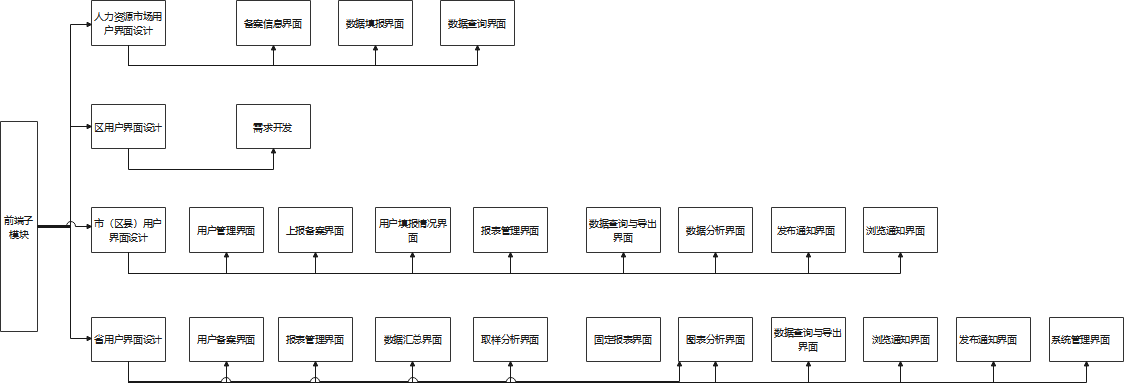
### 项目总体安排WBS图



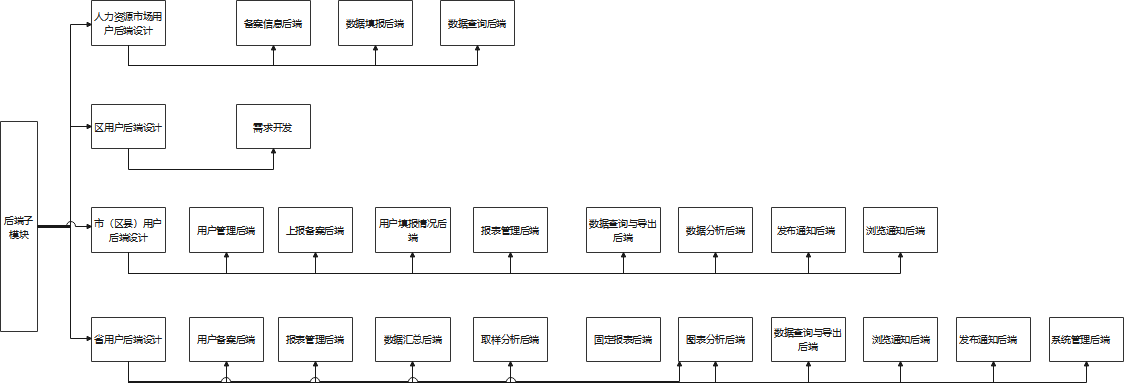
### 详细设计WBS图



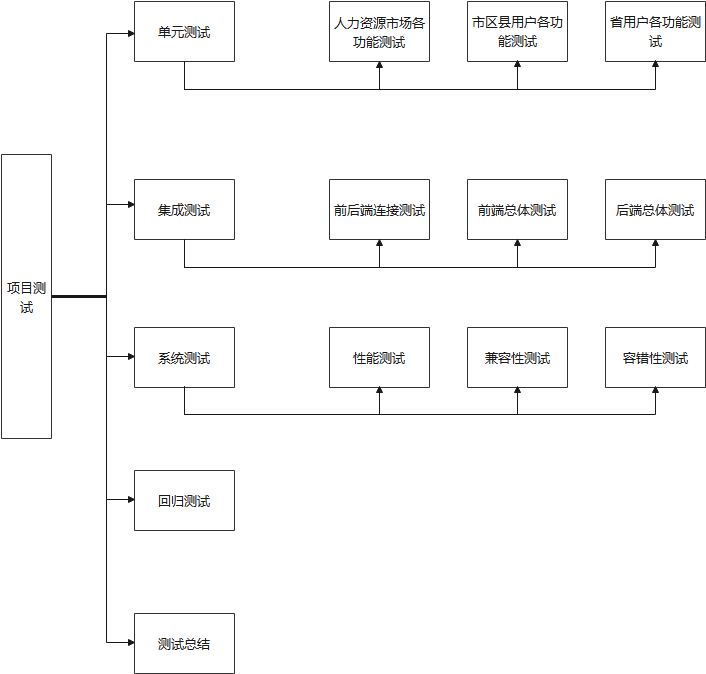
### 前端子模块WBS图



### 后端子模块WBS图



### 测试任务WBS图



## 验收标准

验收标准将基于需求文档中定义的功能需求。每个功能都必须通过单元测试、集成测试和用户验收测试。

## 范围基准

项目范围基准将在项目计划批准后建立，包括批准的项目范围说明、WBS和WBS词典。

## 范围控制

项目团队将定期审查项目进度，以确保工作按照项目范围进行。任何超出范围的工作都必须经过变更控制过程。

## 范围验证

项目团队将与关键利益相关者一起进行范围验证，以确保已完成的工作满足项目需求。

## 变更管理

所有的范围变更请求必须提交给项目经理，项目经理将评估变更请求的影响，并决定是否批准。所有的变更请求和决定都必须进行记录。

# 进度管理计划

## 进度管理概述

省人力资源市场数据采集系统项目计划在两个月内完成，其中一个月用于开发过程，另一个月用于测试过程和项目交付。下面是项目的重要时间节点和各阶段任务。

## 各阶段安排表：

### 需求分析安排

**开始时间：**

2023年5月01日

**完成时间：**

2023年5月04日

**负责人：**

项目经理

**工作内容：**

明确项目的需求

**阶段产出：**

系统软件规格说明书

**备注：**

预期四天

### 总体设计安排

**开始时间：**

2023年5月05日

**完成时间：**

2023年5月06日

**负责人：**

技术总监

**工作内容：**

根据需求，完成项目总体设计，确定设计模式、架构等

**阶段产出：**

总体设计说明文档

**备注：**

预期二天

### 详细设计安排

**前端模块设计：**

**开始时间：**

2023年5月07日

**完成时间：**

2023年5月10日

**负责人：**

前端架构师

**工作内容：**

设计前端各界面

**阶段产出：**

前端设计文档

**接口设计：**

**开始时间：**

2023年5月10日

**完成时间：**

2023年5月11日

**负责人：**

技术总监

**工作内容：**

设计接口，连接前后端

**阶段产出：**

接口设计文档

**后端模块设计：**

**开始时间：**

2023年5月07日

**完成时间：**

2023年5月10日

**负责人：**

后端架构师

**工作内容：**

设计后端各功能

**阶段产出：**

后端设计文档

**备注：**

详细设计总预期不超过五天

### 代码实现安排

**开始时间：**

2023年5月12日

**完成时间：**

2023年5月31日

**负责人：**

技术总监

**工作内容：**

实现各模块代码

**阶段产出：**

初代软件版本

**备注：**

代码实现预计二十天

### 软件测试安排

**单元测试：**

**开始时间：**

2023年6月01日

**完成时间：**

2023年6月10日

**负责人：**

测试经理

**工作内容：**

测试各单元功能，制作测试报告、缺陷报告、质量报告

**集成测试：**

**开始时间：**

2023年6月11日

**完成时间：**

2023年6月15日

**负责人：**

测试经理

**工作内容：**

测试系统集成情况

**系统测试：**

**开始时间：**

2023年6月16日

**完成时间：**

2023年6月20日

**负责人：**

测试经理

**工作内容：**

测试系统性能、非功能性需求

### 文档审查安排

**开始时间：**

2023年6月21日

**完成时间：**

2023年6月22日

**负责人：**

项目经理

**工作内容：**

审查各文档的完整性

**阶段产出：**

整套项目文档

**备注：**

预期二天

### 项目交付安排

**开始时间：**

2023年6月23日

**完成时间：**

2023年6月24日

**负责人：**

项目经理

**工作内容：**

项目最终检验，交付项目

**阶段产出：**

项目交付

**备注：**

预期二天

## 项目里程碑

需求分析：第四天

总体设计：第六天

详细设计：第十一天

代码实现：第三十一天

软件测试：第五十一天

文档审查：第五十三天

项目交付：第五十五天

## 各阶段任务

### 需求分析任务

**任务目标：**

明确项目需求

**注意事项：**

在前期，需要与干系人进行全面的沟通，明确项目需求。在后期，与干系人保持联系，获取需求的变更。

**完成标准：**

制作需求分析规格书，并取得干系人的确认。

### 总体设计任务

**任务目标：**

设计出系统的框架和采取的设计模式

**注意事项：**

需要明确前后端应使用哪种框架，明确指出规划出前后端采用的设计模式，并绘制出系统测试计划。

**完成标准：**

制作总体设计方案书和系统测试计划书，获得项目经理的确认。

### 详细设计任务

**前端模块设计：**

**任务目标：**

为每一个功能点在前后端进行详细设计，并为前后端接口进行设计

**注意事项：**

前端应设计出各界面UI，需包含接口的位置使用。

**完成标准：**

制作前端UI设计书，后端功能框架设计书，接口设计文档。

**接口设计：**

**任务目标：**

设计接口，连接前后端

**负责人：**

技术总监

**完成标准：**

接口设计文档

**后端模块设计：**

**任务目标：**

设计后端各功能

**负责人：**

后端架构师

**完成标准：**

后端设计文档

### 代码实现任务

**任务目标：**

通过接口，实现各功能代码和前端UI代码

**注意事项：**

前端完成界面UI代码编写，获取接口的数据对UI界面进行填充。后端完成各接口方法的编写，实现各功能代码。

**完成标准：**

可运行软件

### 软件测试任务

**任务目标：**

对可运行软件进行测试

**注意事项：**

对软件按测试计划进行单元测试，回归测试，系统测试，性能测试等。

**完成标准：**

软件无较大的缺陷，制作缺陷报告、测试报告、质量报告。

### 文档审查任务

**任务目标：**

对之前所有的文档进行审查

**注意事项：**

确保所有文档均符合标准，并确保所有文档的作用均体现在了项目之中。防止文档缺陷导致的项目缺陷。

**完成标准：**

整套完备的项目文档

### 项目交付任务

**任务目标：**

提交项目

**注意事项：**

确保项目符合合同的情况下，提交项目软件。

**完成标准：**

甲方允许接收项目。

# 成本管理计划

## 软件项目规模

|  |  |  |  |
| --- | --- | --- | --- |
| 项目重要任务 | | 工时（人天） | 总计（人天） |
| 需求分析 | | 4 | 55 |
| 总体设计 | | 2 |
| 详细  设计 | 前端模块设计 | 4 |
| 接口设计 | 2 |
| 后端模块设计 | 4(与前端并行) |
| 代码实现 | | 20 |
| 软件测试 | 单元测试 | 10 |
| 集成测试 | 5 |
| 系统测试 | 5 |
| 文档审查 | | 2 |
| 项目交付 | | 2 |

## 成本

### 人力资源成本

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 成员姓名 | 角色 | 类型 | 标准费率 | 加班费率 | 成本累算 | 基准日历 |
| 郑乐祺 | 项目经理 | 工时 | ￥150.00 /h | ￥180.00 /h | 按比例 | 标准 |
| 李四 | 业务分析员 | 工时 | ￥130.00 /h | ￥150.00 /h | 按比例 | 标准 |
| 王五 | 架构师 | 工时 | ￥130.00 /h | ￥150.00 /h | 按比例 | 标准 |
| 赵六 | 前端开发工程师 | 工时 | ￥140.00 /h | ￥160.00 /h | 按比例 | 标准 |
| 孙七 | 后端开发工程师 | 工时 | ￥140.00 /h | ￥160.00 /h | 按比例 | 标准 |
| 周八 | 数据库管理员 | 工时 | ￥140.00 /h | ￥160.00 /h | 按比例 | 标准 |
| 吴九 | 测试工程师 | 工时 | ￥110.00 /h | ￥130.00 /h | 按比例 | 标准 |
| 郑十 | UI/UX设计师 | 工时 | ￥110.00 /h | ￥130.00 /h | 按比例 | 标准 |
| 钱十一 | 技术支持 | 工时 | ￥110.00 /h | ￥130.00 /h | 按比例 | 标准 |
| 刘十二 | 文档工程师 | 工时 | ￥110.00 /h | ￥130.00 /h | 按比例 | 标准 |

### 软硬件成本

**服务器：**

规格和型号：R730

品牌：戴尔

单价：35000

数量：4

合价：140000

**电脑：**

规格和型号：组装机

品牌：无

单价：4000

数量：10

合价：40000

**电脑桌：**

规格和型号：人造板

品牌：曲臣

单价：700

数量：10

合价：7000

**电脑凳：**

规格和型号：不锈钢

品牌：骏帮

单价：200

数量：10

合价：2000

**插座：**

规格和型号：六眼

品牌：公牛

单价：40

数量：10

合价：400

**交换机：**

规格和型号：10Mbps-100Mbps

品牌：H3C

单价：1000

数量：2

合价：2000

**网线、电源线、副线、底线、水晶头：**

品牌：安普

单价：1000

数量：3

合价：3000

**装修人工费及其他费用：**

合价：30000

**软硬件成本总价 =** ￥229400

## 成本计算

### 人力资源成本计算

首先我们需要对人力资源进行分配：

**需求分析:**

业务分析员和项目经理

**总体设计:**

架构师和项目经理

**详细设计:**

架构师、前端开发工程师、后端开发工程师、数据库管理员和项目经理

**代码实现:**

前端开发工程师、后端开发工程师、数据库管理员和项目经理

**软件测试:**

测试工程师和项目经理

**文档审查:**

文档工程师和项目经理

**项目交付:**

技术支持、UI/UX设计师和项目经理

**考虑到周末加班，计算出总成本为：**

项目经理: ￥76,800

业务分析员: ￥4,160

架构师: ￥16,640

前端开发工程师: ￥38,080

后端开发工程师: ￥38,080

数据库管理员: ￥38,080

测试工程师: ￥17,600

文档工程师: ￥1,760

技术支持: ￥1,760 UI/UX

设计师: ￥1,760

**人力资源总成本 = ￥234,720**

### 总成本计算

直接成本 = 人力资源成本 + 软硬件成本 = ￥234,720 +￥229400 = ￥464,120

间接成本 = 直接成本 \* 25% = ￥467,240 \* 0.25 = ￥116,030

**总估算成本 = 直接成本 + 间接成本 = ￥467,240 + ￥116,810 = ￥580,150**

**项目利润 = 总估算成本 \* 25% = ￥584,050 \* 0.25 = ￥145,037.50**

项目利润包括利润、税费、风险基金：

利润 = 项目利润 \* 10% = ￥145,037.50 \* 0.1 = ￥14,503.75

税费 = 项目利润 \* 5% = ￥145,037.50 \* 0.05 = ￥7,251.88

风险基金 = 项目利润 \* 10% = ￥145,037.50 \* 0.1 = ￥14,503.75

**项目总造价 = 总估算成本 \* 125% = ￥580,150 \* 1.25 = ￥725,187.50**

# 质量管理计划

本质量管理计划的编制目的是为云南省人力资源市场数据采集系统项目的升级改造提供质量保证和控制，确保项目按时交付、满足用户需求，并提高系统的稳定性、性能和可靠性。通过本计划，将明确质量目标和要求，建立质量管理组织架构，确定质量确保对象，并制定相应的质量控制措施。

## 质量目标和要求

### 质量目标

为了确保项目质量，我们制定了以下质量目标：

**可用度：**

软件运行后在任一随机时刻需要执行规定任务或完成规定功能时软件处于可使用状态的概率应不大于98%。

**初期故障率：**

软件交付后三个月内每100小时故障率不大于5%。

**偶然故障率：**

软件交付后三个月内每1000小时故障率不大于2%。

**平均失效时间：**

在失效前正常工作的平均统计时间应大于365天。

**平均失效间隔时间：**

在相继两次失效之间正常工作的平均统计时间大于200天。

**缺陷密度：**

开发阶段为50－60缺陷，交付后为15－18缺陷。

**平均失效恢复时间：**

软件失效后恢复正常工作所需的平均统计时间。

### 完成目标所需方案

为了实现以上质量目标，我们将采取以下方案：

在测试部分，将给予足够的压力测试，以验证系统的可用性和稳定性。

售后部分将跟踪软件质量问题，及时解决问题，确保客户满意度。

项目经理将全程管控开发流程，合理分配时间，避免时间不足导致的软件质量问题。

## 质量管理组织架构

项目经理负责项目规划、执行、监控和结束的各个阶段。配置管理员负责配置项的标识和控制、配置项变更管理、配置库管理和支持项目管理。开发人员负责软件编码、测试和维护，以及技术研究和团队协作。具体见人力管理计划。

## 质量确保对象

### 文档对象

#### 文档质量保证对象

文档质量保证对象包含：

《需求规格说明书》

《质量保证计划》

《配置管理计划》

《成本计划》

《人力资源计划》

《沟通计划》

《软件设计文档》

《风险计划》

#### 文档质量衡量标准

**准确性：**

文档内容应准确无误，避免错误和误导。

**完整性：**

文档内容应包含所需的全部信息，不应漏掉任何重要细节。

**适应性：**

文档应当适应不同读者的需求和背景。考虑到读者的知识水平和背景差异，提供相应的解释、说明或附加材料，以确保文档对各类读者都具有适应性。

**更新性：**

文档应当及时更新以反映最新的信息和变化。随着时间的推移和项目的演进，确保文档的内容保持最新和准确，通过定期审查和更新来保持文档的更新性。

**一致性：**

文档应当保持一致性，即在整个文档中使用一致的术语、表达方式和格式。避免使用相互矛盾或混淆的术语，确保文档中的信息和表述保持一致，以提高读者的理解和阅读体验。

**可追溯性：**

文档应当具备可追溯性，即读者能够追溯文档中所引用的信息的来源和依据。为了确保文档的可信度和透明度，提供引用和参考的信息来源，以便读者能够进一步深入了解和核实所述内容。

**易更新性：**

文档应当易于进行修改和更新。为了满足不断变化的需求和反馈，确保文档具备易于编辑和修改的特性，使得更新文档变得简便和高效。

**清晰度：**

文档应该使用简单明了的语言，避免使用模棱两可或歧义的表述。

**可读性：**

文档应该易于阅读和理解，包括字体大小、排版、图表等方面的设计。

**可理解性：**

文档应该易于理解，包括专有术语的解释、概念的阐述等。

**可靠性：**

文档应该可靠，内容应该基于实际事实和数据，避免虚假或不准确的信息。

**可用性：**

文档应该易于使用，包括易于查找、易于导航等方面的设计。

### 代码对象

#### 代码质量保证对象

代码质量保证对象包括项目中的所有软件，包括.html文件、.css文件、.js文件、.json文件、.xml文件、.dll文件等。这些代码应具备可靠性、可读性、可维护性、性能效率和安全性等标准。

#### 代码质量衡量标准

**可靠性：**

代码质量应确保代码的执行稳定，没有明显的逻辑错误和异常情况。通过进行彻底的测试和调试，包括单元测试、集成测试和系统测试等，以验证代码的可靠性和稳定性。

**可读性：**

代码应具备良好的可读性，即代码结构清晰，命名规范，注释清晰，易于理解和维护。使用有意义的变量和函数命名，提供清晰的注释和文档，遵循一致的代码风格和格式，以使代码易于阅读和理解。

**可维护性：**

代码应具备良好的可维护性，即代码应该模块化，低耦合高内聚，易于修改和扩展。通过良好的设计原则和设计模式，将代码拆分为模块，使得每个模块都有清晰的职责和接口，并且修改一个模块不会影响其他模块的功能。

**性能效率：**

代码应具备高性能效率，即代码执行效率高，响应时间短，资源利用合理。通过优化算法和数据结构的选择，避免冗余计算和资源浪费，以提高代码的执行效率和性能。

**安全性：**

代码应具备必要的安全防护措施，防范潜在的安全风险和漏洞。使用安全编码的最佳实践，避免常见的安全漏洞，如SQL注入、跨站脚本攻击等。实施访问控制和数据验证等安全机制，以保护代码和系统的安全性。

**可靠性：**

代码质量应确保代码的执行稳定，没有明显的逻辑错误和异常情况。通过进行彻底的测试和调试，包括单元测试、集成测试和系统测试等，以验证代码的可靠性和稳定性。

**可读性：**

代码应具备良好的可读性，即代码结构清晰，命名规范，注释清晰，易于理解和维护。使用有意义的变量和函数命名，提供清晰的注释和文档，遵循一致的代码风格和格式，以使代码易于阅读和理解。

**可维护性：**

代码应具备良好的可维护性，即代码应该模块化，低耦合高内聚，易于修改和扩展。通过良好的设计原则和设计模式，将代码拆分为模块，使得每个模块都有清晰的职责和接口，并且修改一个模块不会影响其他模块的功能。

**性能效率：**

代码应具备高性能效率，即代码执行效率高，响应时间短，资源利用合理。通过优化算法和数据结构的选择，避免冗余计算和资源浪费，以提高代码的执行效率和性能。

**安全性：**

代码应具备必要的安全防护措施，防范潜在的安全风险和漏洞。使用安全编码的最佳实践，避免常见的安全漏洞，如SQL注入、跨站脚本攻击等。实施访问控制和数据验证等安全机制，以保护代码和系统的安全性。

## 质量计划

### 需求分析阶段的质量控制措施

在需求分析阶段，计划采取如下措施来确保质量控制：

**需求验证：**

与客户和利益相关者确认需求，确保需求的准确性和完整性。

**需求文档审查：**

对需求文档进行严格的审查和验证，确保文档清晰、一致、可追踪。

**需求变更管理：**

建立变更控制流程，对需求变更进行评估和控制，避免对开发过程造成不必要的干扰。

### 设计阶段的质量控制措施

在设计阶段，计划采取如下措施来确保质量控制：

**设计审查：**

对系统设计进行技术审查，确保设计符合项目要求和最佳实践。

设计文档质量控制：

对设计文档进行审查，确保文档准确、完整、易读，与需求一致。

**设计复审：**

组织设计团队进行复审，识别潜在的问题和改进点，并进行必要的调整和优化。

### 编码和测试阶段的质量控制措施

在编码和测试阶段，计划采取如下措施来确保质量控制：

**编码规范和标准：**

制定统一的编码规范和标准，确保代码的可读性、可维护性和一致性。

**代码审查：**

进行代码审查，发现和修复潜在的缺陷和错误，提高代码质量。

**自动化测试：**

建立自动化测试框架，执行单元测试、集成测试和系统测试，确保代码质量和功能的正确性。

### 上线和运维阶段的质量控制措施

在上线和运维阶段，计划采取如下措施来确保质量控制：

**上线前测试：**

进行系统集成测试和用户验收测试，确保系统的稳定性和完整性。

**系统监控和日志管理：**

建立系统监控和日志管理机制，及时发现和解决问题，保障系统的正常运行。

**定期维护和更新：**

制定维护计划，定期进行系统维护和更新，修复漏洞和改进功能，确保系统持续稳定运行。

## 质量检查和审核

质量检查和审核将在不同阶段进行，审计内容包括需求分析、设计、编码、测试和项目验收等。审计报告将通过内部会议和讨论、缺陷跟踪工具、文件和报告、项目管理工具等方式提交和记录。质量保证人员将与项目经理、高层管理者和其他相关人员合作，确保发现的问题得到纠正和改进。

# 人力管理计划

本人力管理计划旨在明确项目团队中各成员的角色和职责，并通过RACI模型来确保任务的顺利执行和有效的协作。本计划涵盖了十个成员的分工和职责，以及相应的人力资源管理措施。

## 项目团队成员

|  |  |  |
| --- | --- | --- |
| **成员姓名** | **角色** | **职责** |
| 郑乐祺 | 项目经理 | - 全面负责项目的执行、控制和完成  - 协调团队内部和项目各方的交流 |
| 李四 | 业务分析员 | - 负责需求调研、收集和分析  - 与客户保持密切的沟通 |
| 王五 | 架构师 | - 负责项目的技术架构设计  - 与开发团队紧密合作，指导技术实现 |
| 赵六 | 前端开发工程师 | - 负责前端界面的设计和实现  - 与后端开发工程师密切合作，保证系统流畅运行 |
| 孙七 | 后端开发工程师 | - 负责后端系统的开发  - 负责数据库和服务器的运行维护 |
| 周八 | 数据库管理员 | - 负责数据库设计、维护和优化  - 确保数据安全和稳定 |
| 吴九 | 测试工程师 | - 负责测试计划和用例的制定  - 执行测试，保证系统质量 |
| 郑十 | UI/UX设计师 | - 负责用户界面和体验的设计  - 与前端开发工程师紧密合作 |
| 钱十一 | 技术支持 | - 负责系统的上线、运维  - 提供用户的技术支持 |
| 刘十二 | 文档工程师 | - 负责项目文档的撰写和管理  - 包括需求文档、设计文档、测试文档等 |

## RACI模型

谁负责（R = Responsible），负责执行任务的角色，具体负责操控项目、解决问题。

谁批准（A = Accountable），对任务负全责的角色，只有经其同意或签署之后，项目才能得以进行。

咨询谁（C = Consulted），在任务实施前或中提供指定性意见的人员。

告知谁（I = Informed），及时被通知结果的人员，不必向其咨询、征求意见。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 成员姓名 | 角色 | R (负责人) | A (审批人) | C (协调人) | I (参与者) |
| 张三 | 项目经理 | R | A | C | I |
| 李四 | 业务分析员 | R | A | C | I |
| 王五 | 架构师 | R | A |  | I |
| 赵六 | 前端开发工程师 | R |  | C | I |
| 孙七 | 后端开发工程师 | R |  | C | I |
| 周八 | 数据库管理员 | R |  |  | I |
| 吴九 | 测试工程师 | R |  | C | I |
| 郑十 | UI/UX设计师 | R |  | C | I |
| 钱十一 | 技术支持 | R |  |  | I |
| 刘十二 | 文档工程师 | R |  | C | I |

## 人力资源管理措施

招聘和选聘合适的团队成员，确保每个角色都有足够的能力和专业知识。

提供适当的培训和发展机会，以提高团队成员的技能和知识水平。

建立良好的沟通渠道，确保团队成员之间的有效沟通和协作。

设定明确的目标和绩效评估标准，激励团队成员的积极表现。

定期进行团队会议和项目进度评审，及时解决问题和调整资源分配。

提供必要的工具和资源，支持团队成员顺利完成任务。

建立良好的工作氛围，鼓励团队合作和知识共享。

# 沟通管理计划

## 项目沟通

使用”禅道”对整个项目进行任务的管理，项目经理管理该项目管理工具，应及时在各阶段给出相应的任务，各成员通过个人”禅道”账号作为协作者进入项目管理页面后可以看到个人的任务，并且在任务完成后应及时在“禅道”中汇报情况。

## 会议沟通

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **会议** | **沟通内容** | **沟通目的** | **沟通渠道** | **文档** | **沟通时间** | **沟通对象** | **负责人** |
| 例会 | 每周工作总结及计划安排；存在的问题及解决办法 | 每周二的会议将围绕上周的工作总结及新的计划安排，另外，也将讨论存在的问题和解决方案。通过这样的方式，可以确保每个人都了解项目的进展和接下来的计划，同时也可以及时提出存在的问题和提供解决的办法，这对于保证项目的顺利进行非常重要。 | 视频会议 | 《会议记录》、《项目周报》 | 每周一上午7:00 | 例会：项目小组人员；文档：与会人员、抄送领导小组、公司领导 | 郑乐祺 |
| **项目阶段性总结会议** | 项目阶段性总结 | 每到一个项目里程碑，都需要进行阶段性的总结。这个总结应该涵盖在这个阶段内所完成的工作，达到的目标以及存在的问题等。通过这个总结，所有人都能了解项目阶段性的成果，同时也能够针对存在的问题提出改进的策略。 | 视频会议 | 《阶段性总结报告》 | 里程碑日 | 项目小组人员；文档：与会人员、抄送领导小组、公司领导 | 郑乐祺 |
| **项目启动会议** | 标志项目启动，确定人员分工 | 项目启动会议是标志着项目正式启动的一个重要活动。在这个会议中，主要是确定人员的分工，明确每个人的职责和任务。通过这个会议，可以确保每个人都明白自己的职责和目标，为接下来的项目工作做好准备。 | 视频会议 | 《会议纪要》、项目规章制度 | 项目启动当天上午 | 甲方主要领导、项目组主要成员、项目业务人员 | 郑乐祺 |
| **需求分析会议** | 确定甲方需求，制定项目计划 | 需求分析会议主要是为了确定甲方的需求，制定符合这些需求的项目计划。在这个会议中，应该详细讨论甲方的需求，理解这些需求的背景和目标，以便能够制定出满足这些需求的项目计划。 | 视频会议 | 《需求规格说明书》、《会议记录》 | 项目启动当天下午 | 会议：甲方项目负责人，我方项目组主要成员；文档：与会人员 | 郑乐祺 |
| **总体实施方案会议** | 汇报方案，听取对方的意见，最终双方确认 | 总体实施方案是描述项目如何进行的一个重要文件。在这个活动中，需要汇报方案，听取甲方的意见，最终达成双方的确认。这个活动对于确保项目按照计划进行非常重要。 | 邮件、视频会议 | 《项目总体方案》、《会议记录》 |  | 甲乙双方所有人员 | 郑乐祺 |
| **调研会议** | 模拟实际业务，发现软件和实际之间的问题 | 在这个活动中，需要模拟实际业务，以此来发现软件和实际业务之间的问题。这个活动对于保证软件能够满足实际业务的需求非常重要。 | 视频会议 | 《会议记录》 |  | 项目组主要成员 | 郑乐祺 |
| **软件问题交流会议** | 软件问题 | 在项目过程中，如果产生了软件问题，应该及时地将问题通知公司的开发经理和项目经理，同时也应该对问题进行存档，以供将来的参考和解决。这个活动对于保证项目的质量非常重要。 | 邮件、电话、视频会议 | 《问题反馈单》 | 产生软件问题时 | 相关开发人员、开发经理、项目经理 | 郑乐祺 |
| **任务开始会议** | 任务分配及控制 | 在每项任务开始前，需要进行任务的分配和控制。这包括确定任务的执行人，描述任务的详细信息，以及跟踪任务的执行情况。这个活动对于确保任务能够按照计划进行非常重要。 | 邮件 | 《任务单》 | 每项任务开始前 | 任务执行人、项目经理 | 郑乐祺 |
| **任务结束总结会议** | 任务完成质量 | 在每项任务结束后，需要跟踪任务的完成质量。这包括评估任务的完成情况，检查是否满足预定的质量标准，以及提出改进的策略。这个活动对于保证项目的质量非常重要。 | 电话/邮件/谈话 | -无 | 每项任务结束 | 甲方 | 郑乐祺 |
| **日常交流电话会议** | 项目组交流 | 在项目过程中，需要定期地进行项目组的交流。在这个活动中，项目组的成员可以分享他们对于项目的想法和建议，也可以提出存在的问题和解决的办法。这个活动对于保证项目的顺利进行非常重要。 | 谈话 | 无- | 不定期 | 谈话：项目组成员 | 郑乐祺 |
| **重要事件**  **交流会议** | 交流会 | 当项目发生重要的事件时，需要召开交流会。在这个会议中，可以讨论这个事件的影响，提出应对的策略，以及对项目进行重新的规划。这个活动对于应对项目中的突发事件非常重要。 | 专题视频会议 | 《备忘录》/《会议记录》 | 项目发生重要事件 | 项目小组所有成员 | 郑乐祺 |

# 风险管理计划

在项目管理的过程中，风险管理起着关键的作用。正确地处理风险不仅能防止项目出现问题，而且能提高项目的整体效益。因此，制定一份全面且实用的风险管理计划是至关重要的。

制定风险管理计划的目的是通过系统地识别和分析潜在的项目风险，制定有效的风险应对措施，以减少风险对项目进度、质量和成本的影响。通过此风险管理计划，我们希望提高项目团队对风险的认识，改进风险管理的实践，确保项目的成功。

## 风险条目与评估

### 技术风险

#### 数据安全风险

随着信息技术的发展，数据安全已经成为一个严重的问题。项目在处理大量敏感数据时，如果没有充足的安全措施，可能会导致数据泄露，带来严重的后果。此风险的概率评估为4，影响评估为5。

#### 技术实现风险

项目采用了大数据处理、云计算等先进技术。虽然这些技术可以带来很大的效益，但在实施过程中可能会遇到技术难题，导致项目延期或超出预算。此风险的概率评估为3，影响评估为4。

#### 数据读写性能风险

由于系统需要处理大量的数据，必须确保数据读写性能满足系统要求，否则可能会导致系统响应速度变慢，甚至崩溃。此风险的概率评估为3，影响评估为4。

#### 系统稳定性风险

由于系统需要处理大量的数据，必须确保系统的稳定性，否则可能会导致系统崩溃、数据丢失等问题。此风险的概率评估为3，影响评估为4。

#### 安全性风险

由于云南省企业就业失业数据采集系统需要处理大量的敏感数据，必须确保系统的安全性，否则可能会泄露敏感数据，导致安全问题。此风险的概率评估为4，影响评估为5。

#### 系统架构风险

由于系统需要处理大量的数据，必须确保系统架构的合理性，否则可能会导致系统性能下降、稳定性下降等问题。此风险的概率评估为3，影响评估为4。

#### 据备份和恢复风险

由于系统需要处理大量的数据，必须确保数据备份和恢复机制的有效性，否则可能会导致数据丢失，影响系统的正常运行。此风险的概率评估为3，影响评估为4。

### 业务风险

#### 用户需求变更风险

用户需求可能在项目执行过程中发生变更，这可能导致项目范围扩大，增加项目成本和时间。此风险的概率评估为3，影响评估为4。

#### 数据准确性风险

由于系统需要采集和处理各种企业就业失业数据，必须确保数据的准确性，否则可能会对后续数据分析和处理造成困难。此风险的概率评估为3，影响评估为4。

#### 数据完整性风险

由于系统需要采集和处理各种企业就业失业数据，必须确保数据的完整性，否则可能会对后续数据分析和处理造成困难。此风险的概率评估为3，影响评估为4。

#### 业务流程风险

由于系统需要处理大量的企业就业失业数据，必须确保业务流程的合理性，否则可能会导致系统响应速度变慢，甚至崩溃。此风险的概率评估为3，影响评估为4。

#### 复杂业务流程风险

本项目涉及的业务流程复杂，涉及多个部门和角色。如果没有清晰的业务流程描述和有效的协调机制，可能导致项目实施困难。此风险的概率评估为4，影响评估为4。

### 法律风险

#### 数据合规风险

项目需要遵守相关的数据处理和保护法规。如果违反这些法规，可能会导致法律诉讼，严重影响项目的进展。此风险的概率评估为2，影响评估为5。

#### 合同法律风险

由于系统需要与其他企业进行合同签订和履行，必须确保合同法律方面的合规性，否则可能会面临法律风险。此风险的概率评估为3，影响评估为4。

#### 信息安全风险

由于系统需要处理大量的敏感数据，必须确保信息安全方面的法律法规的遵守，否则可能会泄露敏感数据，导致安全问题。此风险的概率评估为4，影响评估为5。

#### 商标法律风险

由于系统涉及到大量的商标，必须确保商标法律方面的保护，否则可能会侵犯他人的商标权，导致法律纠纷。此风险的概率评估为3，影响评估为4。

### 人力资源风险

#### 人力资源不足风险

如果项目团队人力资源不足，可能导致项目无法按照计划进行，甚至可能导致项目失败。此风险的概率评估为3，影响评估为4。

#### 人力资源技术欠缺风险

人力资源技术欠缺风险指的是项目团队在执行项目过程中缺乏必要的技术能力和知识，无法有效地完成任务或应对项目挑战。这可能导致项目进展缓慢、质量下降或出现错误，进而影响项目的成功完成。该风险的概率评估为2，影响评估为3。

#### 项目管理经验缺失风险

项目管理经验缺失风险是指项目团队成员在项目管理方面缺乏充分的经验和知识，无法有效地规划、执行和监控项目活动。这可能导致项目计划不准确、进度延误、沟通问题以及决策错误，从而增加项目失败的风险。该风险的概率评估为3，影响评估为3。

### 软件测试不足风险

软件测试是保证软件质量的重要环节。如果测试不充分，可能导致软件出现问题，影响用户体验，甚至影响项目的成功。此风险的概率评估为3，影响评估为4。

### 项目延期风险

项目可能由于各种原因（如技术难题、需求变更、人力资源不足等）导致延期。项目延期可能导致成本增加，影响项目效益。此风险的概率评估为3，影响评估为4。

### 风险评估结果表

|  |  |  |  |
| --- | --- | --- | --- |
| **风险条目** | **风险概率** | **风险影响** | **风险级别** |
| 数据安全风险 | 4 | 5 | 高 |
| 技术实现风险 | 3 | 4 | 中 |
| 数据读写性能风险 | 3 | 4 | 中 |
| 系统稳定性风险 | 3 | 4 | 中 |
| 安全性风险 | 4 | 5 | 高 |
| 系统架构风险 | 3 | 4 | 中 |
| 数据备份和恢复风险 | 3 | 4 | 中 |
| 用户需求变更风险 | 3 | 4 | 中 |
| 数据准确性风险 | 3 | 4 | 中 |
| 数据完整性风险 | 3 | 4 | 中 |
| 业务流程风险 | 3 | 4 | 中 |
| 复杂业务流程风险 | 4 | 4 | 中 |
| 数据合规风险 | 2 | 5 | 高 |
| 知识产权风险 | 3 | 4 | 中 |
| 合同法律风险 | 3 | 4 | 中 |
| 信息安全风险 | 4 | 5 | 高 |
| 商标法律风险 | 3 | 4 | 中 |
| 人力资源不足风险 | 3 | 4 | 中 |
| 人力资源技术欠缺风险 | 2 | 3 | 中 |
| 项目管理经验缺失风险 | 3 | 3 | 中 |
| 软件测试不足风险 | 3 | 4 | 中 |
| 项目延期风险 | 3 | 4 | 中 |

这里，风险概率和风险影响的评估标准为1-5（1表示影响很小，5表示影响非常大）。

## 风险定性分析

风险定性分析的目的是通过对风险条目进行详细的分析，进一步理解每个风险的性质和可能的影响。这个过程包括了解风险发生的原因，分析风险发生后可能的结果，以及确定风险的影响程度。

具体的定性分析方法包括：风险影响评估、风险概率评估、风险优先级分析。通过定性分析，我们可以确定哪些风险应该优先处理，以及如何有效地处理这些风险。下面是风险定性分析表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 风险类别 | 事件 | 发生概率 | 影响 | 事件发生后果 |
| 技术风险 | 新技术研发失败 | 中 | 严重 | 系统开发延误 |
| 技术风险 | 技术选型错误 | 中 | 严重 | 系统功能缺失 |
| 业务风险 | 业务需求变更频繁 | 高 | 中度 | 项目延期、成本增加 |
| 业务风险 | 业务需求理解偏差 | 中 | 严重 | 业务需求无法实现 |
| 业务风险 | 客户对目标用户规模不够确定 | 低 | 轻度 | 系统不稳定 |
| 业务风险 | 与需求分析人员沟通不协调 | 中 | 中度 | 系统不能满足业务需求 |
| 业务风险 | 客户对目标功能不够清楚 | 高 | 轻度 | 系统不能满足业务需求 |
| 法律风险 | 法规改变 | 低 | 高 | 项目合规性降低 |
| 法律风险 | 合同争议 | 低 | 高 | 项目延期、成本增加 |
| 人力资源风险 | 关键人员离职 | 中 | 高 | 项目延期、成本增加 |
| 人力资源风险 | 人员技术能力不足 | 中 | 中度 | 项目质量降低 |
| 软件测试风险 | 软件缺陷未能及时发现 | 高 | 严重 | 系统质量降低 |
| 软件测试风险 | 测试环境不稳定 | 中 | 中度 | 系统上线延误 |
| 时间风险 | 项目进度延误 | 高 | 高 | 项目延期、成本增加 |
| 时间风险 | 时间评估过于乐观 | 中 | 高 | 项目延期 |

## 应对措施

### 风险应对总体策略

**风险避免：**

一些高风险，例如法规风险，应该尽可能避免。对于数据合规风险，我们应该在项目开始时就充分了解相关法规，确保项目活动符合法规要求。

**风险减轻：**

对于一些不能避免但可能导致严重影响的风险，如数据安全风险和复杂业务流程风险，我们可以通过风险减轻来降低其影响。例如，我们可以采取严格的数据保护措施来减轻数据安全风险，也可以优化业务流程来降低复杂业务流程风险。

**风险接受：**

对于一些低概率或低影响的风险，如用户需求变更风险，我们可能选择接受这些风险，但要做好应对措施，以防风险真的发生时能够快速应对。

### 技术风险应对策略

**新技术研发失败：**

在项目开始前，充分调研新技术的可行性，尽量选择成熟且稳定的技术进行项目开发。如果必须使用新技术，应在项目开始前进行足够的技术储备和预研工作。

**技术选型错误：**

在技术选型时，充分比较不同技术的优缺点，根据项目需求和团队技术能力进行选择。同时，持续关注所选技术的发展情况和市场反馈。

### 业务风险应对策略

**业务需求变更频繁：**

在需求收集阶段，充分理解并沟通需求，确定需求优先级，对于非关键需求，可以推迟实现。在项目执行过程中，控制需求变更，变更必须经过严格审批。

**业务需求理解偏差：**

建立业务培训机制，确保项目团队对业务有深入理解。同时，需求文档应详尽完备，业务流程应清晰明确。

### 法律风险应对策略

**法规改变：**

关注相关法律法规的最新动态，对法规改变及时做出应对，如调整项目计划或设计。

**合同争议：**

在合同签订之初，对合同条款进行详细讨论和确认，避免合同模糊不明引发的争议。

### 人力资源风险应对策略

**关键人员离职：**

对关键人员进行适当的激励，如提供发展机会、调整薪资待遇等。同时，建立人员备份机制，确保关键人员的工作可以由其他人员接替。

**人员技术能力不足：**

定期进行技术培训和技术分享，提升团队技术能力。对于复杂技术问题，可以寻求外部专家支持。

### 软件测试风险应对策略

**软件缺陷未能及时发现：**

建立完整的软件测试流程，包括单元测试、集成测试、系统测试等。同时，使用自动化测试工具，提高测试效率和测试覆盖率。

**测试环境不稳定：**

建立稳定的测试环境，保证测试环境和生产环境一致。定期对测试环境进行维护，及时解决测试环境中的问题。

### 时间风险应对策略

**项目进度延误：**

制定实际可行的项目计划，监控项目进度，对延误情况进行及时调整。如果可能，采取并行工作方式，提高工作效率。

**时间评估过于乐观：**

采用经验数据进行时间评估，避免过于乐观的评估。同时，对于关键任务，留出一定的缓冲时间。

## 风险监控和报告

风险管理是一个持续的过程，需要在项目全过程中进行风险监控和报告。风险监控包括定期评估风险状态，检查风险应对措施的执行情况，以及对新出现的风险进行识别和分析。风险报告则需要向项目管理团队和相关的利益相关者报告风险管理的情况，以便及时调整风险应对策略。

在本项目中，我们团队使用“禅道”项目管理工具进行风险管理，由项目经理郑乐祺进行总体把控。

# 合同管理计划

## 合同管理团队

项目经理：

负责整体的合同管理，包括合同的审查、批准和变更管理。

采购经理：

负责与供应商的沟通，以及合同的执行和管理。

法务顾问：

负责提供合同相关的法律咨询和支持。

## 合同类型

根据项目需求和风险，我们使用固定价格合同、成本补偿合同或时间和材料合同。

## 合同审查和批准

所有合同必须经过项目经理和法务顾问的审查和批准。审查过程将确保合同符合项目需求，并且合同条款明确、公平。

## 合同执行

采购经理将负责合同的执行，包括确保供应商按照合同的要求提供服务或产品，以及处理合同中的任何问题或争议。

## 合同变更管理

任何合同的变更都必须经过正式的变更控制过程。

变更请求必须提交给项目经理，项目经理将评估变更的影响，并决定是否批准。

所有的变更请求和决定都必须进行记录。

## 合同支付

采购经理将负责合同的支付，包括验证供应商的发票，以及处理任何与支付相关的问题。

## 合同终止

如果供应商未能按照合同的要求提供服务或产品，我们可能会考虑终止合同。

合同终止的决定将由项目经理和法务顾问共同做出。

## 记录和文档管理

所有的合同文档和记录都必须保存在一个安全的地方，以便于审计和参考。

# 配置管理计划

本文档是软件配置管理计划，旨在确保项目的软件配置项得到有效的控制和管理。配置管理是项目管理的重要环节，通过标识、版本控制、变更管理和发布交付等活动，保证软件配置的一致性和可控性。本计划将详细描述配置管理的过程、角色和责任，并提供指导和规则，确保团队正确进行配置管理工作。通过有效的配置管理，我们能降低风险、提高效率，保证项目交付的质量和时间性。

## 软件配置管理

### 软件配置管理组织

在本项目中，涉及以下角色和责任：

**项目经理：**

角色：郑乐祺

职责、工作范围：

与各项目组代表一起审批配置管理计划。

审批建库申请表。

与项目成员一起审批配置项的变更、基线化、发布申请。

检查配置管理计划完成情况。

**项目组配置管理员：**

角色：郑乐祺

职责、工作范围：

制定《配置管理计划》。

搭建配置库结构。

申请并配合建立配置库。

配置库的管理。

准备、申请、并实施基线化（或发布入库）工作。

将建库和入库情况及时通知各项目成员。

备份、维护基线库（或发布库）。

配合高级配置管理员完成配置管理状态报告。

**高级配置管理员：**

角色：郑乐祺

职责、工作范围：

根据实际情况审批建库申请，分配服务器资源，完成建库。

管理配置库的用户帐号、权限。

对配置库做物理审计。

配合项目配置管理员完成配置管理状态报告。

定期编制产品库定期报告。

**项目组成员：**

角色：郑乐祺

职责、工作范围：

了解并按权限正确使用配置库。

配合项目组配置管理员准备、申请基线化（或发布入库）工作。

**开发人员：**

角色：郑乐祺

职责、工作范围：

配合配置管理工作，提交配置项和基线变更请求。

### 软件过程生命周期

在本项目中，采用敏捷开发模型，强调灵活性、快速响应和迭代开发。我们将采用短期迭代的方式进行开发，团队紧密合作，通过持续反馈和调整，快速交付高质量的软件。

## 软件配置管理活动

### 配置项标识

为确保每个配置项具有唯一的标识符、可读性高且一致性强，我们采用以下配置项命名规范：

独立文档命名格式：BLG-YUNNANSYS-【编号】-【文档名称】-【版本号v m.n】

说明：BLG为公司名称百丽宫的缩写，YUNNANSYS代表云南人力资源市场数据采集的缩写，编号规则为年份+项目编号+文档编号。例如202301001表示2023年第一个项目10号文件。

示例：BLG-YUNNANSYS-202301010-SRS-v1.0

定期更新文档命名格式：BLG-YUNNANSYSJ-【编号】-【文档名称】-【更新日期】

示例：BLG-YUNNANSYS-202301022-ProjectLog-0601

说明：上例表示6月1日项目日志。

### 配置项计划

|  |  |  |  |
| --- | --- | --- | --- |
| 项目阶段 | 标识符 | 预计入库时间 | 产出物说明 |
| 招投标 | BLG-YUNNANSYS-202301002-QAP-v1.0 | 2023年04月28日 | 技术规范书 |
| 项目启动 | BLG-YUNNANSYS-202301003-CMP-v1.0 | 2023年04月30日 | 立项、管理制度、开发及测试计划、风险管理文档、项目监控、规范、QA等 |
| 需求开发 | BLG-YUNNANSYS-202301004-PSchedule-v1.0 | 2023年05月04日 | 业务需求、软件需求、用例模型、度量、评审、QA等 |
| 设计 | BLG-YUNNANSYS-202301005-PCP-v1.0 | 2023年05月11日 | 框架设计、概要设计、详细设计、度量、评审、QA、需求变更等 |
| 代码开发 | BLG-YUNNANSYS-202301006-HRP-v1.0 | 2023年05月31日 | 代码、开发指南、度量、评审、QA、需求变更等 |
| 测试 | BLG-YUNNANSYS-202301007-RMP-v1.0 | 2023年06月20日 | 测试用例、压力及性能测试等各种测试文档、度量、QA、需求变更等 |
| 部署 | BLG-YUNNANSYS-202301008-PCP-v1.0 | 2023年06月25日 | 部署文档、用户手册、度量、QA、需求变更等 |
| 预验收 | BLG-YUNNANSYS-202301009-DS-v1.0 | 2023年06月30日 | 结项、用户验收、技术验收及交付文档、QA、需求变更等 |
| 维护 | BLG-YUNNANSYS-202301010-SADB-v1.0 | 2023年07月01日 | 代码、各种文档的修改、QA、需求变更等 |

### 配置基线

在本项目中，配置库基线由里程碑基线和日常开发基线共同组成。里程碑基线分为前期基线、计划基线、需求基线、设计基线、代码基线、测试基线和产品基线。其中，前期基线可裁减。本项目中配置库基线分为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **基线类别** | **配置项** | **预计建立时间** | **需求** | **说明** |
| 需求基线 | 需求规格说明书 | 2023年05月01日 | 需求规格说明书 | 基于需求分析和讨论，编写详细的需求规格说明书，确立项目需求的基准。 |
| 总体设计基线 | 系统结构设计、数据库设计 | 2023年05月06日 | 进行系统结构设计和数据库设计 | 基于需求和技术架构，进行系统总体设计和数据库设计，为项目的具体实现提供指导。 |
| 项目开发基线 | 源程序 | 2023年05月11日 | 开发源程序 | 完成关键功能的源程序开发，标志着项目正式进入开发阶段。 |
| 系统测试基线 | 测试计划、测试报告 | 2023年06月01日 | 制定测试计划并生成测试报告 | 制定系统测试计划，进行系统功能和性能的测试，并记录测试结果和问题反馈。 |

### 配置库结构

为便于管理和组织，配置库采用以下结构：

RM：需求管理

CM：软件配置管理

ComM：沟通管理计划

ExM：成本管理计划

QA：软件质量保证

SM：进度管理计划

forum：表单库

riskM：风险管理计划

## 软件配置审核

按照配置管理规程，定期对配置库和配置项的状态进行审核。审核包括基线的完整性、配置记录的准确性、配置库和配置项结构的完整性、配置项的完备性和正确性、审核后的行动跟踪、配置项变更的状态和正确性以及配置库的操作和备份等方面。审核结果将记录在《配置状态报告》中。

具体计划如下：

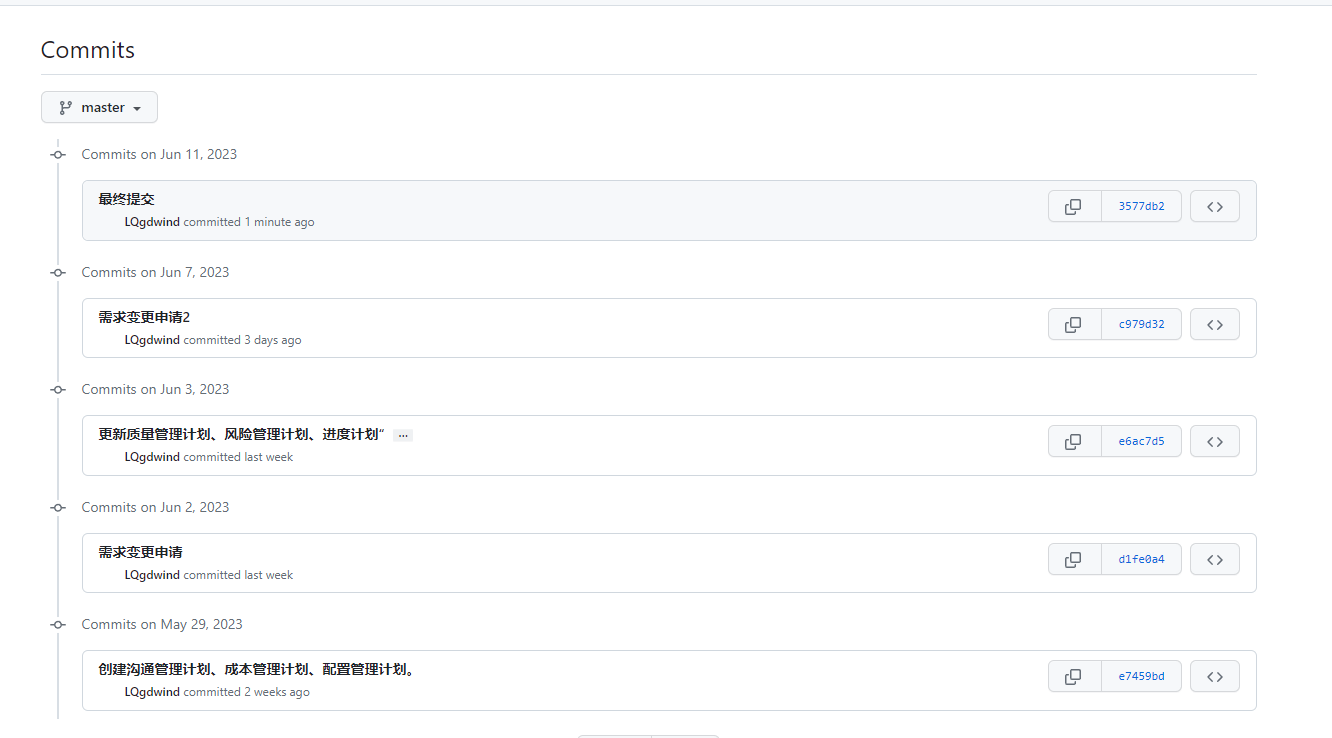
|  |  |  |  |
| --- | --- | --- | --- |
| **审核内容** | **时间计划/频率** | **审核人** | **审核对象和方式** |
| 基线完整性审核 | 1次/周 | 郑乐祺 | 审核基线是否完整。如果基线不完整，则对基线进行调整。 |
| 配置记录审核 | 1次/周 | 郑乐祺 | 审核配置管理记录是否正确反映了配置项的配置情况。 |
| 审查配置库和配置项的结构 | 1次/周 | 郑乐祺 | 根据《配置管理计划》审查配置管理系统中配置项的结构完整性。 |
| 审查配置项的完备性和正确性 | 1次/周 | 郑乐祺 | 以《配置管理计划》中说明的需求和所批准的变更请求的处置为基础来验证配置项是否符合规定的配置管理和质量管理要求。 |
| 配置库操作和备份 | 每日 | 郑乐祺 | 审核每日配置库操作的备份情况，确保数据的安全性。 |

## 软件配置控制

### 版本管理

版本管理是软件配置管理的重要环节，我们采用Git进行版本管理。技术总监将负责版本的控制工作，严格按照变更控制过程进行版本的变更。

截图如下：



版本管理将包括以下内容：

**创建项目：**

在Git上创建项目，设置权限和分支等。

**克隆项目：**

将项目克隆到本地，使用Git命令进行代码提交和管理。

**创建分支：**

在Git上创建新的分支，分支可用于开发新功能或修复缺陷。

**编写代码：**

在本地编写代码，使用Git命令进行代码提交和管理。

**提交代码：**

将本地代码提交到Git上的分支中，以便其他开发人员进行合并和审核。

**合并代码：**

经过审核后，将代码合并到主分支中，确保代码的稳定性和可维护性。

**发布版本：**

代码合并到主分支后，可打标签并发布版本，以方便用户下载和使用。

**回滚代码：**

若出现问题，可使用Git命令将代码回滚到之前的版本，以保证代码的安全性和稳定性。

### 项目管理

本项目将使用"禅道"管理软件，对整个项目进行管理。项目经理作为负责人，将在"禅道"上进行项目、产品、测试等管理工作。禅道是一款全功能的项目管理工具，支持任务管理、缺陷管理、文档管理、项目计划、团队协作等功能。通过在禅道上建立项目和相应的任务、缺陷等，团队成员可以方便地进行沟通、协作和追踪工作进展。