**山东省企业就业失业数据采集系统**

**项目计划**

计算机学院 软件工程

学号：1120173805 班级：08111706

姓名：任翔渝

目录

[一、 导言 4](#_Toc38139255)

[1.1 目的 4](#_Toc38139256)

[1.2 范围 4](#_Toc38139257)

[1.3 缩写说明 4](#_Toc38139258)

[1.4 术语定义 4](#_Toc38139259)

[1.5 版本更新记录 4](#_Toc38139260)

[二、 项目概述 5](#_Toc38139261)

[三、 项目任务范围 6](#_Toc38139262)

[3.1 整体功能 6](#_Toc38139263)

[3.2 任务分解结构（WBS） 7](#_Toc38139264)

[3.3 可交付成果 8](#_Toc38139265)

[四、 项目目标 8](#_Toc38139266)

[五、 项目组织结构 9](#_Toc38139267)

[5.1 各角色职责 9](#_Toc38139268)

[5.2 人力资源及角色映射 10](#_Toc38139269)

[六、 项目生存期 11](#_Toc38139270)

[七、 项目进度计划 13](#_Toc38139271)

[八、 项目成本计划 15](#_Toc38139272)

[8.1 成本计划概述 15](#_Toc38139273)

[8.2 项目成本估算 15](#_Toc38139274)

[8.3 项目成本预算 19](#_Toc38139275)

[九、 质量管理计划 22](#_Toc38139276)

[9.1 质量目标 22](#_Toc38139277)

[9.2 质量管理职责 22](#_Toc38139278)

[9.3 质量管理流程 23](#_Toc38139279)

[9.4 质量活动 24](#_Toc38139280)

[十、 配置管理计划 27](#_Toc38139281)

[10.1 配置管理流程 27](#_Toc38139282)

[10.2 配置项标识 28](#_Toc38139283)

[10.3 配置库建立 28](#_Toc38139284)

[10.4 入库程序 29](#_Toc38139285)

[10.5 出库程序 29](#_Toc38139286)

[10.6 基线变更程序 30](#_Toc38139287)

[十一、 项目风险计划 30](#_Toc38139288)

[11.1 风险识别 30](#_Toc38139289)

[11.2 风险评估 31](#_Toc38139290)

[11.3 风险应对策略 33](#_Toc38139291)

[十二、 项目沟通计划 36](#_Toc38139292)

[12.1 外部协调 36](#_Toc38139293)

[12.2 内部沟通 36](#_Toc38139294)

[12.3 沟通方式说明 38](#_Toc38139295)

# 导言

## 目的

本文档的目的是为山东省企业就业失业数据采集系统提供实施计划，其主要内容包括：

* 项目范围和目标
* 项目的组织和管理方式
* 项目的生存期和交付物
* 时间计划和成本计划

## 范围

本文档定义了项目实施的方式和计划，未定义项目实施的过程规范和产品标准，有关内容请查阅相关文档。

## 缩写说明

WBS：Work Breakdown Structure的缩写。

CM：Configuration Management的缩写。

CI：Configuration Item的缩写。

QA：Quality Assurance的缩写。

SWOT：Strength、Weakness、Opportunity、Threads的缩写。

EV：Expected Value的缩写。

## 术语定义

无。

## 版本更新记录

本文档的修订记录如表1所示

|  |  |  |
| --- | --- | --- |
| 版本 | 修改内容 | 日期 |
| 0.1 | 增加：范围计划 | 2020.3.24 |
| 0.2 | 增加：进度计划 | 2020.3.27 |
| 0.3 | 增加：成本计划、质量计划 | 2020.4.2 |
| 0.4 | 增加：人力资源和沟通计划 | 2020.4.3 |
| 0.5 | 增加：风险管理计划 | 2020.4.5 |
| 0.6 | 增加：配置管理计划 | 2020.4.6 |
| 1.0 | 集成项目计划 | 2020.4.6 |
| 1.1 | 需求变更：要能从手机上实现报送 | 2020.4.6 |
| 1.2 | 更新：范围计划相关内容 | 2020.4.10 |
| 1.3 | 更新：进度计划相关内容 | 2020.4.15 |
| 1.4 | 更新：成本计划相关内容 | 2020.4.16 |
| 1.5 | 更新：质量计划相关内容 | 2020.4.17 |
| 1.6 | 更新：配置管理计划、风险计划相关内容 | 2020.4.18 |
| 2.0 | 需求变更后的项目计划发布 | 2020.4.18 |

表1 版本更新记录

# 项目概述

本项目将开发出一套全方位的企业就业失业数据采集系统，该系统具有极强稳定性、规范性、实用性、经济型、安全性、可靠性、先进性、可扩展性。通过本次项目的实施与开发，将满足省级用户企业管理、报表管理、数据管理、数据分析、通知管理、系统管理的需求，使企业级用户可以方便、快捷的实现企业备案、数据填报、数据查询。本系统在充分考虑多源数据接口、数据之间的关联以及网络环境开放性的基础上，以完备的信息数据库为基础，以开放的信息共享平台为依托，实现了省级部门与各企业的交互与信息共享，大大提高了工作效率，规范了工作流程，加强了部门协同，将全面提升管理与服务质量。

# 项目任务范围

## 整体功能

山东省企业就业失业数据采集系统主要针对省级用户和企业用户进行开发，分为省级用户的企业查询、企业备案信息、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、系统管理等功能和企业用户的备案上报、数据填报、数据查询功能。如图1所示的整体功能预览。

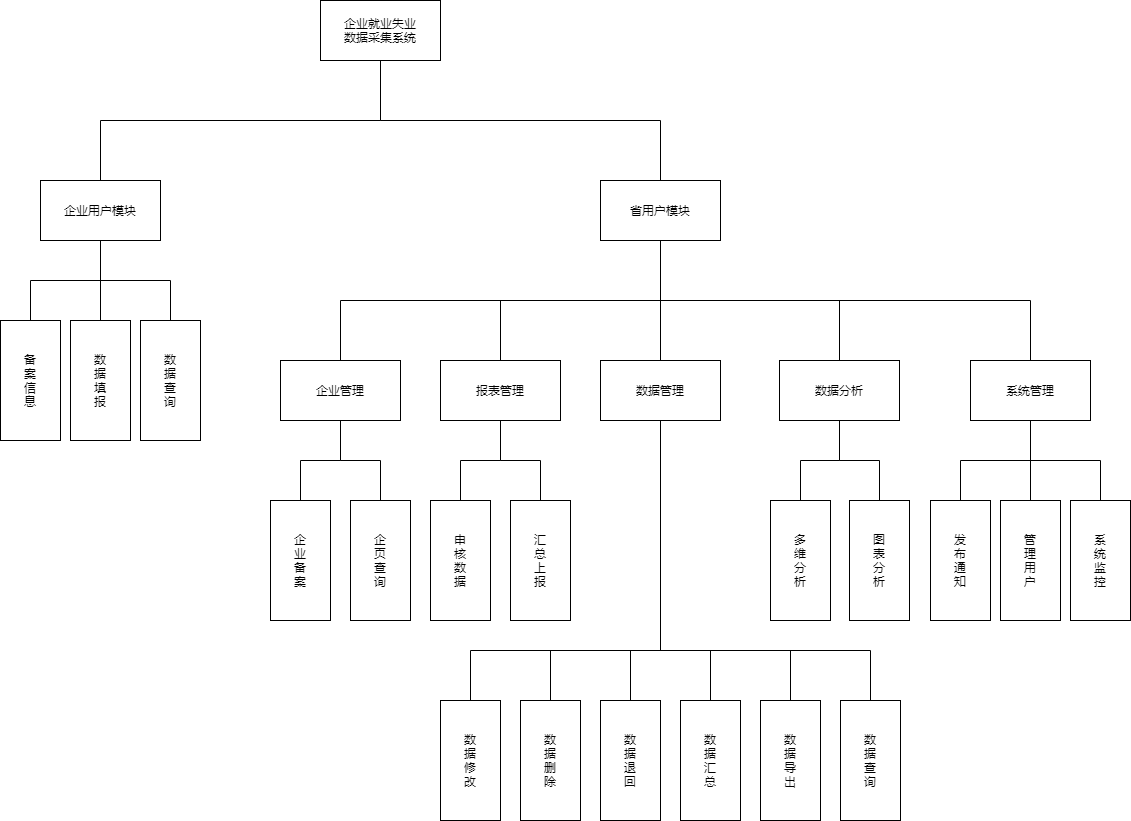


图1 整体功能预览图

## 任务分解结构（WBS）

完成项目本身是一个复杂的过程，必须采取分解的手段把主要的可交付成果分成更容易管理的单元，即任务分解结构（WBS）。WBS的建立对于项目来说意义非常重大，它使得原来看起来非常笼统、模糊的项目目标清晰起来，使得项目管理有依据、项目团队的工作目标清楚明了。山东省企业就业失业数据采集系统按照生存期阶段标准进行分解的结果如图2：

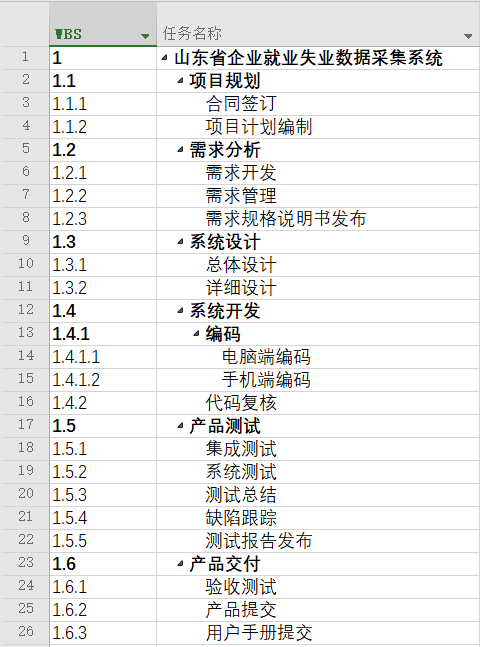


图2 任务分解结构

## 可交付成果

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 成果名称 | 作用 | 交付形式 |
| 1 | 合同 | 制定项目章程 | 文档 |
| 2 | 需求规格说明书 | 收集需求、定义范围、规划范围 | 文档 |
| 3 | 项目计划 | 指导与管理项目执行 | 文档 |
| 4 | 系统设计说明书 | 明确系统处理流程、模块、功能 | 文档 |
| 5 | 原型系统 | 最终的交付产品 | 源代码 |
| 6 | 测试报告 | 对发现的问题和缺陷进行分析 | 文档 |
| 7 | 用户手册 | 给用户准确、清晰的说明 | 文档 |

表2 可交付成果表

# 项目目标

山东省企业就业失业数据采集系统针对山东省内各大企业采集就业失业数据，利用该平台使得省级部门便于进行分析与处理数据，同时将上报、审批、通知等工作集为一体，实现管理的高效性。本系统可以达到以下目标：

* 提高工作效率
* 节省日常开支
* 业务过程的自动化
* 建立统一的数据模型
* 汇总、分析和展示业务数据
* 规范企业上报流程
* 完善部门管理体系
* 加强部门协同、信息共享
* 提高服务质量

# 项目组织结构

由于在项目实施过程中需要涉及不同组织的各方面人员，而各组织之间的任务和职责也不尽相同，因此明确定义组织结构和各自职责可保证系统开发活动的顺利进行。本项目的组织结构如图3所示：

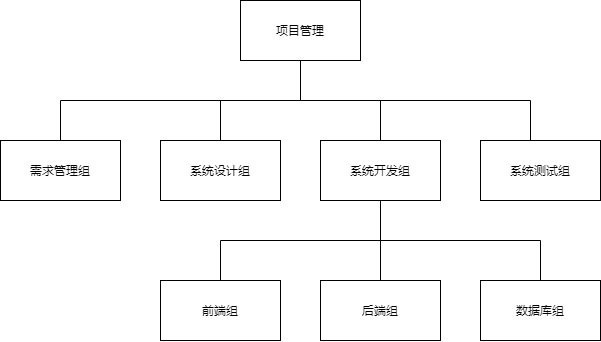


图3 组织结构图

## 各角色职责

1. **项目管理**

* 负责项目的组织和规划
* 负责项目计划制定和维护
* 负责项目的跟踪和管理
* 负责资源的分配和协调活动

1. **需求管理组**

* 负责业务流程的定义和维护
* 负责需求获取、需求分析、需求验证
* 负责项目的需求管理

1. **系统设计组**

* 负责页面结构的设计
* 负责数据库的设计
* 负责系统业务逻辑的设计

1. **系统开发组**

* 负责系统的软件开发
* 负责产品质量控制的工作
* 负责配合质量保证的活动
* 配合产品验收的相关活动

1. **系统测试组**

* 负责测试案例的设计
* 负责根据过程规范制定检查表
* 负责系统的测试

## 人力资源及角色映射

1. 项目经理：1名
2. 系统分析员：1名
3. 系统设计员：1名
4. 数据库管理员：1名
5. 程序员：4名
6. 测试人员：1名
7. 质量管理、配置管理人员：1名

**角色映射如表3所示：**

|  |  |  |
| --- | --- | --- |
| 标识号 | 人员名称 | 角色 |
| 1 | 任翔渝 | 项目经理、风险管理人员 |
| 2 | 张佳明 | 系统分析员 |
| 3 | 刘震宇 | 系统设计员 |
| 4 | 饶东来 | 数据库管理员 |
| 5 | 徐靖垚 | 程序员 |
| 6 | 冯开宇 | 程序员 |
| 7 | 李金 | 程序员 |
| 8 | 彭威 | 程序员 |
| 9 | 张鉴昊 | 测试人员 |
| 10 | 张航 | 质量管理、配置管理人员 |

表3 角色映射表

# 项目生存期

根据该项目的特点并结合已有的软件生存期模型定义，本项目生存期采用瀑布模型，如图4所示：

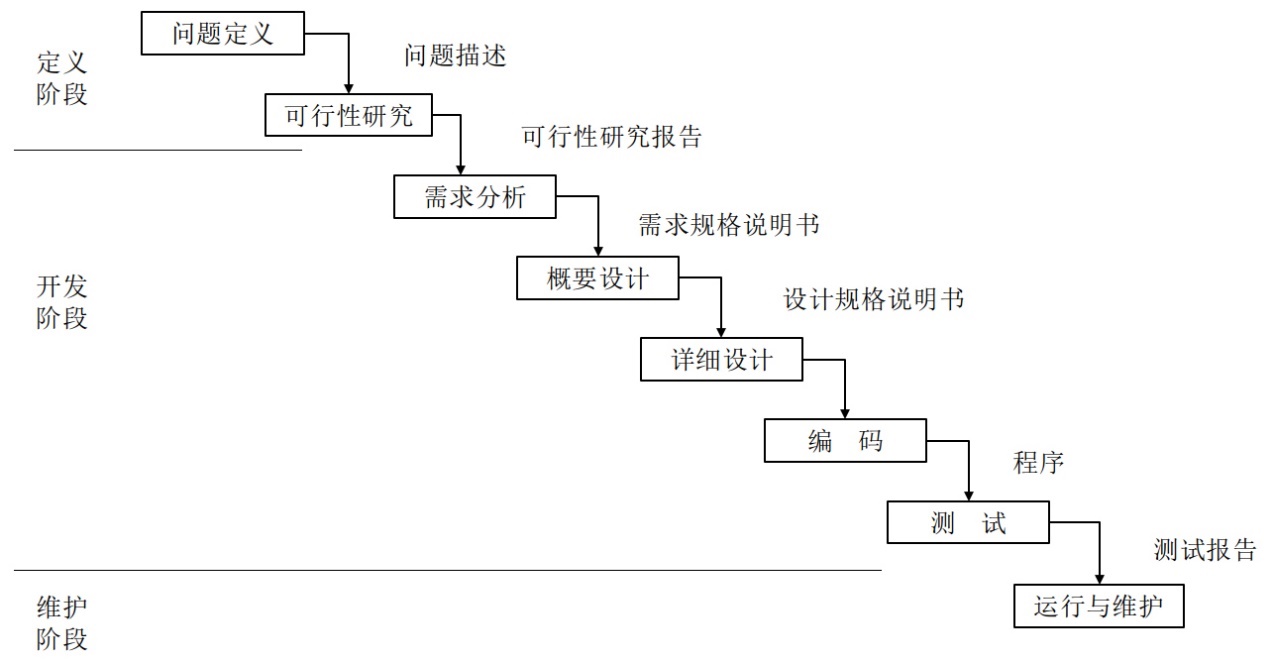


图4 瀑布模型生存期

生存期的各阶段定义如下：

1. **需求分析**

这一阶段至关重要，它包含了获取客户需求与定义的信息，以及对需要解决的问题所能达到的最清晰的描述。分析包含了理解客户的商业环境与约束，产品必需实现的功能，产品必需达到的性能水平，以及必需实现兼容的外部系统。需求分析阶段的结果是一份正式的需求规格说明书。

1. **系统设计**

这一阶段需要定义硬件和软件架构、组件、模块、界面和数据等来满足指定的需求。它包括了硬件和软件架构的定义，确定性能和安全参数，设计数据存储容器和限制，选择集成开发环境和编程语言，并指定异常处理、资源管理和界面连接性的策略。

1. **系统开发**

这一阶段需根据设计说明书来构建产品，将生成一个或多个产品组件，它们是根据每一条编码标准而编写的，并且经过了调试、测试并进行集成以满足系统架构的需求。通常开发团队包括了程序员、界面设计师、数据库架构师、服务端工程师。

1. **系统测试**

在这一阶段，独立的组件和集成后的组件都将进行系统性验证，以确保没有错误且完全符合第一阶段所制定的需求。一个独立的质量保证小组将定义“测试实例”来评估产品是完全实现了需求还是只有部分满足。有三种测试方法可以使用：对独立的代码模块进行单元测试、对集成产品进行系统测试、客户参与的验收测试。如果发现了缺陷，将会对问题进行记录并向开发团队反馈以进行修正。在这一阶段，产品文档会经过准备、评估并发布，比如用户手册等。

1. **产品提交**

最终系统软件的全部技术文档、相关技术资料、源代码和数据文件等全部技术成果全都需提交给客户。

# 项目进度计划

1. 根据用户对项目的进度要求，项目活动的起止时间如下：

开始日期：2020年3月5日

截止日期：2020年9月10日

1. 项目进度计划基本采取远粗近细的原则，粗略计划如图5所示，它展示了各阶段的时间进度：

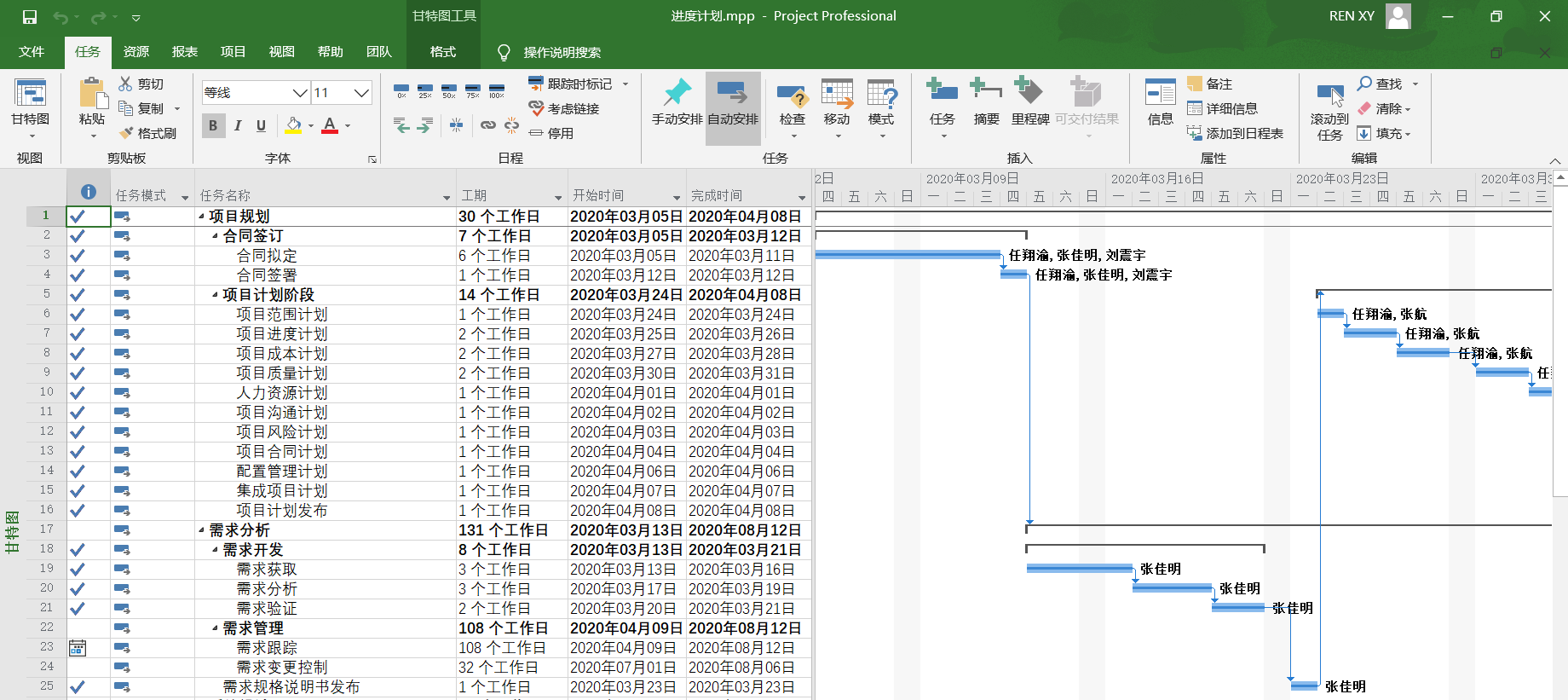


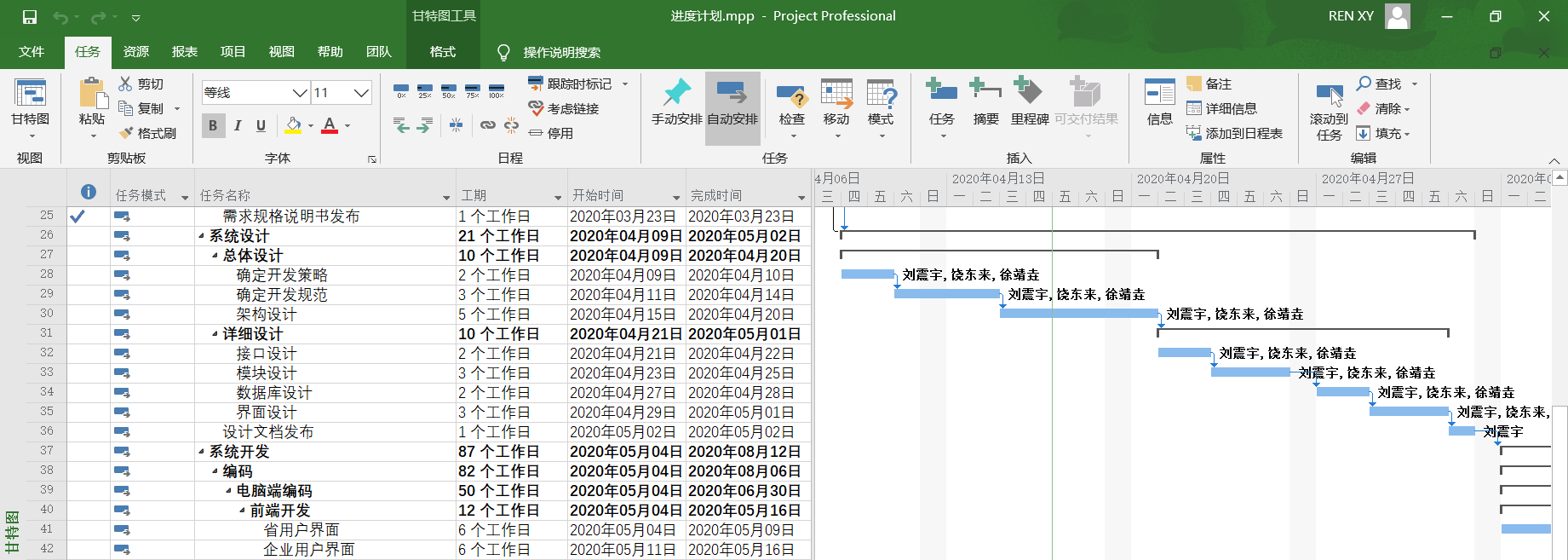
图5 粗略计划

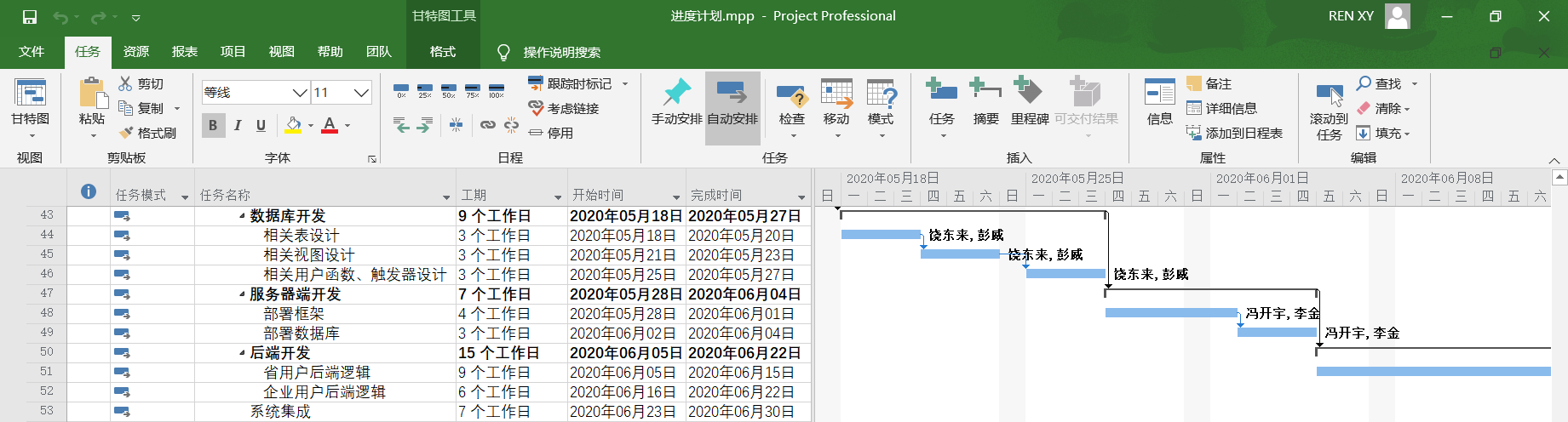
注：需求管理贯穿整个设计、开发过程

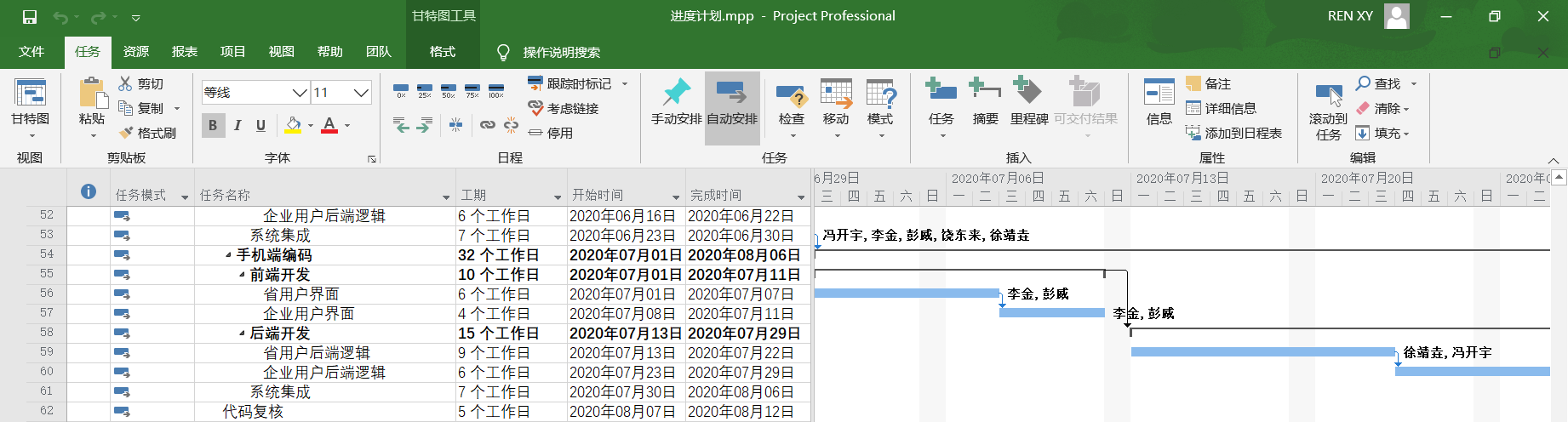
1. 项目将逐步确定详细计划，制定出项目进度计划：

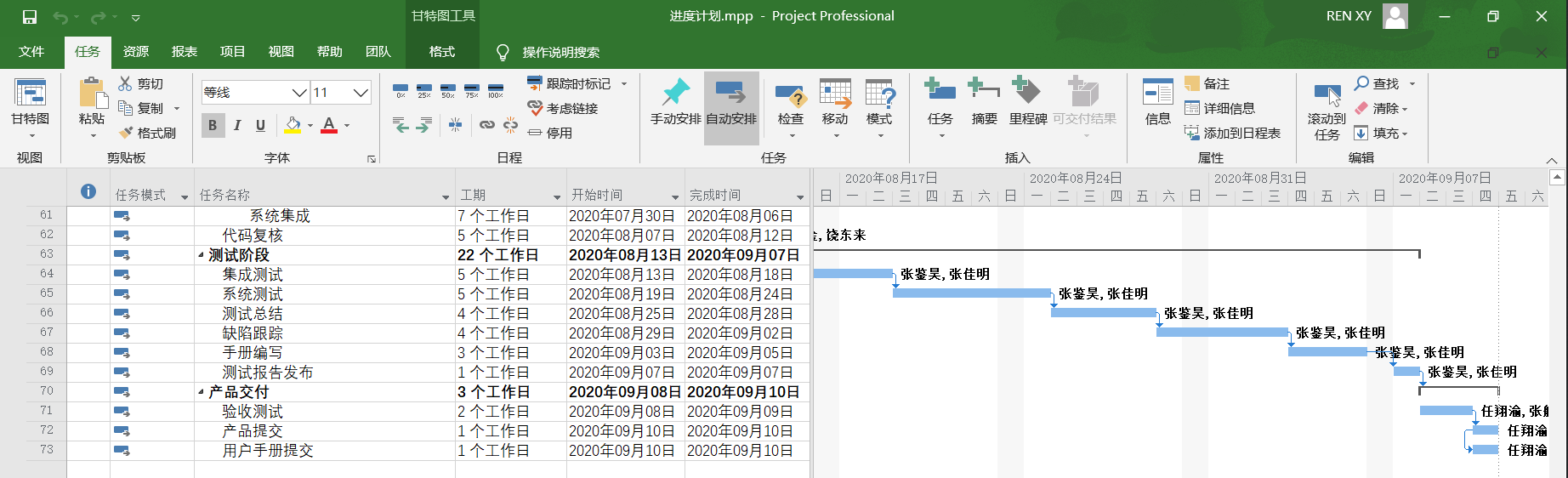
**（详情请见同文件夹中的 进度计划.mpp 文件）**











# 项目成本计划

## 成本计划概述

项目管理的目的在于使项目能够按照预定的成本、进度、质量顺利完成。与常规项目相同,软件项目开发过程中,成本、进度和质量是项目的三大关键,成本管理是保障利润的关键,但它并不是孤立的存在的,它伴随着项目的开发进度和质量同步进行。 项目成本管理包含项目成本估算、项目成本预算、项目成本控制等，项目成本管理一直是难题，成本超支是很常见的现象，因此我们需要制定项目成本计划对生产经营所消耗的人力资源、物质资源和费用开支进行指导、监督、调节和限制，及时纠正将要发生和已经发生的偏差，把各项生产费用控制在计划成本的范围之内，保证成本目标的实现。

## 项目成本估算

估算不是精确的科学计算，尤其软件项目更是如此，软件项目中存在太多的不确定性。软件成本估算是成本管理的核心，是预测开发一个软件系统所需要的总工作量的过程，而软件领域的成本计算中，主要是人的劳动的消耗。软件开发成本的估算应该以软件项目管理、需求分析、设计、编码、测试等过程所花费的代价作为依据。

**估算步骤如下：**

1. **获取项目分解结果WBS以及每个任务的规模**

任务分解是根据项目的过程进行分解的

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 标识号 | 名称 | 工期（天） | 人员数（人） | 小计（人天） | 总计（人天） |
| **1** | **项目规划** | **30** |  |  |  |
| **1.1** | **合同签订** | **7** |  |  | **21** |
| 1.1.1 | 合同拟定 | 6 | 3 | 18 |  |
| 1.1.2 | 合同签署 | 1 | 3 | 3 |  |
| **1.2** | **项目计划阶段** | **14** |  |  | **28** |
| 1.2.1 | 项目范围计划 | 1 | 2 | 2 |  |
| 1.2.2 | 项目进度计划 | 2 | 2 | 4 |  |
| 1.2.3 | 项目成本计划 | 2 | 2 | 4 |  |
| 1.2.4 | 项目质量计划 | 2 | 2 | 4 |  |
| 1.2.5 | 人力资源计划 | 1 | 2 | 2 |  |
| 1.2.6 | 项目沟通计划 | 1 | 2 | 2 |  |
| 1.2.7 | 项目风险计划 | 1 | 2 | 2 |  |
| 1.2.8 | 项目合同计划 | 1 | 2 | 2 |  |
| 1.2.9 | 配置管理计划 | 1 | 2 | 2 |  |
| 1.2.10 | 集成项目计划 | 1 | 2 | 2 |  |
| 1.2.11 | 项目计划发布 | 1 | 2 | 2 |  |
| **2** | **需求分析** | **131** |  |  | **225** |
| **2.1** | **需求开发** | **8** |  |  | 8 |
| 2.1.1 | 需求获取 | 3 | 1 | 3 |  |
| 2.1.2 | 需求分析 | 3 | 1 | 3 |  |
| 2.1.3 | 需求验证 | 2 | 1 | 2 |  |
| **2.2** | **需求管理** | **108** |  |  | 216 |
| 2.2.1 | 需求跟踪 | 108 | 2 | 216 |  |
| 2.2.2 | 需求变更控制 | 32 | 2 | 64 |  |
| 2.3 | 需求规格说明书发布 | 1 | 1 | 1 |  |
| **3** | **系统设计** | **21** |  |  | **60** |
| **3.1** | **总体设计** | **10** |  |  | 30 |
| 3.1.1 | 确定开发策略 | 2 | 3 | 6 |  |
| 3.1.2 | 确定开发规范 | 3 | 3 | 9 |  |
| 3.1.3 | 架构设计 | 5 | 3 | 15 |  |
| **3.2** | **详细设计** | **10** |  |  | 30 |
| 3.2.1 | 接口设计 | 2 | 3 | 6 |  |
| 3.2.2 | 模块设计 | 3 | 3 | 9 |  |
| 3.2.3 | 数据库设计 | 2 | 3 | 6 |  |
| 3.2.4 | 界面设计 | 3 | 3 | 9 |  |
| 3.2.5 | 设计文档发布 | 1 | 1 | 1 |  |
| **4** | **系统开发** | **87** |  |  |  |
| **4.1** | **编码** | **82** |  |  | **231** |
| **4.1.1** | **电脑端编码** | **50** |  |  | **131** |
| **4.1.1.1** | **前端开发** | **12** |  |  | 24 |
| 4.1.1.1.1 | 省用户界面 | 6 | 2 | 12 |  |
| 4.1.1.1.2 | 企业用户界面 | 6 | 2 | 12 |  |
| **4.1.1.2** | **数据库开发** | **9** |  |  | 18 |
| 4.1.1.2.1 | 相关表设计 | 3 | 2 | 6 |  |
| 4.1.1.2.2 | 相关视图设计 | 3 | 2 | 6 |  |
| 4.1.1.2.3 | 相关用户函数、触发器设计 | 3 | 2 | 6 |  |
| **4.1.1.3** | **服务器端开发** | **7** |  |  | 14 |
| 4.1.1.3.1 | 部署框架 | 4 | 2 | 8 |  |
| 4.1.1.3.2 | 部署数据库 | 3 | 2 | 6 |  |
| **4.1.1.4** | **后端开发** | **15** |  |  | 30 |
| 4.1.1.4.1 | 省用户后端逻辑 | 9 | 2 | 18 |  |
| 4.1.1.4.2 | 企业用户后端逻辑 | 6 | 2 | 12 |  |
| 4.1.1.5 | 系统集成 | 7 | 5 | 35 | 35 |
| **4.1.2** | **手机端编码** | **32** |  |  | **85** |
| **4.1.2.1** | **前端开发** | **10** |  |  | 20 |
| 4.1.2.1.1 | 省用户界面 | 6 | 2 | 12 |  |
| 4.1.2.1.1 | 企业用户界面 | 4 | 2 | 8 |  |
| **4.1.2.2** | **后端开发** | **15** |  |  | 30 |
| 4.1.2.2.1 | 省用户后端逻辑 | 9 | 2 | 18 |  |
| 4.1.2.2.2 | 企业用户后端逻辑 | 6 | 2 | 12 |  |
| 4.1.2.3 | 系统集成 | 7 | 5 | 35 | 35 |
| 4.2 | 代码复核 | 5 | 3 | 15 |  |
| **5** | **测试阶段** | **22** |  |  | **44** |
| 5.1 | 集成测试 | 5 | 2 | 10 |  |
| 5.2 | 系统测试 | 5 | 2 | 10 |  |
| 5.3 | 测试总结 | 4 | 2 | 8 |  |
| 5.4 | 缺陷跟踪 | 4 | 2 | 8 |  |
| 5.5 | 手册编写 | 3 | 2 | 6 |  |
| 5.6 | 测试报告发布 | 1 | 2 | 2 |  |
| **6** | **产品交付** | **3** |  |  | **6** |
| 6.1 | 验收测试 | 2 | 2 | 4 |  |
| 6.2 | 产品提交 | 1 | 2 | 2 |  |
| 6.3 | 用户手册提交 | 1 | 2 | 2 |  |

表4 任务规模估算

**可知项目开发规模是615人天**

1. **计算开发成本**

资源费用比例如表5所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 标识号 | 名称 | 类型 | 最大单位 | 标准费率 | 加班费率 | 成本累算 | 备注 |
| 1 | 任翔渝 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 项目经理、风险管理人员 |
| 2 | 张佳明 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 系统分析员 |
| 3 | 刘震宇 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 系统设计员 |
| 4 | 饶东来 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 数据库管理员 |
| 5 | 徐靖垚 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 程序员 |
| 6 | 冯开宇 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 程序员 |
| 7 | 李金 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 程序员 |
| 8 | 彭威 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 程序员 |
| 9 | 张鉴昊 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 测试人员 |
| 10 | 张航 | 工时 | 100% | ¥100.00/工时 | ¥150.00/工时 | 按比例 | 质量管理、配置管理人员 |

表5 资源费用比例

每天工作8小时，开发人员成本参数800元/人天，项目规模是615人天，则：

内部的开发成本是615人天×800元/人天=492000元

1. **计算管理成本**

由于任务分解的结果主要是针对开发过程的分解，没有分解出管理任务（项目管理任务和质量管理任务），针对本项目，管理成本=开发成本×10%，所以：

管理成本为492000元×10%=49200元

1. **计算直接成本**

因为直接成本=开发成本+管理成本，所以：

直接成本=492000元+49200元=541200元

1. **计算间接成本**

此处采用简易的计算方法，即间接成本=直接成本×间接成本系数，本项目中间接成本系数为20%，所以：

间接成本=541200元×20%=108240元

1. **计算总估算**

因为项目总估算成本=直接成本+间接成本，所以

项目总估算成本=541200元+108240元=649440元

1. **重新评估项目报价**

签订合同时的报价为500000元，需求变更追加了额外成本300000元，合计800000元。

假设项目的利润为30%，其中风险基金10%，利润15%，税费5%，则：

项目的总报价=649440元×130%=844272元

**由以上计算结果可知，两次的报价结果都偏低。**

## 项目成本预算

项目成本预算是将项目的总成本按照项目的进度分摊到各个工作单元中去，即将总的成本安排到各个任务中，这些任务项是基于WBS分解的结果，所以WBS、WBS字典、每个任务的成本估计、进度、资源日历等可以作为成本预算的输入。项目任务编排好了执行的先后关系并分配了资源后，项目中每个任务的成本预算就可以确定了。成本预算提供了对实际成本的一种控制机制，为项目管理者控制项目提供了一把有效的尺度。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标识号 | 名称 | 工期（天） | 开始时间 | 完成时间 | 预算（元） | 合计（元） |
| **1** | **项目规划** | **30** | **2020年3月5日** | **2020年4月8日** |  | **83664** |
| **2** | **合同签订** | **7** | **2020年3月5日** | **2020年3月12日** |  | **27888** |
| 3 | 合同拟定 | 6 | 2020年3月5日 | 2020年3月11日 | 23904 |  |
| 4 | 合同签署 | 1 | 2020年3月12日 | 2020年3月12日 | 3984 |  |
| **5** | **项目计划阶段** | **14** | **2020年3月24日** | **2020年4月8日** |  | **55776** |
| 6 | 项目范围计划 | 1 | 2020年3月24日 | 2020年3月24日 | 3984 |  |
| 7 | 项目进度计划 | 2 | 2020年3月25日 | 2020年3月26日 | 7968 |  |
| 8 | 项目成本计划 | 2 | 2020年3月27日 | 2020年3月28日 | 7968 |  |
| 9 | 项目质量计划 | 2 | 2020年3月30日 | 2020年3月31日 | 7968 |  |
| 10 | 人力资源计划 | 1 | 2020年4月1日 | 2020年4月1日 | 3984 |  |
| 11 | 项目沟通计划 | 1 | 2020年4月2日 | 2020年4月2日 | 3984 |  |
| 12 | 项目风险计划 | 1 | 2020年4月3日 | 2020年4月3日 | 3984 |  |
| 13 | 项目合同计划 | 1 | 2020年4月4日 | 2020年4月4日 | 3984 |  |
| 14 | 配置管理计划 | 1 | 2020年4月6日 | 2020年4月6日 | 3984 |  |
| 15 | 集成项目计划 | 1 | 2020年4月7日 | 2020年4月7日 | 3984 |  |
| 16 | 项目计划发布 | 1 | 2020年4月8日 | 2020年4月8日 | 3984 |  |
| **17** | **需求分析** | **131** | **2020年3月13日** | **2020年8月12日** | 521904 | **35856** |
| **18** | **需求开发** | **8** | **2020年3月13日** | **2020年3月21日** |  | **31872** |
| 19 | 需求获取 | 3 | 2020年3月13日 | 2020年3月16日 | 11952 |  |
| 20 | 需求分析 | 3 | 2020年3月17日 | 2020年3月19日 | 11952 |  |
| 21 | 需求验证 | 2 | 2020年3月20日 | 2020年3月21日 | 7968 |  |
| **22** | **需求管理** | **101** | **2020年4月17日** | **2020年8月12日** |  |  |
| 23 | 需求跟踪 | 101 | 2020年4月17日 | 2020年8月12日 |  |  |
| 24 | 需求变更控制 | 32 | 2020年7月1日 | 2020年8月6日 |  |  |
| 25 | 需求规格说明书发布 | 1 | 2020年3月23日 | 2020年3月23日 | 3984 |  |
| **26** | **系统设计** | **21** | **2020年4月9日** | **2020年5月2日** |  | **83664** |
| **27** | **总体设计** | **10** | **2020年4月9日** | **2020年4月20日** |  | **39840** |
| 28 | 确定开发策略 | 2 | 2020年4月9日 | 2020年4月10日 | 7968 |  |
| 29 | 确定开发规范 | 3 | 2020年4月11日 | 2020年4月14日 | 11952 |  |
| 30 | 架构设计 | 5 | 2020年4月15日 | 2020年4月20日 | 19920 |  |
| **31** | **详细设计** | **10** | **2020年4月21日** | **2020年5月1日** |  | **39840** |
| 32 | 接口设计 | 2 | 2020年4月21日 | 2020年4月22日 | 7968 |  |
| 33 | 模块设计 | 3 | 2020年4月23日 | 2020年4月25日 | 11952 |  |
| 34 | 数据库设计 | 2 | 2020年4月27日 | 2020年4月28日 | 7968 |  |
| 35 | 界面设计 | 3 | 2020年4月29日 | 2020年5月1日 | 11952 |  |
| 36 | 设计文档发布 | 1 | 2020年5月2日 | 2020年5月2日 | 3984 |  |
| **37** | **系统开发** | **87** | **2020年5月4日** | **2020年8月12日** |  | **346608** |
| **38** | **编码** | **82** | **2020年5月4日** | **2020年8月6日** |  | **326688** |
| **39** | **电脑端编码** | **50** | **2020年5月4日** | **2020年6月30日** |  | **199200** |
| **40** | **前端开发** | **12** | **2020年5月4日** | **2020年5月16日** |  | **47808** |
| 41 | 省用户界面 | 6 | 2020年5月4日 | 2020年5月9日 | 23904 |  |
| 42 | 企业用户界面 | 6 | 2020年5月11日 | 2020年5月16日 | 23904 |  |
| **43** | **数据库开发** | **9** | **2020年5月18日** | **2020年5月27日** |  | **35856** |
| 44 | 相关表设计 | 3 | 2020年5月18日 | 2020年5月20日 | 11952 |  |
| 45 | 相关视图设计 | 3 | 2020年5月21日 | 2020年5月23日 | 11952 |  |
| 46 | 相关用户函数、触发器设计 | 3 | 2020年5月25日 | 2020年5月27日 | 11952 |  |
| **47** | **服务器端开发** | **7** | **2020年5月28日** | **2020年6月4日** |  | **27888** |
| 48 | 部署框架 | 4 | 2020年5月28日 | 2020年6月1日 | 15936 |  |
| 49 | 部署数据库 | 3 | 2020年6月2日 | 2020年6月4日 | 11952 |  |
| **50** | **后端开发** | **15** | **2020年6月5日** | **2020年6月22日** |  | **59760** |
| 51 | 省用户后端逻辑 | 9 | 2020年6月5日 | 2020年6月15日 | 35856 |  |
| 52 | 企业用户后端逻辑 | 6 | 2020年6月16日 | 2020年6月22日 | 23904 |  |
| 53 | 系统集成 | 7 | 2020年6月23日 | 2020年6月30日 | 27888 |  |
| **54** | **手机端编码** | **32** | **2020年7月1日** | **2020年8月6日** |  | **127488** |
| **55** | **前端开发** | **10** | **2020年7月1日** | **2020年7月11日** |  | **39840** |
| 56 | 省用户界面 | 6 | 2020年7月1日 | 2020年7月7日 | 23904 |  |
| 57 | 企业用户界面 | 4 | 2020年7月8日 | 2020年7月11日 | 15936 |  |
| **58** | **后端开发** | **15** | **2020年7月13日** | **2020年7月29日** |  | **59760** |
| 59 | 省用户后端逻辑 | 9 | 2020年7月13日 | 2020年7月22日 | 35856 |  |
| 60 | 企业用户后端逻辑 | 6 | 2020年7月23日 | 2020年7月29日 | 23904 |  |
| 61 | 系统集成 | 7 | 2020年7月30日 | 2020年8月6日 | 27888 |  |
| 62 | 代码复核 | 5 | 2020年8月7日 | 2020年8月12日 | 19920 |  |
| **63** | **测试阶段** | **22** | **2020年8月13日** | **2020年9月7日** |  | **87648** |
| 64 | 集成测试 | 5 | 2020年8月13日 | 2020年8月18日 | 19920 |  |
| 65 | 系统测试 | 5 | 2020年8月19日 | 2020年8月24日 | 19920 |  |
| 66 | 测试总结 | 4 | 2020年8月25日 | 2020年8月28日 | 15936 |  |
| 67 | 缺陷跟踪 | 4 | 2020年8月29日 | 2020年9月2日 | 15936 |  |
| 68 | 手册编写 | 3 | 2020年9月3日 | 2020年9月5日 | 11952 |  |
| 69 | 测试报告发布 | 1 | 2020年9月7日 | 2020年9月7日 | 3984 |  |
| **70** | **产品交付** | **3** | **2020年9月8日** | **2020年9月10日** |  | **11952** |
| 71 | 验收测试 | 2 | 2020年9月8日 | 2020年9月9日 | 7968 |  |
| 72 | 产品提交 | 1 | 2020年9月10日 | 2020年9月10日 | 3984 |  |
| 73 | 用户手册提交 | 1 | 2020年9月10日 | 2020年9月10日 | 3984 |  |

表6 项目预算表

备注：

1. 需求跟踪包括整个设计、开发过程，需求变更控制包括整个手机端开发过程
2. 需求管理阶段不另做预算

# 质量管理计划

## 质量目标

质量管理客观地核实软件项目的实施行动与开发的产品遵从于对应的需求、过程描述、标准及规程，提前发现并派出项目中存在的问题和缺陷，保证项目的实施质量，具体目标包括：

* 通过监控软件开发过程来保证产品质量。
* 保证开发的软件和软件开发过程符合相应标准与规程。
* 保证软件产品、软件过程中存在的不合理问题得到处理，必要时将问题反映给管理者。
* 确保项目组制定的计划、标准和规程适合项目组需要，同时满足评审和审计需要。

## 质量管理职责

质量控制是项目组内每个成员的职责，涉及的主要角色包括项目经理、项目质量管理员、项目配置管理员、系统开发组各小组长、测试人员。

1. **项目经理**

负责任务的分配和监督项目进度，制定相关的工作计划和联系客户；负责审核本项目质量监控流程、质量管理办法；负责本项目质量评估问题的改善行动计划的执行，针对项目质量管理员提出的不符合问题协调项目组成员进行整改。

1. **项目质量管理员**

负责制订、检查和督促本计划的实施，及时发现项目工作中的问题，并通过评审总结报告、项目周报等形式向各项目组成员汇报质量活动的结果；按照质量评估计划，制定项目质量评估，登记质量问题表，指导和监控行动计划的有效执行。

1. **项目配置管理员**

负责有关软件配置项及项目各生命周期交付文档管理和变更控制工作。

1. **系统开发组各小组长**

负责在每个软件开发生命周期阶段结束后，总结本模块的软件质量状况和质量目标的实现情况,以确保整个项目目标的实现。对质量目标应定期进行考核，以追求质量管理体系的持续改进。

1. **测试人员**

负责对软件的质量和对需求实现的程度进行把关，并定期整理测试情况分析报告交项目经理、系统分析人员。

## 质量管理流程

在项目实施过程中，每个项目成员都要对自己工作成果的质量负责，并且每个项目成员都是质量管理过程的参与者。质量管理流程如下：

1. **质量计划**

软件质量计划过程是确定项目应达到的质量标准，以及决定如何满足质量标准的计划安排和方法。合适的质量标准是质量计划的关键。质量计划主要是依据项目的质量方针、产品描述以及质量标准和规则等制定出来实施策略，其内容全面反映用户的要求，为质量小组成员有效工作提供指南，为项目小组成员以及项目相关人员了解在项目进行中如何实施质量保证和控制过程提供依据，为确保项目质量得到保障提供坚实的基础。

1. **质量保证**

质量保证是为了提供信用，证明项目将会达到有关质量标准而开展的有计划、有组织的工作活动。它贯穿整个项目生命周期的系统性活动，经常性地针对整个项目质量计划的执行情况进行评估、检查与改进等工作，向管理者、顾客或其他地方提供新人，确保项目质量与计划保持一致。

1. **质量控制**

质量控制是确定项目结果与质量标准是否相符，同时确定不符的原因和消除方法，控制产品的质量，及时纠正缺陷的过程。质量控制对阶段性的成果进行检测、验证，为质量保证提供参考依据。软件质量控制主要是发现和消除软件产品的缺陷，对于高质量的软件来讲，最终产品应该尽可能达到零缺陷。

## 质量活动

1. **问题级别划分**
2. **需求分析阶段**



1. **系统设计阶段**



1. **系统开发阶段**



1. **系统测试阶段**



1. **各阶段质量目标和检查标准**
2. **需求分析阶段**



1. **系统设计阶段**



1. **系统开发阶段**



1. **系统测试阶段**



# 配置管理计划

## 配置管理流程

本项目配置管理首先进行配置管理计划，并据此搭建配置计划中所需要的配置环境，同时确定配置管理活动，包括配置项标识，建立基线，编写配置状态报告，执行配置审计，确定变更控制管理。流程如下图所示：

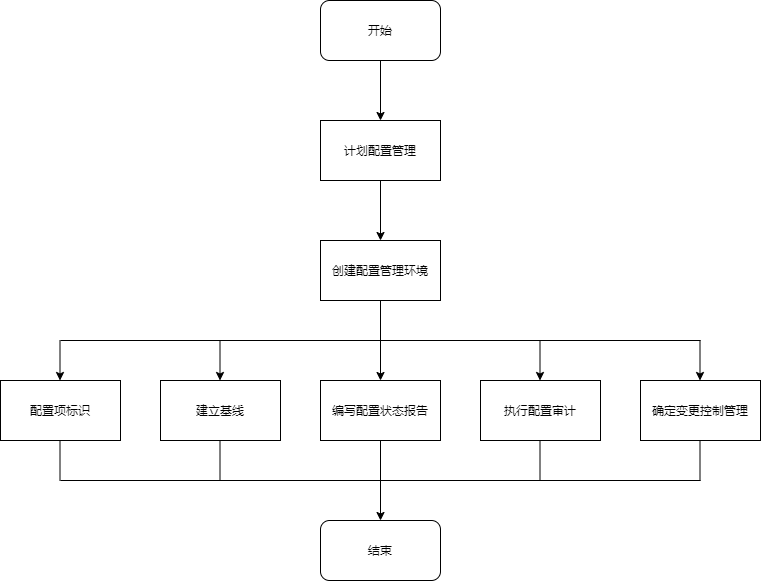


图6 配置管理流程图

## 配置项标识

本项目的配置项便是即文件名规则，包括5个部分，其中第一部分代表公司，第二部分代表项目名，第三部分代表项目阶段，第四部分代表文档类型，第五部分代表版本号。

**公司：3个字符**

**项目：最长10个字符**

**类型：最长5个字符**

**编号：最长8位数字**

**版本号：V m.n**

|  |
| --- |
| **MY--SD–RM–SRS-v1.0** |

图7 配置项标识

## 配置库建立

建库程序如下：

1. 确定纳入配置管理的工作产品和不纳入配置管理的工作产品
2. 确定基线产品和非基线产品的命名规则。
3. 采用配置管理工具建立软件配置库
4. 根据项目管理者的要求，对可以操作配置库的项目人员进行授权，包括读、写等权限，如下表所示：

|  |  |  |
| --- | --- | --- |
| 类别 | 人员 | 权限说明 |
| 配置管理者 | 张航 | 负责项目配置管理，对库拥有所有权限 |
| 项目经理 | 任翔渝 | 访问、读 |
| 质量保证人员 | 张航 | 访问、读 |
| 开发人员 | 饶东来、徐靖垚、冯开宇、李金、彭威 | 访问、读 |
| 测试人员 | 张鉴昊 | 访问、读 |

表7 配置库的用户权限

1. 确定项目在配置库中的项目名，最好与项目标识一致。
2. 此项目的配置管理者获得此项目名的最高权限。

## 入库程序

入库程序包括基线产品入库程序和非基线产品入库程序。

1. **基线产品入库程序**
2. 配置管理者将此配置项导入配置库中对应项目的相应目录中，并进行版本标识，在描述栏给出一定的描述
3. 确定与此配置项关联的其他已知的产品，并在基线状态表中增加此配置项的关联项，同时标识覆盖关系。
4. 确定此配置项相关联的其他已知的基线的配置项，并在基线状态表中修改与此配置相关联的其他配置项的关联项，同时标识覆盖关系。
5. 生成基线状态记录表，并将基线状态记录表导入配置库中。
6. **非基线产品入库程序**
7. 配置管理者将此产品导入配置库中对应项目的项目目录中，并进行版本标识，在描述栏给出一定的描述
8. 确定与此非基线产品关联的其他已知的基线产品的配置项，在基线关系表中修改这些基线的配置项的关联记录，同时标识覆盖关系。
9. 生成基线状态记录表，并将基线状态记录表导入配置库。

## 出库程序

出库程序如下：

1. 有权限读取此配置库的项目人员可以根据需求从配置库中以可读的方式导出相应的工作产品。
2. 当某工作产品变更时，配置管理者以可写的方式将此产品从配置库导出，期间此工作产品不能以可写的方式出库，只能以可读的方式出库。

## 基线变更程序

基线变更程序如下图所示，当有新的需求变更产生时，对其进行需求变更的审核，如果审核通过之后则在下一个工作阶段将其实现，如果没有通过则拒绝变更。

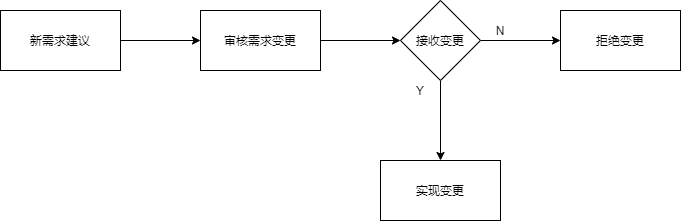


图8 变更流程

# 项目风险计划

项目风险指在项目建设过程中出现某种影响项目正常开展或对项目造成直接损失的可能性，常见的风险有进度风险、质量风险、技术风险、人员风险和成本风险等。这里需要注意的是，如果项目中存在某方面的风险，并不意味着在项目建设过程中就一定会出现该方面的问题，而只是说有一定的概率会出现这方面的问题，其中风险越高，问题发生的概率也越高。因此，项目经理进行项目风险管理工作的主要目标就是通过相应的管理手段，来控制项目中风险发生的可能性、降低其对项目工作的影响。

## 风险识别

根据对项目需求的了解和用户对实现时间及技术的要求，本项目在实施过程中可能存在如下风险：

1. **需求风险**

系统所有者对系统实现的需求也是随着项目的进展而不断具体化，而每一次需求的变化都可能由于影响设计和开发而造成时间和资源的调整，这给项目实施带来一定的需求风险。

1. **时间风险**

该系统的开发工作量较大且时间有限，这给项目实施带来了一定的时间风险。

1. **资源风险**

由于目前可以投入的开发人员悠闲，而新员工又面临磨合和培训的过程，因此项目在实施中可能存在一定的资源风险。

1. **质量风险**

在系统开发过程中可能出现设计缺陷、开发人员对系统理解出现偏差、开发人员编码能力差、测试不充分等问题，这会使项目的实施出现质量风险。

1. **人员风险**

一旦项目组织和人员发生变动，往往会关系到整个项目的成败，如人员因特殊原因而不能参加工作，人员之间产生冲突，人员丧失工作积极性等，这些因素是不可预估却可能发生的。

## 风险评估

风险评估指对在风险识别过程中所发现的客观存在或潜在的风险，从其发生的可能性和损失程度等方面来分析其对项目建设工作造成的影响。本项目的风险评估表如下（可能性、严重性、不可控性、风险等级的风险指数分布为1-10，风险指数越大代表风险系数越高）：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **项目管理过程** | **风险识别** | | **风险评估** | | | |
| **潜在的风险事件** | **风险发生的后果** | **可能性** | **严重性** | **不可控性** | **风险等级** |
| **需求**  **分析** | 需求不明确 | 客户不接受产品或拒绝付款 | 5 | 8 | 6 | 5 |
| 项目范围定义不明确 | 项目没完没了 | 8 | 8 | 5 | 6 |
| 与客户沟通不够 | 软件不能满足客户需求 | 5 | 8 | 6 | 5 |
| 需求小组没有真正理解客户需求 | 软件不能跟踪客户需求 | 8 | 9 | 7 | 9 |
| 需求不断变化 | 项目变得没完没了 | 8 | 8 | 5 | 6 |
| 缺乏有效的需求变化管理过程 | 项目不能按时、按预算完成 | 5 | 7 | 4 | 2 |
| **系统设计** | 缺乏有经验的分析员 | 分析错误或不可行 | 4 | 9 | 5 | 3 |
| 设计偏离客户需求 | 软件不能萍踪需求，客户拒绝接受 | 5 | 9 | 5 | 6 |
| 软件功能漏项 | 客户不满意 | 4 | 7 | 5 | 2 |
| **系统**  **开发** | 程序员对系统设计的理解上出现偏差 | 软件实现不了设计的功能，客户拒绝接受 | 6 | 8 | 4 | 6 |
| 程序员开发能力差 | （5）法律、行政法规和国务院规定的其他建设项目。项目进度拖期 | 规划环境影响的跟踪评价应当包括下列内容：4 | （一）建设项目环境影响评价的分类管理8 | 5 | 5 |
| **（4）建设项目环境保护措施及其技术、经济论证。**程序员不熟悉开发工具 | （三）环境标准和环境影响评价技术导则项目进度拖期、质量问题 | 环境影响评价工程师课主持进行下列工作：3 | 8 | 4 | 3.完整性原则；3 |
| 设计错误导致编码实现困难 | （二）规划环境影响评价的技术依据和基本内容质量问题 | 4 | 9 | 5 | 6 |
| 客户要求增加功能 | 项目进度拖期、成本超支 | 8 | 6 | 5 | 5 |
| 开发团队内部沟通不够 | 接口混乱、质量问题 | 5 | 7 | 4 | 4 |
| **系统**  **测试** | 没有切实可行的测试计划 | 项目拖期、质量问题发现不了 | 2 | 8 | 5 | 5 |
| 测试人员不能按时到位 | 项目进度拖期 | 2 | 6 | 3 | 4 |
| 测试人员经验不够 | 程序问题发现不了 | 4 | 5 | 3 | 5 |
| 没有有效的备份方案 | 数据丢失无法挽救 | 4 | 8 | 4 | 3 |
| 测试发现的问题迟迟解决不了 | 项目进度拖期 | 3 | 8 | 5 | 7 |
| **产品**  **交付** | 质量问题多 | 客户投诉 | 6 | 7 | 4 | 6 |
| 重要的记录、文件、数据丢失 | 客户投诉、要求赔偿 | 3 | 8 | 5 | 7 |
| 系统故障 | 客户要求承担损失 | 2 | 9 | 3 | 6 |
| 用户手册错误多 | 客户投诉 | 3 | 5 | 4 | 3 |

表8 风险评估表

## 风险应对策略

在完成项目风险的识别和评估工作后，项目经理应基于相应的分析、评估结果，结合项目的实际情况，对所发现的风险给出适当的应对措施，并督促、监督相关人员及时的对风险进行处理，以降低这些风险发生的概率、减少其对项目建设工作造成的损失。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目管理过程** | **风险识别** | | **风险应对措施** | |
| **潜在的风险事件** | **风险发生的后果** | **应对措施** | **预防措施** |
| **需求分析** | 需求不明确 | 客户不接受产品或拒绝付款 | 派遣经验丰富的需求分析师与客户进行深入的交流，明确客户的主要需求，引导客户对项目做出正确的描述。 | 事先进行需求评审 |
| 项目范围定义不明确 | 项目没完没了 | 要求需求小组按照客户的要求变更项目范围。 | 需求要在事先定义清楚并获得客户的确认。 |
| 与客户沟通不够 | 软件不能满足客户需求 | 立即与客户进行沟通 | 制定沟通管理计划 |
| 需求小组没有真正理解客户需求 | 软件不能萍踪客户需求 | 根据客户需求修改 | 让客户确认需求报告 |
| 需求不断变化 | 项目变得没完没了 | 提交CCB讨论、决定 | 建立范围变更程序 |
| 缺乏有效的需求变化管理过程 | 项目不能按时、按预算完成 | 对需求变化进行评审 | 建立需求变更程序 |
| **系统设计** | 缺乏有经验的分析员 | 分析错误或不可行 | 培训或换人 | 配备有经验的分析员 |
| 设计偏离客户需求 | 软件不能萍踪需求，客户拒绝接受 | 修改设计 | 进行设计评审 |
| 软件功能漏项 | 客户不满意 | 增加相应的功能 | 进行设计评审、获得客户确认 |
| **系统开发** | 程序员对系统设计的理解上出现偏差 | 软件实现不了设计的功能，客户拒绝接受 | 修改代码 | 进行设计评审 |
| 程序员开发能力差 | （5）法律、行政法规和国务院规定的其他建设项目。项目进度拖期 | 培训或换人 | 为了有别于传统的忽视环境价值的理论和方法，环境经济学家把环境的价值称为总经济价值（TEV），包括环境的使用价值和非使用价值两个部分。配备精兵强将 |
| **（4）建设项目环境保护措施及其技术、经济论证。**程序员不熟悉开发工具 | （三）环境标准和环境影响评价技术导则项目进度拖期、质量问题 | 立即改进 | 提前准备 |
| 设计错误导致编码实现困难 | （二）规划环境影响评价的技术依据和基本内容质量问题 | 1.规划环境影响评价的技术依据修改设计 | 编码之前进行设计评审 |
| 客户要求增加功能 | 项目进度拖期、成本超支 | 修改程序 | 事先确定范围目标 |
| 开发团队内部沟通不够 | 接口混乱、质量问题 | 修改程序 | 制定内部沟通计划 |
| **系统测试** | 没有切实可行的测试计划 | 项目拖期、质量问题发现不了 | 修改测试计划 | 事先评审测试计划 |
| 测试人员不能按时到位 | 项目进度拖期 | 临时安排测试人员 | 制定出人力资源计划 |
| 测试人员经验不够 | 程序问题发现不了 | 培训或换人 | 选择有经验的人员 |
| 没有有效的备份方案 | 数据丢失无法挽救 | 重新开始 | 异地双重备份 |
| 测试发现的问题迟迟解决不了 | 项目进度拖期 | 加快解决 | 专家会诊解决 |
| **产品提交** | 质量问题多 | 客户投诉 | 即时解决问题 | 事先进行局部运行 |
| 重要的记录、文件、数据丢失 | 客户投诉、要求赔偿 | 重新生成数据 | 做好备份 |
| 系统崩溃 | 客户要求承担损失 | 加紧修复 | 事先备份 |
| 用户手册错误多 | 客户投诉 | 修改错误 | 专人检查 |

表9 风险应对策略表

# 项目沟通计划

项目沟通计划分为外部协调和内部沟通两部分。

## 外部协调

对于外部协调，应注意以下两点：

1. 原则上由合同管理者负责与客户进行协调。为减少交流成本，项目人员也可直接与用户联系，但必须讲联系内容通报合同管理者和项目助理，并由项目助理记录沟通记录。
2. 建立每周定期报告制度，由项目管理者向客户进行工作汇报，报告内容包括项目进展状态、下步安排、项目管理问题协商等。联系方式为E-mail，突发事件可通过电话联系。E-mail地址格式如下：

我方：[rxy@xxx.com](mailto:rxy@xxx.com)

客户：customer@xxx.com

E-mail标识：WeeklyReport-mmdd，其中mmdd表示月日，使用两位数字表示，如0305表示3月5日。

## 内部沟通

在项目开发中，要进行频繁沟通，主要的3个沟通会议是每日站立会议（一般15分钟）、项目规划会议、项目复审会议。

1. **每日站立会议**
2. **会议时间**

* 每天晚上8点钟。

1. **会议目的：**

* 协调每日任务，讨论遇到的问题。
* 任务板能够帮助团队聚焦于每日活动之上，要在这个时候更新任务板。

1. **基本要求：**

* 项目团队所有人员参加。
* 每天15分钟，同样时间，同样地点。
* 团队成员都应该认真聆听他人发言
* 项目经理在会议时不要营造类似于师生教学的气氛。

1. **会议输出：**

* 团队彼此明确知道各自的工作、最新的工作进度图。
* 得到最新的“本迭代产品状况”

1. **项目阶段会议**
2. **会议时间**

在每个阶段第一天召开。

1. **会议目的**

估算本阶段的工作项，明确优先级排序，确定本阶段可提交结果，给出设计方案，估算本阶段的工作量。

1. **会议内容：**

* 该会议的工作以分析为主，目的是详细理解最终用户到底要什么，产品开发团队可以从该会议中详细了解最终用户的真实需要，决定他们能够交付那些东西。
* 开发团队可以为他们要实现的解决方案完成计划工作，团队要知道如何有效开展当前阶段的重要工作。
* 估算本阶段的工作任务列表。

1. **会以输出：**

* 阶段工作计划。
* 经过估算和排序的工作任务列表。

1. **项目复审会议**
2. **会议时间**

* 项目阶段结束。

1. **会议目的**

* 向最终用户展示工作成果，得到用户的反馈，并据此创建或变更列表条目。

1. **会议内容**

* 向参与者展示团队的最新进度
* 有可能发布的产品增量，由团队展示。

1. **会议输出**

* 用户的反馈。
* 更新的产品任务列表。

## 沟通方式说明

为确保项目管理的有效进行，建立沟通事件记录通报制度，事件包括与用户的电话记录、各方建议等。

1. **邮件沟通**

邮件沟通在项目实施过程中是使用最频繁的沟通方式，邮件沟通约定如下：

1. 邮件收件人为对邮件内容必须知晓或邮件必须反馈的人员。
2. 邮件抄送人为对邮件内容了解或对邮件可以但不强制反馈的人员。
3. 邮件主题“【”+ 组织结构名称 + “-”+邮件主题目“】”+邮件子题目。
4. 邮件正文分为几种类型，邮件正文约定如下：

称谓，大家好

主要内容要清晰无歧义

落款 日期

联系方式

1. **电话沟通**

电话沟通是要清晰无歧义。电话沟通的结果（如需要）可以以邮件方式记录后发给相关人员。

1. **文件沟通与口头沟通**

文件沟通特指通过纸质文件进行沟通的方式，在满足纸质文件流转规定的同时尽快推进。口头沟通时，遇到争议暂无法解决的问题，先记录下来之后讨论。口头沟通的结果（如需要）可以以邮件方式记录后发给相关人员。