在线聊天室项目文档

项目计划文档

李梓瑞 学号SA22225007

2022

目录

[一 WBS工作分解 1](#_Toc119244577)

[1.1 分解方式 1](#_Toc119244578)

[1.2 WBS分解图 1](#_Toc119244579)

[1.3 WBS分解列表 1](#_Toc119244580)

[1000 设计文档 1](#_Toc119244581)

[2000 在线聊天室系统前端 1](#_Toc119244582)

[3000 在线聊天室系统后端 3](#_Toc119244583)

[4000 测试文档 3](#_Toc119244584)

[二 关键路径图 4](#_Toc119244585)

[2.1 规划方式 4](#_Toc119244586)

[2.2 关键路径图 4](#_Toc119244587)

[三 工作量估算 7](#_Toc119244588)

[3.1 估算方式 7](#_Toc119244589)

[3.2 估算过程以及结果 7](#_Toc119244590)

[3.2.1系统对象列表 7](#_Toc119244591)

[3.2.2 ILF和EIF分析 7](#_Toc119244592)

[3.2.3 EI/EO/EQ分析 8](#_Toc119244593)

[3.2.4调整因子计算 8](#_Toc119244594)

[3.2.5最终工作量估算 9](#_Toc119244595)

[四 风险管理计划 10](#_Toc119244596)

[4.1 风险分析 10](#_Toc119244597)

[4.1.1 风险识别 10](#_Toc119244598)

[4.1.2 风险预测 11](#_Toc119244599)

[4.3 风险评估 12](#_Toc119244600)

[4.2 风险应对 13](#_Toc119244601)

[五 质量管理计划 14](#_Toc119244602)

[5.1 质量管理 14](#_Toc119244603)

[5.1.1 细化设计 14](#_Toc119244604)

[5.1.2 代码开发 14](#_Toc119244605)

[5.1.3 测试 15](#_Toc119244606)

[5.2 质量保证 15](#_Toc119244607)

[5.2.1文档 15](#_Toc119244608)

[5.2.2 评审活动 16](#_Toc119244609)

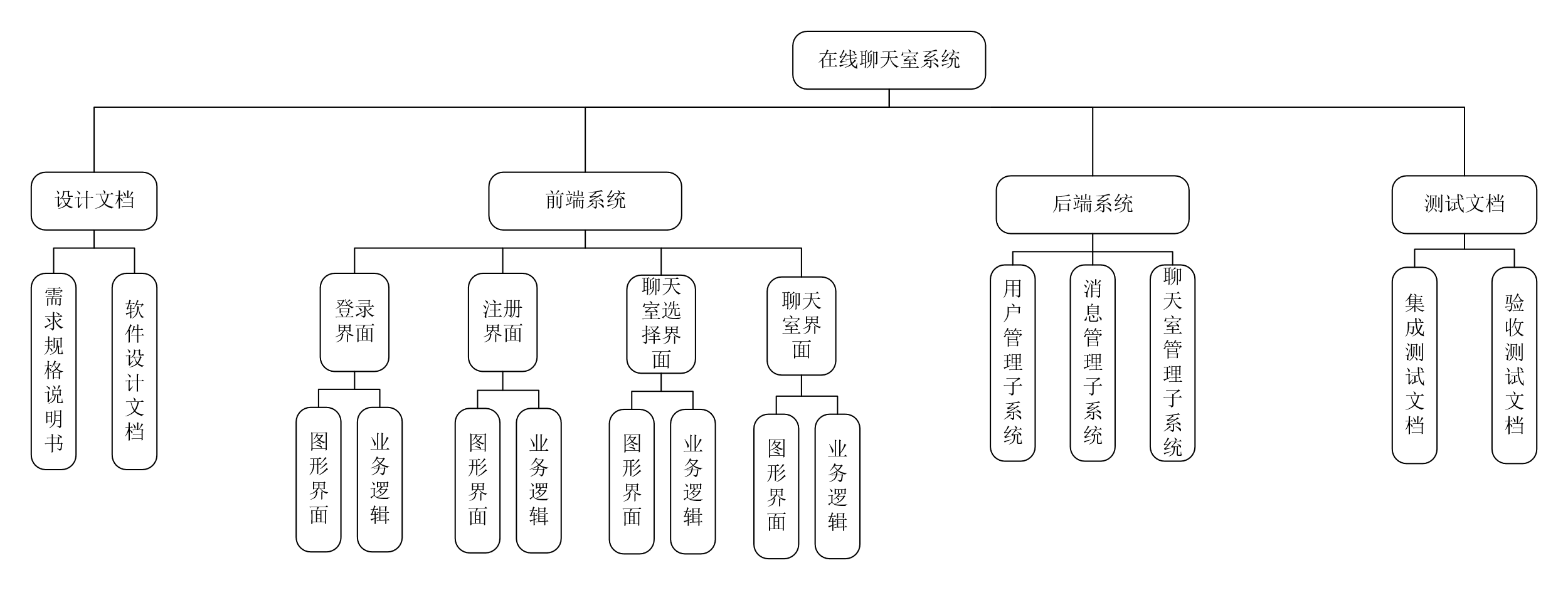
[5.2.3 软件配置管理 16](#_Toc119244610)

# 一 WBS工作分解

## 1.1 分解方式

本次工作分解计划采取基于成果的分解方式，将系统按照其必须的子工作成果划分成子系统，逐步分解直至最终的工作粒度为4至10小时的工作任务。

## 1.2 WBS分解图



## 1.3 WBS分解列表

对1.2中的各工作项进行编号并简述任务内容

### 1000 设计文档

任务描述：对软件的需求进行分析，并进行设计工作

#### 1100 需求规格说明书

任务描述：进行需求分析，得到需求规格说明书

#### 1200 软件设计文档

任务描述：对软件进行初步设计，得到初步UML建模文档

### 2000 在线聊天室系统前端

任务描述：本次项目最终目标的前端页面，包含支持用户注册，登录，发送以及实时交流多个页面的在线聊天系统。

#### 2100 登录界面

任务描述：用户进入系统的入口，通过输入用户名以及密码，经过系统的验证，用户才能进入聊天室进行各种操作。

##### 2110 用户界面

任务描述：设计登录界面的网页图形界面。

##### 2120 业务逻辑

任务描述：完成登录界面背后的程序逻辑，与后台交互。

#### 2200 注册界面

任务描述：未拥有账号的用户进行注册的界面，用户可以选择头像，输入自己希望的用户名以及密码，经服务器确认后获得进入聊天室的账号。

##### 2210 用户界面

任务描述：设计注册界面的网页图形界面。

##### 2220 业务逻辑

任务描述：完成注册界面背后的程序逻辑，与后台交互。

#### 2300 聊天室界面

任务描述：聊天室的主要界面，用户将在该界面与其他用户进行交流。

##### 2310 用户界面

任务描述：设计聊天室界面的网页图形界面。

##### 2320 业务逻辑

任务描述：完成聊天室界面背后的程序逻辑，与后台交互。

#### 2400 聊天室选择界面

任务描述：一个用户可能加入多个聊天室，该界面提供了用户加入不同的聊天室的入口

##### 2410 用户界面

任务描述：设计聊天室选择界面的网页图形界面

##### 2420 业务逻辑

任务描述：完成聊天室选择界面背后的程序逻辑，与后台交互。

### 3000 在线聊天室系统后端

任务描述：本次项目最终目标的后台系统，支持处理前端系统发送的各种请求并返回合适的应答。

#### 3100 用户管理子系统

任务描述：用于支持各种用户相关数据操作的子系统

#### 3200 消息管理子系统

任务描述：用于支持各种消息相关数据操作的子系统

#### 3300 聊天室管理子系统

任务描述：用于支持各种聊天室相关数据操作的子系统

### 4000 测试文档

任务描述：对整个系统进行集成测试和验收测试并得到测试文档

#### 4100 集成测试文档

任务描述：进行集成测试，得到集成测试文档

#### 4200 验收测试文档

任务描述：进行验收测试，需求确认，得到验收测试文档

# 二 关键路径图

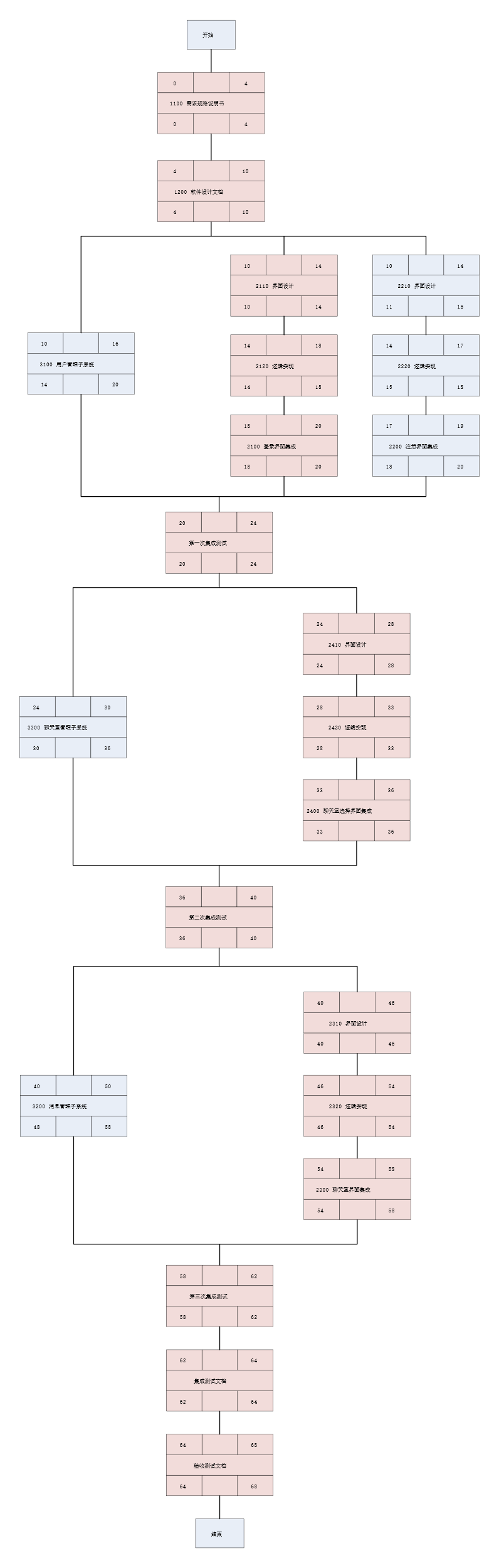
## 2.1 规划方式

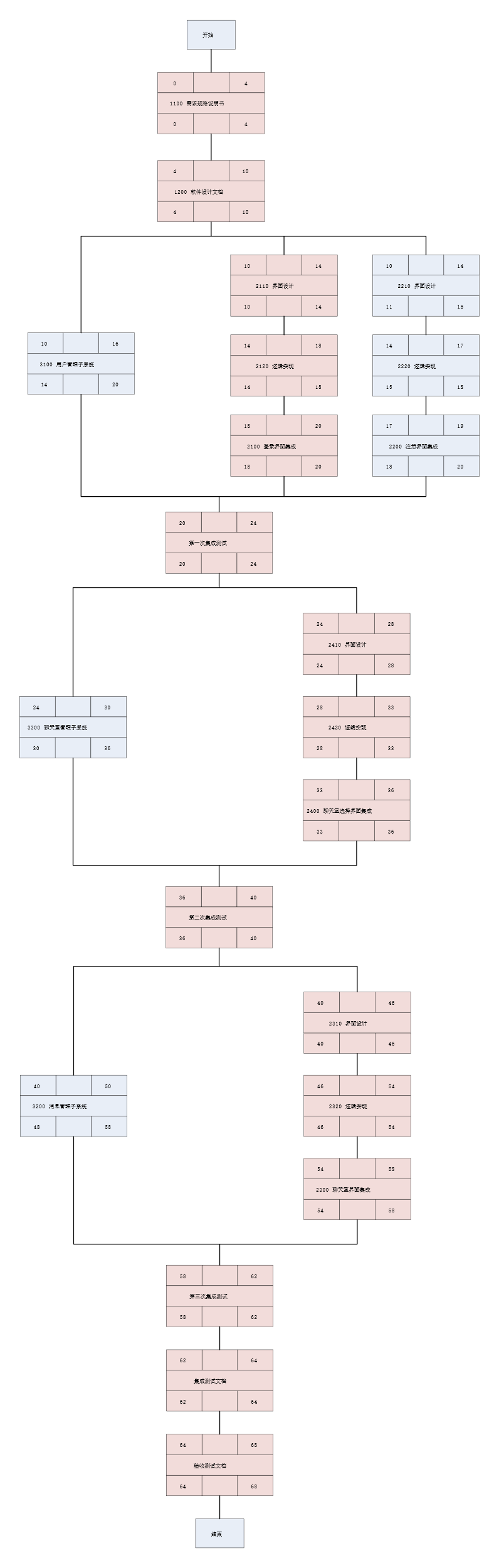
依托WBS的分解成果，在其之上添加部分里程碑，随后对每一个工作项的耗时进行估算，先采用正推法得到活动的最早开始和结束时间，再用逆推法得到活动的最晚开始和结束时间，逐步构建整个项目的关键路径图。

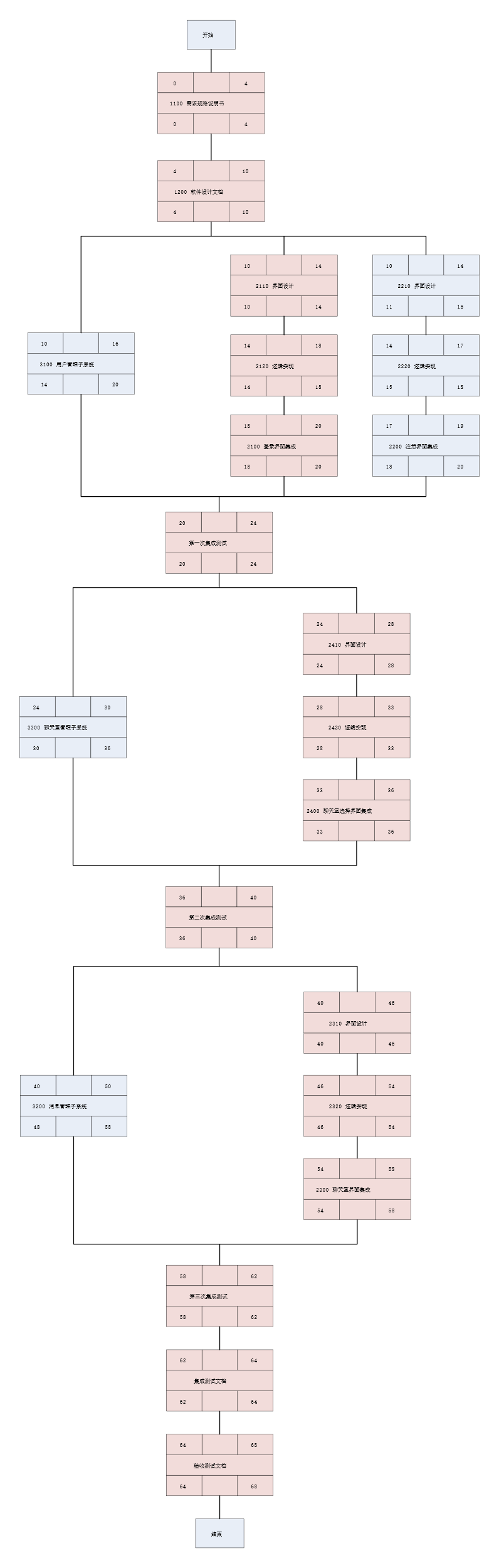
## 2.2 关键路径图

活动左上方框表示活动的最早开始时间，右上方框表示活动的最早结束时间，左下方框表示活动的最晚开始时间，右下方框表示活动的最晚结束时间，单位：小时（实际工时）

图中的关键路径使用淡红色背景进行了标注：（图片较长进行了裁剪，完整图片随附件）







# 三 工作量估算

## 3.1 估算方式

本次项目计划采用了基于功能点（FP）的规模估算方法，按照IFPUG的标准执行

## 3.2 估算过程以及结果

### 3.2.1系统对象列表

用户：

用户ID（标签）

用户名

密码

头像

消息：

消息ID（标签）

消息发送者

消息发送时间

消息所属聊天室

消息类型

消息内容

聊天室：

聊天室ID（标签）

聊天室编号

聊天室名

聊天室备注

### 3.2.2 ILF和EIF分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ILF内部逻辑文件** | **RET** | **DET** | **复杂度** | **未调整FP个数** |
| 用户信息 | 用户基本信息 | 3 | 低 | 7 |
| 消息信息 | 消息基本信息，聊天室基本信息，用户基本信息 | 8 | 低 | 7 |
| 聊天室信息 | 聊天室基本信息 | 3 | 低 | 7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EIF内部逻辑文件** | **RET** | **DET** | **复杂度** | **未调整FP个数** |
| 无 | 无 | 0 | 无 | 0 |

|  |  |
| --- | --- |
| 总计FP数 | **21** |

### 3.2.3 EI/EO/EQ分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EI** | **FTR** | **DET** | **复杂度** | **未调整FP个数** |
| 添加用户 | 用户信息 | 3 | 低 | 3 |
| 修改用户信息 | 用户信息 | 3 | 低 | 3 |
| 消息发送 | 消息信息，用户信息，聊天室信息 | 5 | 高 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EO** | **FTR** | **DET** | **复杂度** | **未调整FP个数** |
| 无 | 无 | 0 | 无 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EQ** | **FTR** | **DET** | **复杂度** | **未调整FP个数** |
| 查询用户信息 | 用户信息s | 3 | 低 | 3 |
| 查询消息 | 消息信息，用户信息，聊天室 | 5 | 低 | 3 |
| 查询聊天室 | 聊天室信息 | 3 | 低 | 3 |

|  |  |
| --- | --- |
| 总计FP数 | **15** |

### 3.2.4调整因子计算

本系统各个系统特性的需求度为：

|  |  |  |
| --- | --- | --- |
| **序号** | **模块名称** | **最终结果** |
| 1 | 系统需要可靠的备份和恢复吗？ | 0 |
| 2 | 需要数据通信吗？ | 3 |
| 3 | 有分布处理的功能吗？ | 0 |
| 4 | 性能是否关键？ | 1 |
| 5 | 系统是否在一个已有的实用的操作环境中运行 | 1 |
| 6 | 系统需要联机数据项吗？ | 3 |
| 7 | 联机数据项是否需要在多屏幕或多操作之间切换以完成输入 | 0 |
| 8 | 需要联机更新主文件吗？ | 4 |
| 9 | 输入，输出，文件或查询很复杂吗？ | 1 |
| 10 | 内部处理很复杂吗？ | 1 |
| 11 | 代码需要设计成可复用吗？ | 2 |
| 12 | 设计中需要包括转换和安装吗？ | 0 |
| 13 | 系统的设计支持不同组织的多次安装吗？ | 0 |
| 14 | 应用的设计方便用户的修改以及使用吗？ | 4 |
| 总计 |  | 20 |
| 调整因子：（0.65 + 0.01 \* 20）=0.85 | | |

### 3.2.5最终工作量估算

FP = (15 + 21) \* 0.85 = 30.6

# 四 风险管理计划

## 4.1 风险分析

### 4.1.1 风险识别

#### 4.1.1.1风险源分析

本次项目计划的风险分析从风险检测表出发，考虑项目组本身以及项目开发过程中的诸多要素，并根据风险程度从低到高得到1至5的风险评分，然后综合风险程度得分较高的各个因素产生项目风险。

风险检测表如下：

|  |  |
| --- | --- |
| **问题** | **风险程度** |
| 开发人员的水平如何？ | 2 |
| 开发人员在技术上的配套？ | 3 |
| 开发人员是否能自始至终地参加开发活动？ | 1 |
| 开发人员是否能全身心地投入到开发工作中？ | 5 |
| 项目给予的时间是否充足？ | 1 |
| 项目的规模是否过大？ | 1 |
| 开发人员数量是否充足？ | 2 |
| 项目需求是否明确？ | 0 |
| 开发人员对项目的各项工作是否有较为清晰的理解？ | 1 |
| 是否有支持项目开发，运行以及测试的物理环境？ | 1 |
| 项目是否制定了良好且合理的进度计划？ | 2 |
| 项目需求是否容易发生变更？ | 0 |

#### 4.1.1.2风险概述

根据上述风险预测表中的各项潜在风险项所对应的风险程度，可以总结出本次项目中的主要风险如下：

1. 开发人员对于项目所需的技术不够熟悉
2. 开发人员可能面临项目与同期的其他工作的影响
3. 项目没有制定良好的进度管理

此外还有若干次要风险：

1. 缺少进行严格测试的物理环境
2. 开发人员开发速度不足以应对整个项目

### 4.1.2 风险预测

#### 4.1.2.1估算风险概率

本次风险分析采用概率区间来描述风险的发生概率，其区间分布如下：

|  |  |
| --- | --- |
| **风险发生概率区间** | **实际发生概率** |
| 低 | 0%~20% |
| 中低 | 20%~40% |
| 中 | 40%~60% |
| 中高 | 60%~80% |
| 高 | 80%~100% |

为了得到各风险的发生概率区间，考虑在风险检测表中，可能导致该风险的因素的风险程度得分，综合经验分析得到其大概的风险概率。

#### 4.1.2.2估算结果

根据1.2.1中的估算方法，得到各风险的发生概率如下表，同时表中还标注了各风险的种类与风险的潜在影响。

|  |  |  |  |
| --- | --- | --- | --- |
| **风险名** | **风险种类** | **风险发生概率** | **风险影响** |
| 1. 开发人员对于项目所需的技术不够熟悉 | 技术风险 | 中 | 开发人员需要额外的时间学习开发所需技术，占用开发时间 |
| 1. 开发人员可能面临项目与同期的其他工作的影响 | 项目风险 | 高 | 开发人员会受到其他工作的影响，分散开发精力，降低开发速率，提高错误发生率 |
| 1. 项目没有制定良好的进度管理 | 项目风险 | 中低 | 在不同工作上所付出的时间与其工作量不能很好匹配，导致项目面对变化的应急能力下降，与其他风险协同容易产生未知的结果 |
| 1. 缺少严格测试的物理环境 | 技术风险 | 低 | 需要额外精力准备测试环境以及测试样例，产生额外时间开销或是软件错误 |
| 1. 开发人员开发速度不足以应对整个项目 | 项目风险 | 低 | 开发速度不足，潜在的延期风险 |

风险按发生概率的排名如下：

② 开发人员可能面临项目与同期的其他工作的影响

① 开发人员对于项目所需的技术不够熟悉

③ 项目没有制定良好的进度管理

⑤ 开发人员开发速度不足以应对整个项目

④ 缺少严格测试的物理环境

### 4.3 风险评估

#### 4.1.3.1评估方法

本次评估考虑可能直接导致项目失败的各个因素，并将这些因素指定为风险参考水准，随后考虑各个风险组合发生时，其组合产生的现实影响以及这些影响对风险参考水准的作用，进而得到这些风险大致评估结果。

#### 4.1.3.2风险参考水准

在本次项目中，能够导致项目失败的只有最终期限deadline后无法提交可运行的可用软件，故唯一的风险参考水准为“进度延迟delay”，为了确认项目终止的状态集合，额外引入一个参考水准“当前时间戳timestamp”表示客观世界的时间进度，其随着时间自然增长不受风险影响。则项目的终止条件为：

此时意味着项目已经不可能在期限时间内完成，项目失败终止。

除此之外，作为开发者必须考虑的因素，为了保证软件质量，错误率fault也应当纳入风险参考水准中，只是对于此次课程项目，fault指标不会导致项目失败，这里引入只是为了使得参考水准影响更为全面。

#### 4.1.3.3评估结果

考虑各个风险之间的组合，其产生的组合效果，并分析其对参考水准的影响，进一步得到风险组合对项目的影响，并对风险等级进行划分，划分的依据以及得到的结果如下：

|  |  |
| --- | --- |
| **风险等级** | **描述** |
| 低 | 风险产生影响不大，不会威胁项目生存 |
| 中 | 风险具有一定影响力，一定程度上会威胁项目生存 |
| 高 | 风险影响力较大，很大概率会威胁项目生存 |
| 紧急 | 风险影响极大，如不处理项目无法完成 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **风险组合** | **组合效果** | **参考水准影响** | **项目影响** | **风险等级** |
| ①③ | 不够优秀的进度管理可能忽略技术上的支持问题，开发者临时学习相关技术会导致开发效率下降，错误率提升 | delay：中等概率提升  fault：大概率提升 | 失败概率中幅增加，产品质量大幅下降 | 中 |
| ②③ | 进度管理如果没有充分考量开发人员因外界干扰而导致的开发速率下降，可能造成预定计划无法完成 | delay：大概率提升  fault：小概率提升 | 失败概率大幅增加，产品质量小幅下降 | 紧急 |
| ③⑤ | 如果进度管理计划没有充分考虑到开发人员的开发水平并适当调整计划，可能导致必要的需求无法完成 | delay：大概率提升 | 失败概率大幅增加 | 高 |
| ③④ | 进度计划若不能给予测试充分的时间，可能导致最终的测试不够充分 | fault：大概率提升 | 产品质量大幅下降 | 低 |
| ①② | 开发人员在干扰情况下还需要学习新的技术容易导致技术熟练度不足 | delay：小概率提升  fault：大概率提升 | 失败概率小幅增加，产品质量大幅下降 | 中 |
| ①⑤ | 如果开发者不熟悉技术又不具备快速开发的能力，则对项目进度会产生显著负面影响 | delay：大概率提升 | 失败概率大幅增加 | 高 |
| ②⑤ | 开发者本身开发速度不足，还会受到其他工作干扰，会严重影响项目进度 | delay：大概率提升 | 失败概率大幅增加 | 紧急 |

## 4.2 风险应对

充分考虑在1.1节中识别的各项风险以及1.3节中的各种风险组合的影响，提出以下风险应对建议：

i. 开发人员技术培训，测试环境搭建等风险应对方案，应当被纳入项目进度计划之中，占据一定的项目时间。

ii. 考虑开发人员开发速率的因素，应当使得待开发的系统尽可能精简，舍弃多余的以及次要的功能，以最小的成本完成用户的最基本需求，在截至时间前能够提交可运行软件。

iii. 增强开发人员之间的沟通（虽然只有1人但假装好像有很多人），了解其项目开发期间的额外工作分布，并进行统筹规划，减少项目开发与其他工作之间的冲突，例如使开发人员能够交替地进行开发以及处理自身其他工作，或是进一步划分工作降低最小粒度，使得开发人员能够在更小的时间单位内进行开发。

iv. 由于高端技术以及丰富功能并非本次项目的主要目标，尽可能减少“新技术”的使用，以能够满足开发需求为标准，降低技术复杂度。

# 五 质量管理计划

## 5.1 质量管理

本次项目的质量管理借助DevCloud开发平台协助完成，以每个迭代周期为基本单位实行质量管理。每个迭代周期将由细化设计，代码开发，测试三个重复环节组成，分别在这三个环节中实现质量管理。

### 5.1.1 细化设计

这一阶段，主要对基于初步设计，结合前后端交互，开发语言，开发框架等信息，完善初步设计中的疏漏，进一步细化设计使之符合需求。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **管理举措** | **执行时间** | **描述** | **通过条件** | **不通过措施** |
| 文档规范检查 | 设计中，设计结束 | 审查细化设计中更新的文档，确保其建模描述统一无歧义 | 细化后的文档采用统一建模语言完成 | 对不符合要求的文档使用UML重新表述 |
| 需求检查 | 设计结束 | 审查细化设计中的各设计项是否符合需求规格说明书，以及需求规格说明书中的相关需求是否被体现 | 细化设计与需求规格说明书一致 | 对不符合需求规格说明书的部分重新设计 |

### 5.1.2 代码开发

这一阶段将使用特定编程语言对初步设计以及细化设计的设计结果进行实现，得到可运行的软件增量。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **管理举措** | **执行时间** | **描述** | **通过条件** | **不通过措施** |
| 单元测试 | 软件模块开发完成后 | 核对单元测试计划，查看各测试用例是否充分，是否得到执行以及是否通过 | 测试用例覆盖能力充分，均得到执行通过 | 补充新的测试用例，或修改代码，并对未完成的测试用例重新执行 |
| 配置管理 | 软件配置变更时 | 检查配置修改是否符合规范以及是否得到执行 | 配置变更符合规范且更新完毕 | 拒绝配置变更，回滚配置项 |
| 非功能需求度量 | 定期 | 审查与预定非功能需求相关的模块，了解其是否为满足此需求进行了相应实现 | 非功能需求得到满足 | 对相关模块的实现进行调整补充 |

### 5.1.3 测试

这一阶段将综合代码开发得到的软件产品，使用测试用例测试其正确性，并核对需求是否得到满足

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **管理举措** | **执行时间** | **描述** | **通过条件** | **不通过措施** |
| 集成测试 | 各模块实现完成以及单元测试后 | 核对集成测试计划，查看各测试用例是否充分，是否得到执行以及是否通过 | 测试用例覆盖能力充分，均得到执行通过 | 补充新的测试用例，或修改代码，并对未完成的测试用例重新执行 |
| 需求校验 | 集成测试后 | 审查当前迭代所需要满足的需求是否得到实现 | 当前迭代所需要满足的需求是已经实现 | 对未满足的需求进行补充设计与实现 |

## 5.2 质量保证

为保证最终软件产品的质量，最终产品符合预期的需求规约，对本次项目流程采取如下的质量保证措施

### 5.2.1文档

#### 5.2.1.1 需求规格说明书

文档描述：描述用户对于软件产品的功能需求以及非功能性需求，包括用例图，用例说明以及非功能性需求约束等信息。

#### 5.2.1.2 软件设计说明书

文档描述：体现软件体系结构以及模块划分，包括UML建模，模块划分，模块关联以及模块交互接口等信息。

#### 5.2.1.3 软件验证以及确认报告

文档描述：囊括了软件验证的计划以及手段，以及软件验证的结果。包括需求的校验，设计校验，软件测试以及相关的审查报告。

#### 5.2.1.1 用户文档

文档描述：从用户的角度阐述软件的使用以及功能，要求能够简洁且有效地展示软件如何使用，便于用户理解

### 5.2.2 评审活动

#### 5.2.2.1 软件需求评审

描述：对初始的需求描述进行评审，组织会议由参与者对初始需求进行打分评价，从中获取实际需求，并逐步生成软件规格说明书。

#### 5.2.2.2 概要设计评审

描述：对软件概要设计文档进行评审，审查其是否符合软件的需求规格说明书

#### 5.2.2.3 详细设计评审

描述：对软件概要设计文档进行评审，审查其是否符合软件的需求规格说明书以及概要设计文档

#### 5.2.2.4 软件验证与确认评审

描述：对实现完成的整个软件的软件产品以及各个过程评审进行验证与确认

#### 5.2.2.5 综合检查

描述：对整个项目进行整体检查，得到最终的评价报告

### 5.2.3 软件配置管理

本次项目的软件配置借助开发工具DevCloud进行托管以及管理并遵循其变更流程以及变更标准。