

电子科技大学

实验报告

学生姓名：Lolipop 学号：2018091202000 指导教师：xx

实验地点：信软楼 304 实验时间：2019.12.12

一、实验名称：项目管理

二、实验学时：4 学时

三、实验目的：

1. 体会软件度量方法的实际应用
2. 体会项目计划的制定

四、实验原理：

软件项目管理定义(IEEE610.12-90)：计划、协调、度量、监控、控制及报告等管理方法在软件开发和维护中的具体应用，以保证整个过程是系统的、有原则的、可量化的。

软件项目管理先于任何技术活动之前，贯穿整个软件生命周期。通过软件项目管理，可以使软件项目能够按照预定的成本、进度和质量顺利完成。

五、实验内容：

1. 小组讨论，确定人员组织方式及分工
2. 将人员组织及分工用 Microsoft Visio 以图形方式绘制
3. 小组讨论确定使用的软件过程模型
4. 估计软件规模，给出估算方法
5. 估计软件工作量和进度（开发时长），给出估算方法
6. 基于估算得出的项目工作量，使用 Microsoft Project 制定项目进度计划
7. 撰写软件项目计划书

六、实验器材（设备、元器件）：

搭载 Windows10 系统的 PC 机；

Microsoft Visio；

Microsoft Project。

七、 实验步骤：

- 1. 小组讨论确定人员组织方式及分工；
- 2. 将人员组织分工绘制成 Visio 图；
- 3. 小组讨论确定试用瀑布模型；
- 4. 估计软件规模，给出估算方法；
- 5. 基于给出的项目工作量，试用 Microsoft project 指定项目进度计划；
- 6. 撰写软件项目计划书。

八、 实验结果与分析（含重要数据结果分析或核心代码流程分析）

- 1. 内部组织结构及人员职责规划如图 1、表 1 所示：

// removed

图 1 人员组织方式及分工

表 1 角色与职责划分

| 姓名 | 身份 | 职责 |
|-----|----|----------------------------|
| xxx | 组长 | 1. 结构化需求分析与软件设计 2. 软件测试 |
| xxx | 组员 | 1. 程序开发 2. 软件测试 |
| xx | 组员 | 面向对象需求分析与软件设计 |

- 2. 软件过程模型：

开发选择采用瀑布模型。

- 3. 软件规模估计：

采用功能点技术对本软件规模进行估计，对应内容如表 2 所示。

表 2 功能点估计表

| 功能分类 | 功能点数 | 具体内容 |
|------|------|---|
| 外部输入 | 11 | 1. 公司组织结构管理 2. 职位信息及职位间工作关系管理 3. 公司人力规划 |

| | | |
|------|---|--|
| | | 4. 对人事成本进行计算和管理 5. 员工的试用、转正、解聘、调职或退休 6. 计划招聘岗位 7. 发布招聘信息 8. 设置绩效考核 9. 培训课程及其详细信息 10. 假期设置、班别设置、考勤项目设置 11. 调班、加班、公出、请假、迟到、早退 |
| 外部输出 | 1 | 生成机构编制表、组织结构图 |
| 外部查询 | 3 | 1. 查询员工信息 2. 管理面试结果 3. 显示考核结果 |
| 外部接口 | 3 | 1. 采集应聘者简历 2. 通知试用 3. 邮件通报考勤信息 |
| 内部文件 | 5 | 1. 公司组织结构文件 2. 员工信息文件 3. 招聘信息文件 4. 培训信息文件 5. 考勤项目信息文件 |

综合上表，计算可得未调节功能点 $UFP = 11*3+1*4+3*3+3*5+5*7 = 96$

影响因素构成的影响度 $N = 3+3+3+2+4+5+4+4+1+0+0+1+0+0 = 30$

复杂度调节因子 $CAF = 0.65+0.01*N = 0.95$

可交付功能点 $DFP = 0.95*96 = 91.2$

可得 Android (Java) 代码行数 $L = 91.2*63 \approx 5.7KLOC$

4. 软件工作量及进度估计：

采用基本 COCOMO 模型。估算出开发本人事资源管理系统项目特征如表 3 所示：

表 3 项目特征表

| 特征 | 特征值 |
|--------|---|
| 工作量 | $MM = 2.4 * (5.7)^{1.05} \approx 14.92$ 人月 |
| 生产率 | $5700 \text{ LOC} / 14.92 \text{ MM} \approx 382.04 \text{ LOC/MM}$ |
| 进度 | $TDEV = 2.5 * (14.92)^{0.38} \approx 7$ 月 |
| 平均配置人员 | $14.92 \text{ 人月} / 7 \text{ 月} \approx 2.13 \text{ FSP}$ |

九、 总结及心得体会：

一个软件，尤其是大型软件的开发需要一个团队，只有团队之间相互协调，理解对方的意思，才能做出一个好的产品。

要对软件规模进行估计，把握好软件开发的时间。循序渐进才能把软件开发好。

十、 对本实验过程及方法、手段的改进建议：

我们可以改进我们的软件过程模型，因为瀑布模型的开发风险大，无法适应变化的需求，软件开发需要迭代。可以使用增量模型或者螺旋模型。降低软件开发后期的风险。

报告评分：

指导教师签字：

附录：

软件项目计划书

项目名称：人事资源管理系统

日期：2019/12/23

1 概述

1.1 项目概述

本人事资源管理系统基于 Android 10（API 29）开发，使用 MySQL 作为后台数据库，预期人事资源管理员在安卓设备上运行使用。主要包括组织管理、人事信息管理、招聘管理、培训管理和考勤管理五大功能，为管理人员提供公司人事资源管理移动端解决方案。

1.2 参考资料

[1] 软件工程——理论，方法与实践，孙家广等编著，高等教育出版社
[2] UML 系统分析与设计教程，冀振燕编著，人民邮电出版社，2009 年

2 项目组织

2.1 内部组织结构

项目开发组织结构图如图 2-1 所示。

// removed
图 2-1 内部组织结构图

2.2 角色与职责划分

项目开发职位分配表如表 2-1 所示。

表 2-1 职位分配表

| 姓名 | 身份 | 职责 |
|-----|----|----------------------------|
| xxx | 组长 | 1. 结构化需求分析与软件设计 2. 软件测试 |
| xxx | 组员 | 1. 程序开发 2. 软件测试 |
| xx | 组员 | 面向对象需求分析与软件设计 |

3 过程计划

3.1 过程模型

项目采用瀑布模型，从一开始选择题目后进行需求分析，再到后来软件设计，设计完成后实现了重要部分代码，再进行了软件测试，运行。由前至后，相互衔接的次序，符合瀑布模型的概念。

4 进度计划

4.1 软件规模估计

采用功能点技术对本软件规模进行估计，对应内容见表 4-1。

表 4-1 功能点估计表

| 功能分类 | 功能点数 | 具体内容 |
|------|------|--|
| 外部输入 | 11 | 12. 公司组织结构管理 13. 职位信息及职位间工作关系管理 14. 公司人力规划 15. 对人事成本进行计算和管理 16. 员工的试用、转正、解聘、调职或退休 17. 计划招聘岗位 18. 发布招聘信息 19. 设置绩效考核 20. 培训课程及其详细信息 21. 假期设置、班别设置、考勤项目设置 22. 调班、加班、公出、请假、迟到、早退 |
| 外部输出 | 1 | 生成机构编制表、组织结构图 |
| 外部查询 | 3 | 4. 查询员工信息 5. 管理面试结果 6. 显示考核结果 |
| 外部接口 | 3 | 4. 采集应聘者简历 5. 通知试用 |

| | | |
|------|---|--|
| | | 6. 邮件通报考勤信息 |
| 内部文件 | 5 | 6. 公司组织结构文件 7. 员工信息文件 8. 招聘信息文件 9. 培训信息文件 10. 考勤项目信息文件 |

假设所有功能的复杂性均为简单。

综合上表，计算可得未调节功能点 $UFP = 11*3+1*4+3*3+3*5+5*7 = 96$

假定影响因素取值如下：

- 1) 备份与恢复：3
- 2) 数据通信：3
- 3) 分布式处理：3
- 4) 性能：2
- 5) 系统配置要求：4
- 6) 联机数据输入：5
- 7) 终端用户效率：4
- 8) 联机更新：4
- 9) 易操作性：1
- 10) 内部处理复杂：0
- 11) 可复用性：0
- 12) 易安装性：1
- 13) 多工作场所：0
- 14) 可维护性：0

影响因素构成的影响度 $N = 3+3+3+2+4+5+4+4+1+0+0+1+0+0 = 30$

复杂度调节因子 $CAF = 0.65+0.01*N = 0.95$

可交付功能点 $DFP = 0.95*96 = 91.2$

可得 Android (Java) 代码行数 $L = 91.2*63 \approx 5.7\text{KL0C}$

4.2 软件工作量和进度估算

对软件工作量的估计采用基本 COCOMO 模型。

本软件总体类型为“组织型”。

可估算出开发本人资源管理系统项目有如下特征如表 4-2 所示。

表 4-2 项目特征表

| 特征 | 特征值 |
|--------|---|
| 工作量 | $MM = 2.4 * (5.7)^{1.05} \approx 14.92 \text{ 人月}$ |
| 生产率 | $5700 \text{ LOC} / 14.92 \text{ MM} \approx 382.04 \text{ LOC/MM}$ |
| 进度 | $TDEV = 2.5 * (14.92)^{0.38} \approx 7 \text{ 月}$ |
| 平均配置人员 | $14.92 \text{ 人月} / 7 \text{ 月} \approx 2.13 \text{ FSP}$ |

4.3 项目进度计划

由项目工作量和进度估算可得项目开发周期约 7 个月，根据项目特征，以图示通过活动列表和时间刻度表示出项目的顺序与持续时间绘制甘特图及表格如图 4-1 和 4-2 所示。

| 标识号 | 任务名称 | 工期 | 开始时间 | 完成时间 | 前置任务 | 资源名称 |
|-----|-------------------|-------------------|---------------------|---------------------|----------------|--------------|
| 1 | 项目范围规划 | 3.5 工作日 | 2019年9月2 E | 2019年9月5 E | | |
| 2 | 确定项目范围 | 4 工时 | 2019年9月2日 | 2019年9月2日 | | 组员1 |
| 3 | 获得项目所需资金 | 1 工作日 | 2019年9月2日 | 2019年9月3日 | 2 | 组员2 |
| 4 | 定义预备资源 | 1 工作日 | 2019年9月3日 | 2019年9月4日 | 3 | 组长 |
| 5 | 获得核心资源 | 1 工作日 | 2019年9月4日 | 2019年9月5日 | 4 | 组长 |
| 6 | 项目范围规划完成 | 0 工作日 | 2019年9月5日 | 2019年9月5日 | 5 | |
| 7 | 分析/软件需求 | 17 工作日 | 2019年9月5 E | 2019年9月30 E | | |
| 8 | 行为需求分析 | 5 工作日 | 2019年9月5日 | 2019年9月12日 | 6 | 组员2 |
| 9 | 起草初步的软件规范 | 3 工作日 | 2019年9月12日 | 2019年9月17日 | 8 | 组长 |
| 10 | 制定初步预算 | 2 工作日 | 2019年9月17日 | 2019年9月19日 | 9 | 组员1 |
| 11 | 工作组共同审阅软件规范/预算 | 4 工时 | 2019年9月19日 | 2019年9月19日 | 10 | 组员2, 组员1 |
| 12 | 根据反馈修改软件规范 | 4 工作日 | 2019年9月20日 | 2019年9月25日 | 11 | 组员1 |
| 13 | 确定交付期限 | 1 工作日 | 2019年9月26日 | 2019年9月26日 | 12 | 组长 |
| 14 | 获得开展后续工作的批准(概要) | 4 工时 | 2019年9月27日 | 2019年9月27日 | 13 | 组员2, 组员1 |
| 15 | 获得所需资源 | 1 工作日 | 2019年9月27日 | 2019年9月30日 | 14 | 组长 |
| 16 | 分析工作完成 | 0 工作日 | 2019年9月30日 | 2019年9月30日 | 15 | |
| 17 | 设计 | 22.5 工作日 | 2019年9月30 E | 2019年10月30 E | | |
| 18 | 审阅初步的软件规范 | 2 工作日 | 2019年9月30日 | 2019年10月2日 | 16 | 组员1 |
| 19 | 制定功能规范 | 5 工作日 | 2019年10月2日 | 2019年10月9日 | 18 | 组员1 |
| 20 | 根据功能规范开发原型 | 4 工作日 | 2019年10月9日 | 2019年10月15日 | 19 | 组员1 |
| 21 | 审阅功能规范 | 10 工作日 | 2019年10月15日 | 2019年10月29日 | 20 | 组长 |
| 22 | 根据反馈修改功能规范 | 1 工作日 | 2019年10月29日 | 2019年10月30日 | 21 | 组员1 |
| 23 | 获得开展后续工作的批准 | 4 工时 | 2019年10月30日 | 2019年10月30日 | 22 | 组员2, 组员1 |
| 24 | 设计工作完成 | 0 工作日 | 2019年10月30日 | 2019年10月30日 | 23 | |
| 25 | 开发 | 22.75 工作日 | 2019年10月31 E | 2019年12月2 E | | |
| 26 | 审阅功能规范 | 1 工作日 | 2019年10月31日 | 2019年10月31日 | 24 | 组员1 |
| 27 | 确定模块化/分层设计参数 | 2 工作日 | 2019年11月1日 | 2019年11月4日 | 26 | 组员1 |
| 28 | 分派任务给开发人员 | 1 工作日 | 2019年11月5日 | 2019年11月5日 | 27 | 组员1 |
| 29 | 编写代码 | 15 工作日 | 2019年11月6日 | 2019年11月26日 | 28 | 组员1 |
| 30 | 开发人员测试(初步调试) | 15 工作日 | 2019年11月11日 | 2019年12月2日 | 29FS-75% | 组员1 |
| 31 | 开发工作完毕 | 0 工作日 | 2019年12月2日 | 2019年12月2日 | 30 | |
| 32 | 测试 | 69.25 工作日 | 2019年10月31 E | 2020年2月5 E | | |
| 33 | 根据产品规范制定单元测试计划 | 4 工作日 | 2019年10月31日 | 2019年11月5日 | 24 | 组员1 |
| 34 | 根据产品规范制定整体测试计划 | 4 工作日 | 2019年10月31日 | 2019年11月5日 | 24 | 组员1 |
| 35 | 单元测试 | 20.5 工作日 | 2019年12月2 E | 2019年12月31 E | | |
| 36 | 审阅模块化代码 | 5 工作日 | 2019年12月2日 | 2019年12月9日 | 33, 31 | 组员1 |
| 37 | 测试组件模块是否符合产品规范 | 2 工作日 | 2019年12月9日 | 2019年12月11日 | 31, 36 | 组员1 |
| 38 | 找出不符合产品规范的异常 | 1.5 工作日 | 2019年12月11日 | 2019年12月13日 | 37 | 组员1, 组员2 |
| 39 | 修改代码 | 10 工作日 | 2019年12月13日 | 2019年12月27日 | 38 | 组员1 |
| 40 | 重新测试经过修改的代码 | 2 工作日 | 2019年12月27日 | 2019年12月31日 | 39 | 组员1 |
| 41 | 单元测试完成 | 0 工作日 | 2019年12月31日 | 2019年12月31日 | 40 | |
| 42 | 整体测试 | 26 工作日 | 2019年12月31 E | 2020年2月5 E | | |
| 43 | 测试模块集成情况 | 10 工作日 | 2019年12月31日 | 2020年1月14日 | 41 | 组员1 |
| 44 | 找出不符合规范的异常情况 | 3 工作日 | 2020年1月14日 | 2020年1月17日 | 43 | 组员1 |
| 45 | 修改代码 | 3 工作日 | 2020年1月17日 | 2020年1月22日 | 44 | 组员1 |
| 46 | 重新测试经过修改的代码 | 10 工作日 | 2020年1月22日 | 2020年2月5日 | 45 | 组员1 |
| 47 | 整体测试完成 | 0 工作日 | 2020年2月5日 | 2020年2月5日 | 46 | |
| 48 | 培训 | 46.75 工作日 | 2019年10月31 E | 2020年1月3 E | | |
| 49 | 制定针对最终用户的培训规范 | 3 工作日 | 2019年10月31日 | 2019年11月4日 | 24 | 组员2 |
| 50 | 制定针对产品技术支持人员的培训规范 | 3 工作日 | 2019年10月31日 | 2019年11月4日 | 24 | 组员2 |
| 51 | 确定培训方法(基于计算机的课堂) | 2 工作日 | 2019年10月31日 | 2019年11月1日 | 24 | 组员2 |
| 52 | 编写培训材料 | 3 周工时 | 2019年12月2日 | 2019年12月23日 | 49, 31, 50, 51 | 组员2 |
| 53 | 研究培训材料的可用性 | 4 工作日 | 2019年12月23日 | 2019年12月27日 | 52 | 组员2 |
| 54 | 对培训材料进行最后处理 | 3 工作日 | 2019年12月27日 | 2020年1月1日 | 53 | 组员2 |
| 55 | 制定培训机制 | 2 工作日 | 2020年1月1日 | 2020年1月3日 | 54 | 组员2 |
| 56 | 培训材料完成 | 0 工作日 | 2020年1月3日 | 2020年1月3日 | 55 | |
| 57 | 文档 | 41.5 工作日 | 2019年10月31 E | 2019年12月27 E | | |
| 58 | 制定“帮助”规范 | 1 工作日 | 2019年10月31日 | 2019年10月31日 | 24 | 组员1 |
| 59 | 开发“帮助”系统 | 5 周工时 | 2019年11月15日 | 2019年12月20日 | 58, 29FS-50% | 组员1 |
| 60 | 审阅“帮助”文档 | 3 工作日 | 2019年12月20日 | 2019年12月25日 | 59 | 组员2 |
| 61 | 根据反馈修改“帮助”文档 | 2 工作日 | 2019年12月25日 | 2019年12月27日 | 60 | 组员2 |
| 62 | 制定用户手册规范 | 2 工作日 | 2019年10月31日 | 2019年11月1日 | 24 | 组员1 |
| 63 | 编写用户手册 | 5 周工时 | 2019年11月15日 | 2019年12月20日 | 62, 29FS-50% | 组员1 |
| 64 | 审阅所有的用户文档 | 2 工作日 | 2019年12月20日 | 2019年12月24日 | 63 | 组员1 |
| 65 | 根据反馈修改用户文档 | 2 工作日 | 2019年12月24日 | 2019年12月26日 | 64 | 组员1 |
| 66 | 文档完成 | 0 工作日 | 2019年12月27日 | 2019年12月27日 | 65, 61 | |
| 67 | 试生产 | 108.25 工作日 | 2019年9月30 E | 2020年2月27 E | | |
| 68 | 确定测试群体 | 1 工作日 | 2019年9月30日 | 2019年10月1日 | 16 | 组长 |
| 69 | 确定软件分发机制 | 1 工作日 | 2019年10月1日 | 2019年10月2日 | 68 | |
| 70 | 安装/部署软件 | 1 工作日 | 2020年2月5日 | 2020年2月6日 | 47, 69, 66, 56 | 组员1 |
| 71 | 获得用户反馈 | 3 周工时 | 2020年2月6日 | 2020年2月27日 | 70 | 组长, 组员1, 组员2 |
| 72 | 评估测试信息 | 0.5 工作日 | 2020年2月27日 | 2020年2月27日 | 71 | 组长, 组员1 |
| 73 | 试生产工作完成 | 0 工作日 | 2020年2月27日 | 2020年2月27日 | 72 | |
| 74 | 部署 | 5 工作日 | 2020年2月27 E | 2020年3月5 E | | |
| 75 | 确定最终部署策略 | 1 工作日 | 2020年2月27日 | 2020年2月28日 | 73 | 组长 |
| 76 | 确定部署方法 | 1 工作日 | 2020年2月28日 | 2020年3月2日 | 75 | 组长 |
| 77 | 获得部署所需资源 | 1 工作日 | 2020年3月2日 | 2020年3月3日 | 76 | 组长 |
| 78 | 培训技术支持人员 | 1 工作日 | 2020年3月3日 | 2020年3月4日 | 77 | 组长 |
| 79 | 部署软件 | 1 工作日 | 2020年3月4日 | 2020年3月5日 | 78 | 组长 |
| 80 | 部署工作完成 | 0 工作日 | 2020年3月5日 | 2020年3月5日 | 79 | |
| 81 | 实施工作结束后的回顾 | 3 工作日 | 2020年3月5 E | 2020年3月10 E | | |
| 82 | 将经验教训记录存档 | 1 工作日 | 2020年3月5日 | 2020年3月6日 | 80 | 组长 |
| 83 | 分发给工作组成员 | 1 工作日 | 2020年3月6日 | 2020年3月9日 | 82 | 组长 |
| 84 | 建立软件维护小组 | 1 工作日 | 2020年3月9日 | 2020年3月10日 | 83 | 组长 |
| 85 | 回顾完成 | 0 工作日 | 2020年3月10日 | 2020年3月10日 | 84 | |
| 86 | 软件开发模板结束 | 0 工作日 | 2020年3月10日 | 2020年3月10日 | 85 | |

图 4-1 甘特图表格

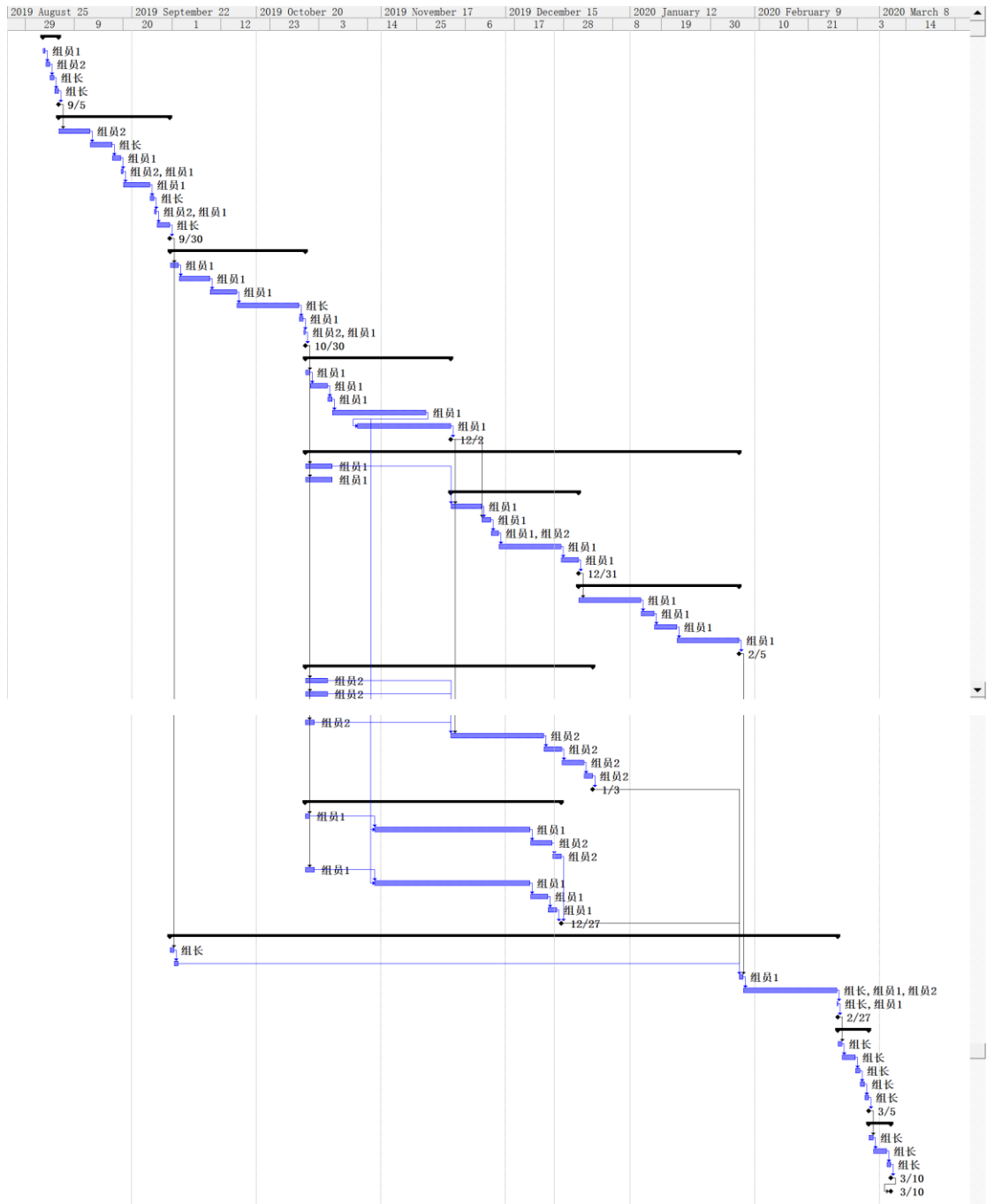


图 4-2 甘特图