

分类号: \_\_\_\_\_

单位代码: \_\_\_\_\_ 10300

密 级: \_\_\_\_\_

学 号: \_\_\_\_\_ 20211243028

# 南京信息工程大学

## 硕士专业学位论文



论文题目: \_\_\_\_\_ 基于敏捷开发的 M 公司软  
件项目管理流程优化研究 \_\_\_\_\_

申请人姓名: \_\_\_\_\_ 闫晶晶 \_\_\_\_\_

指导教师: \_\_\_\_\_ 蒋莹 \_\_\_\_\_

类别名称: \_\_\_\_\_ 工商管理 \_\_\_\_\_

领域名称: \_\_\_\_\_

培养学院: \_\_\_\_\_ 商学院 \_\_\_\_\_

提交时间: \_\_\_\_\_ 2024 年 06 月 15 日 \_\_\_\_\_

二〇二四 年 六 月

# 目 录

第 1 章 绪论.....	1
1.1 研究背景与意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	1
1.2 研究方法与内容.....	2
1.2.1 研究方法.....	2
1.2.2 研究内容.....	3
第 2 章 相关理论与方法 .....	5
2.1 软件项目管理理论.....	5
2.1.1 软件项目管理的内涵.....	5
2.1.2 软件项目管理的流程.....	5
2.1.3 软件项目管理的常见模型.....	6
2.1.4 项目管理常见模型对比分析.....	8
2.2 敏捷开发管理理论.....	9
2.2.1 敏捷开发的内涵.....	9
2.2.2 敏捷开发的管理理念.....	9
2.2.3 敏捷开发的常见管理方法.....	10
2.2.4 敏捷开发常见方法对比分析.....	13
2.3 本章小结.....	14
第 3 章 M 公司项目管理现状调研及问题原因分析.....	16
3.1 M 公司概况和项目管理现状.....	16
3.1.1 公司基本情况.....	16
3.1.2 公司组织架构体系.....	16
3.1.3 公司项目开发管理流程.....	18
3.1.4 公司项目管理流程中问题观察.....	19
3.2 问卷调研收集归纳问题.....	20
3.2.1 调研设计实施与结果分析.....	20
3.2.2 第一次问卷结果收集整理.....	22
3.2.3 第二次问卷结果收集整理.....	23
3.2.4 问卷结果综合归纳.....	24
3.3 层次分析法确定问题权重 .....	25
3.3.1 构建决策问题判断矩阵.....	26
3.3.2 调研问卷对矩阵问题打分.....	26
3.3.3 标准层对目标层的权重计算.....	28

3.3.4 方案层对标准层的权重计算.....	28
3.3.5 各层因素权重及总排序.....	31
3.4 核心问题总结和原因分析 .....	33
3.4.1 业务需求管理存在偏差.....	34
3.4.2 项目过程控制效率有待提高.....	35
3.4.3 研发测试工作质量需要改善.....	35
3.4.4 组织协同机制存在不足.....	36
第 4 章 M 公司敏捷项目管理流程优化方案设计.....	37
4.1 组织协同机制优化方案设计 .....	37
4.1.1 组织架构敏捷化.....	37
4.1.2 构建敏捷看板人力资源池.....	38
4.1.3 引入敏捷项目绩效考评机制.....	39
4.2 项目过程控制优化方案设计 .....	41
4.2.1 定制敏捷看板工作流程.....	41
4.2.2 敏捷看板实现进度可视化.....	43
4.2.3 明确敏捷项目经理职责.....	45
4.2.4 促进团队成员沟通协作.....	46
4.3 业务需求管理优化方案设计 .....	47
4.3.1 需求管理敏捷化.....	47
4.3.2 提升需求设计能力.....	50
4.3.3 制定需求变更流程.....	51
4.4 研发测试工作优化方案设计 .....	53
4.4.1 开发质量提升敏捷化.....	53
4.4.2 测试质量提升敏捷化.....	55
4.4.3 回顾和持续改进.....	56
第 5 章 优化方案的实施与效果跟踪 .....	58
5.1 试点项目背景介绍.....	58
5.2 优化方案实施步骤.....	58
5.2.1 组织协同机制优化方案实施.....	58
5.2.2 项目过程控制优化方案实施.....	59
5.2.3 业务需求管理优化方案实施.....	60
5.2.4 研发测试工作优化方案实施.....	61
5.3 优化方案实施效果跟踪.....	63
5.3.1 成员沟通协作成本降低.....	63
5.3.2 项目交付准期率提升.....	64
5.3.3 业务需求满足度提高.....	65
5.3.4 项目总体交付质量提高.....	66
5.4 本章小结.....	67

第 6 章 结论与展望 .....	68
6.1 研究结论.....	68
6.2 未来展望.....	68
参考文献.....	70
附录 1 德尔菲专家问卷 .....	74
附录 2 问题评价重要性问卷 .....	76

## 插图清单

图 1.1 研究框架图 .....	4
图 2.1 瀑布模型开发流程示例图 .....	6
图 2.2 增量模型开发流程示例图 .....	7
图 2.3 快速原型模型开发流程示例图 .....	7
图 2.4 敏捷模型开发流程示例图 .....	8
图 2.5 看板可视化流程示例图 .....	12
图 3.1 独立网站事业部的组织架构图 .....	17
图 3.2 瀑布项目实施流程图 .....	18
图 3.3 问卷结果综合归纳图 .....	25
图 3.4 标准层对目标层重要性权重图 .....	32
图 3.5 方案层重点考虑因素权重值图 .....	33
图 4.1 项目-需求-任务层级关系图 .....	41
图 4.2 项目立项流程节点图 .....	42
图 4.3 Bug 处理流程 .....	43
图 4.4 项目上线发布流程图 .....	43
图 4.5 项目看板示例图 .....	44
图 4.6 看板个人视角任务示例图 .....	45
图 4.7 需求优先级列表示例图 .....	48
图 4.8 需求变更流程图 .....	51
图 4.9 需求变更操作看板图 .....	52
图 4.10 开发工作流程图 .....	53
图 5.1 沟通协作成本降低比率效果图 .....	64
图 5.2 项目交付准期率对比图 .....	65
图 5.3 线下缺陷数量对比图 .....	66
图 5.4 线上缺陷数量对比图 .....	67

## 附表清单

表 2.1 软件项目管理过程表 .....	6
表 2.2 软件项目管理模型对比表 .....	8
表 2.3 敏捷价值观内容表 .....	10
表 2.4 敏捷实践原则内容表 .....	10
表 2.5 XP 的核心实践内容表 .....	11
表 2.6 敏捷 Scrum 方法实践内容表 .....	11
表 2.7 看板核心实践内容表 .....	13
表 2.8 敏捷常见开发方法对比表 .....	13
表 3.1 德尔菲问卷设计思路表 .....	21
表 3.2 问卷邀请专家成员信息表 .....	22
表 3.3 第一次问卷收集专家问题清单 .....	23
表 3.4 第二次问卷收集专家新增问题清单 .....	24
表 3.5 软件项目管理问题层次结构模型表 .....	26
表 3.6 T.L.Saaty1~9 标度法表 .....	27
表 3.7 问题评价重要性问卷设计思路表 .....	27
表 3.8 标准层对目标层矩阵表 .....	28
表 3.9 层次分析结果表 .....	28
表 3.10 需求管理方案层对标准层矩阵表 .....	28
表 3.11 A 矩阵层次分析结果表 .....	29
表 3.12 项目过程方案层对标准层矩阵表 .....	29
表 3.13 B 矩阵层次分析结果表 .....	29
表 3.14 研发测试方案层对标准层矩阵表 .....	30
表 3.15 C 矩阵层次分析结果表 .....	30
表 3.16 组织架构方案层对标准层矩阵表 .....	30
表 3.17 D 矩阵层次分析结果表 .....	31
表 3.18 权重总排序表 .....	31
表 3.19 业务需求管理存在偏差原因总结表 .....	34
表 3.20 项目过程控制效率有待提高原因分析表 .....	35
表 3.21 研发测试工作质量需要改善原因总结表 .....	35

表 3.22 组织协同机制存在不足原因分析表 .....	36
表 4.1 项目绩效评估表 .....	39
表 4.2 360 评分方案表 .....	40
表 4.3 项目经理工作流程表 .....	45
表 4.4 需求基线内容表 .....	49
表 4.5 测试用例编写标准表 .....	55
表 5.1 项目所属需求示例表 .....	59
表 5.2 任务分解看板示例表 .....	59
表 5.3 项目工作会议示例表 .....	62
表 5.4 业务需求满足度评估表 .....	65

## 摘要

伴随着互联网行业的不断发展，软件项目的需求应用逐渐呈现多元化、个性化的特征。软件项目管理方法也需要与时俱进以响应市场变化和满足用户的需求。敏捷开发管理以其灵活高效、团队协作、过程明晰的特点备受青睐。但是敏捷开发管理在企业实践中也会面临一些问题，因此了解敏捷开发管理不同方法的适用环境，和多角度提供适合企业管理的解决方案，总体提升项目管理能力是目前软件企业比较重视的问题。

本文主要基于敏捷模型看板方法，结合 M 公司软件开发项目管理流程实践，提出了项目管理流程优化方案设计。首先对于软件项目管理、敏捷开发管理理论等相关文献做出阐述、对比和总结。其次，介绍了 M 公司软件项目管理现状和存在问题，并通过在公司选择具有代表性的专家成员，采用德尔菲问卷调研方法收集公司现阶段在开发流程中存在的具体问题，然后采用层次分析法对相关问题因素进行一致性检验和权重排序以确定各问题因素的重要性和优先级，在对排序结果进行原因总结和分析后找出项目管理流程中待优化的方向。再次，本文结合敏捷模型看板管理方法以及其实践原则，针对组织协同机制、项目过程控制、业务需求管理、研发测试工作等四个方面给出对应优化方案设计，并通过试点项目跟踪实施，并选取流程优化前的项目与其进行效果对比评估。最后，对全文研究内容进行总结展望。

本文针对 M 公司的软件项目管理流程提出针对性的解决方案，有利于帮助公司以更高的效率和质量构建更优越的产品和服务，也为其他类似地软件企业进行敏捷开发管理转型提供参考依据。

**关键词：**软件项目管理，敏捷开发管理，敏捷看板方法，流程优化



## Abstract

With the continuous development of the internet industry, the demand for software projects has gradually become more diversified and personalized. The methods of software project management also need to evolve to respond to market changes and meet users' demands. Agile development management, known for its flexibility, efficiency, team collaboration, and clear processes, is favored. However, agile development management faces some challenges in enterprise practice. Therefore, understanding the suitable environments for different agile management methods and providing multi-faceted solutions suitable for enterprise management to overall enhance project management capabilities is a significant concern for software companies today.

This article is mainly based on the agile model Kanban method, combined with the software development project management process practice of Company M, and proposes a project management process optimization scheme design. Firstly, it discusses, compares, and summarizes the relevant literature on software project management and agile development management theories. Secondly, it introduces the current situation and existing problems of software project management in Company M. By selecting representative expert members in the company, the Delphi questionnaire survey method is used to collect specific problems in the current development process. Then, the Analytic Hierarchy Process (AHP) is used for consistency testing and weight sorting of related problem factors to determine the importance and priority of each problem factor. After summarizing and analyzing the sorting results, directions for optimization in the project management process are identified. Furthermore, this article combines the agile model Kanban management method and its practical principles to propose corresponding optimization scheme designs for four aspects: organizational coordination mechanism, project process control, business requirement management, and R&D testing work. The effectiveness is evaluated by comparing it with projects before process optimization through pilot project tracking and implementation. Finally, the article summarizes and looks forward to the research content.

This article proposes targeted solutions for the software project management process of Company M, which is beneficial in helping the company build superior products and services with higher efficiency and quality. It also provides a reference for other similar software companies undergoing agile development management transformation.

**Keywords:** Software Project Management, Agile Development Management, Agile Kanban Method, Process Optimization

## 第1章 绪论

### 1.1 研究背景与意义

#### 1.1.1 研究背景

在目前的数字化浪潮背景下，企业管理和运营日益依赖信息系统和软件。软件已成为支持企业核心业务和产生竞争优势的关键因素，是企业的核心生产力工具。当今商业环境快速变革，新技术、新模式层出不穷，企业业务需要更高的灵活性去应对各种挑战。如何提高软件价值，最终实现敏捷性由软件层向业务侧的传导，以满足多变的业务需求并有效应对市场风险，成为软件企业亟待解决的难题。

目前软件行业市场竞争激烈，用户的需求变化快速且难以预测，这要求软件企业能够快速响应需求变化，缩短产品迭代周期，及时交付软件产品。同时，软件系统日益复杂庞大，对软件质量和安全提出更高要求。而一些早期项目管理方法，比如瀑布管理在面对变化时应对能力不强，以及可能存在的人力资源浪费和一些交付时间长的问题，继而影响到项目交付时间和质量。敏捷开发项目管理作为目前备受欢迎的项目管理方法，强调以人为本、团队协作，采用迭代式、可视化开发流程，可以快速适应变化的需求。敏捷相比传统的瀑布管理更具灵活性，有效减少人力成本和降低项目执行错误率，可提供更系统完整的解决方案。

随着电子商务的蓬勃发展，跨境电商已成为重要的电商模式之一。可以预见，未来一段时期跨境电商仍将保持较高增速，这为一些电商企业提供了广阔的市场机遇。许多企业正在积极开发面向跨境电商场景的软件产品，以满足订单、仓储、物流等管理需求。M公司在跨境电商软件领域占有一席之地，也需要逐步优化软件开发的项目管理方法，以适应公司业务发展的需要。但是敏捷开发方法在企业实践中也会有各自的适用性，目前公司通过 Jira 项目管理工具施行瀑布模型的项目管理方法，在迭代过程中面临业务需求堆积、忽略项目文档、各团队成员沟通协作低效、项目交付质量不高等压力。因此如何构建适合自身的敏捷项目管理方法和流程体系，提高软件交付效率和质量，是企业提升业务运营效率和对外竞争力的关键。

#### 1.1.2 研究意义

为敏捷开发项目管理方法在同类型企业的应用及优化提供了理论基础。本文研究了

具有代表性的中型跨境电商企业 M 公司，在面对业务需求逐渐复杂多元、项目交付效率亟待提升的背景下，在目前开发流程的基础上，对于高复杂性业务使用 Jira 项目管理工具难以有效跟踪项目进度的局限性引入敏捷模型看板(Kanban)管理方法进行可视化系统管理，充分运用已有理论指导分析问题并寻找合适且相对优越的解决方案。希望通过本文的研究能够拓展领域内相关学者对同类型企业应用敏捷开发项目管理方法的研究。

为 M 公司优化现有的开发项目管理流程，最终在公司范围内形成相对成熟完善的项目管理优化方案。本文基于敏捷开发的方法论，分别从组织协同机制、项目过程控制、业务需求管理、研发测试工作等问题方面进行分析探讨。并将对应的优化方案应用在项目实践中，开发效率相对提高，产品缺陷大幅减少。最终在公司范围内形成相对成熟完善的敏捷开发项目管理优化方案，也希望可为国内同类型进行敏捷流程优化的软件企业提供一些借鉴和参考。

## 1.2 研究方法 with 内容

### 1.2.1 研究方法

本文采用德尔菲问卷调查法、层次分析法、案例分析法等，对 M 公司软件项目管理过程中的问题进行分析并提出解决方案。

(1) 德尔菲问卷调查法。本文通过分析相关领域文献和其他企业的案例来设计匿名调研问卷。调研目标群体为在公司软件研发团队的管理层和技术专家。在此调研中，参与者将独立填写问卷，不能与其他参与者或者外部人员交流和讨论，确保意见的独立性。通过两轮问卷调查对收集的问题进行修正和细化，旨在汇总专家们广泛一致的观点，从而获取软件开发流程中存在的问题，为本文后续的研究探讨提供相应解决方案的思路和方向。

(2) 层次分析法。通过前面的问卷调查，将 M 公司当前的软件项目开发的问题分解为多层次的分析结构模型，通过构建判断矩阵，计算权重并做一致性检验分析，然后根据方案层各因素权重进行总排序得出本文要重点解决的问题方向。

(3) 案例分析法。以 M 公司的软件开发团队和项目为样本，结合专家们的反馈意见和现行的项目管理流程实践，分析项目流程中所遇到的问题，结合敏捷开发模型看板(Kanban)方法，给出切实可行的优化方案并跟踪实施。

## 1.2.2 研究内容

本文以 M 公司软件项目管理流程为研究对象，在公司现行软件项目管理流程现状的基础上，结合软件项目管理、敏捷项目管理、敏捷看板管理等理论，利用德尔菲专家问卷调查法、层次分析法得出 M 公司软件项目管理流程中待优化的方向并对其原因进行分析总结。根据原因分析结论给出针对性的优化方案设计，并在公司选择 S 项目为试点项目，跟踪优化方案实施过程，并对实施效果进行评价。

论文共分为以下六个章节：

第一章为绪论。主要介绍研究背景与意义、研究方法和内容。并特别强调采用敏捷看板管理方法以增强团队在管理工作流程中的可视化过程管理、沟通协作和研发效率。

第二章对相关理论与方法进行阐述。介绍软件项目管理、敏捷开发管理等理论基础和研究小结。并阐述各管理理论各自的优势和其在企业中实践可行性。

第三章为 M 公司项目管理现状调研及问题原因分析。介绍 M 公司团队组织架构体系和软件项目管理现状和存在问题等进行细致分析，并对问题进行总结和原因分析。

第四章为敏捷开发的项目管理流程优化方案内容。分析以敏捷看板方法从组织协同机制、项目过程控制、业务需求管理、研发测试工作等四个方面来进行项目管理流程优化。

第五章为优化方案的实施和效果跟踪。从产研团队选取试点项目 S 进行优化方案实施跟踪，并结合优化方案实施前选取的项目 T 进行综合对比，根据实施结果阐述了项目流程优化的效果。

第六章为总结与展望。针对本文的研究做了简要总结，在证明流程优化效果的基础上，找到其不足之处并展望未来的研究思路。

本文研究框架见图 1.1。

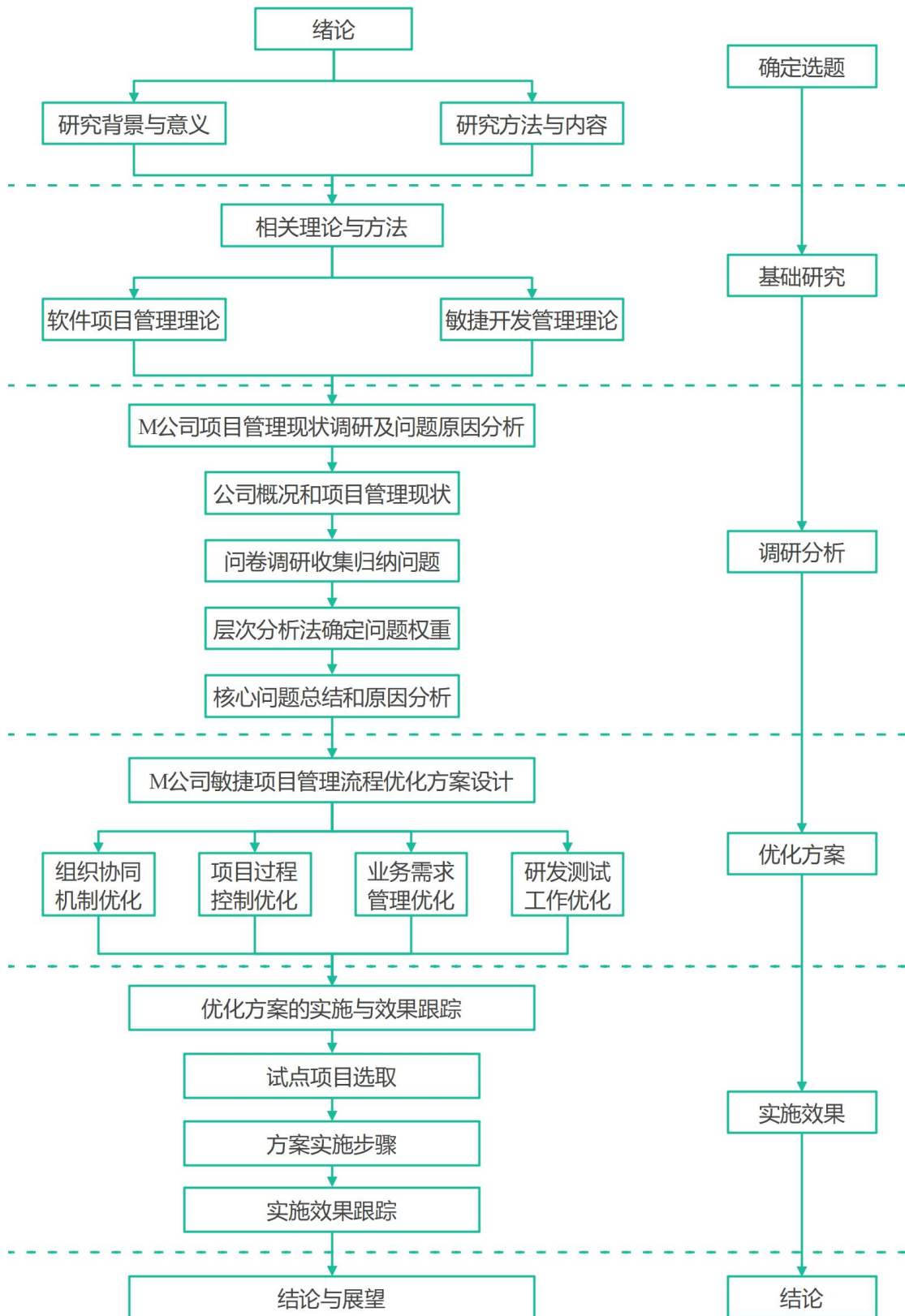


图 1.1 研究框架图

## 第2章 相关理论与方法

### 2.1 软件项目管理理论

#### 2.1.1 软件项目管理的内涵

20 世纪 40 年代后, 软件项目管理学科迅速发展起来, 并被视为解决“软件危机”的关键措施之一, 国内的学者也在软件开发项目管理领域进行了大量的研究。软件项目管理是在规定的时间、成本和质量要求下, 对软件开发过程进行规划、组织、协调和控制的活动。当前, 软件项目管理仍处于不断发展阶段, 需要不断探索创新。由美国项目管理协会制定的《PMBOK 指南》即项目管理知识体系指南目前已经更新到第七版, 该书籍也已经给成为项目管理专业人士认证考试的重要参考资料之一。

随着软件发挥的作用愈发重要, 软件项目管理的复杂度不断提升, 仅依靠项目管理人员可能难以突破工作效率瓶颈<sup>[1]</sup>。在软件开发过程中重视软件开发流程和方法可以显著提升开发效率和软件的可复用性, 优化软件的运行效率, 提高软件产品的质量<sup>[2]</sup>。其中团队成员协作低效和对风险不够重视会影响项目管理的有效性, 所以, 管理负责人需要关注改进项目管理系统和提升相关人员的专业技能和职业素养问题<sup>[3]</sup>。目前企业内部的需求是多变和逐步细化的, 期间不同的工作环节有不同的工具和流程, 这些内容无法快速整合, 就无法系统的管理软件开发全过程, 这给今天的软件开发团队带来了挑战。

因此, 软件工程项目需要多人协作, 建立有效和完善的项目管理机制, 对保证软件项目实现自身价值至关重要<sup>[4]</sup>。在软件开发项目管理的技术标准制定中, 缺乏完善的总体计划和阶段性计划是一个常见的问题, 从而导致软件开发的方向不够明确, 严重影响项目的顺利实施<sup>[5]</sup>, 软件项目失败的事件仍时有发生。

#### 2.1.2 软件项目管理的流程

软件项目管理的主旨是以实现项目目标为中心, 通过启动、规划、执行、监控、收尾等五大过程管理来满足用户需求的过程<sup>[6]</sup>。对这五个过程的管理控制, 就是对项目进行了管理, 五个过程有明确的依存关系并在项目中按照顺序执行, 五个过程相互重叠作用, 构成了项目管理的完整框架。软件项目管理五大过程见表 2.1。

表 2.1 软件项目管理过程表

项目管理阶段	关键活动
启动阶段	定义项目目标、确定项目范围、识别项目相关者、初步制定计划、成本预算和时间表。
规划阶段	详细规划任务分配和资源分配、安排时间线、制定风险管理计划、确定项目里程碑和交付物。
执行阶段	执行开发任务、监督项目进展、实施质量保证活动，如代码审查和测试。
监控阶段	监控项目进度、与团队和项目相关者沟通、根据需要进行项目调整。
结束阶段	交付最终产品、完成文档工作、释放资源和解散团队、进行项目回顾，总结经验。

### 2.1.3 软件项目管理的常见模型

#### (1) 瀑布模型

1970 年温斯顿·罗伊斯 (Winston Royce) 提出了著名的“瀑布模型”，它是项目管理相关理论中最早被提出的，也一直是被广泛采用的一种软件开发项目管理模型。直到现在，仍有很多软件开发企业选择瀑布项目管理。严格把软件项目的开发分成各个开发阶段：需求规划、需求分析、软件设计、程序编码、软件测试、运行维护等<sup>[7]</sup>。这种管理流程使用里程碑的方式，严格定义了各开发阶段的输入和输出，如果达不到输出的要求，下一阶段的工作就不展开<sup>[8]</sup>。因此，瀑布管理模型严格定义了每个节点的要求，软件需求不能频繁变更而且每个开发阶段需要非常清晰明了。瀑布模型开发流程示例见图 2.1。

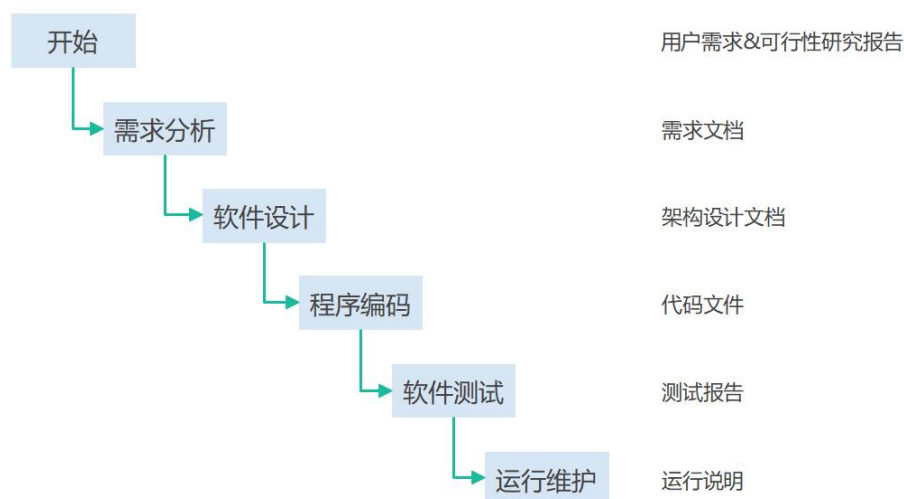


图 2.1 瀑布模型开发流程示例图

#### (2) 增量模型

增量模型也称渐增模型。它将一个完整的软件拆分成不同的组件，然后逐个组件地开发测试，每完成一个组件就展现给客户，让客户确认这一组件功能和性能是否达到客户需求，最终确定无误，将组件集成到软件体系结构中<sup>[9]</sup>。整个开发工作被组织为一系列短期、简单的小项目，称为迭代，每一个迭代都需要经过需求分析，软件设计，编码，

测试的过程。增量模型开发流程示例见图 2.2。

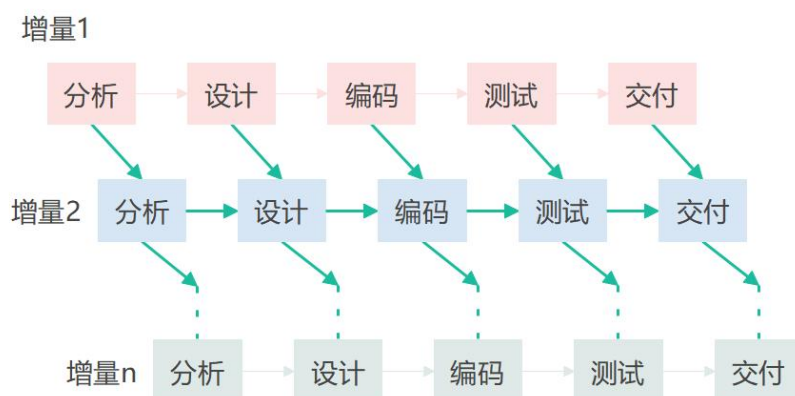


图 2.2 增量模型开发流程示例图

### （3）快速原型模型

快速原型模型是指在开发真实系统之前，先构造一个原型，在该原型的基础上，逐渐完成整个系统的开发工作。原型模型主要有两个阶段：原型开发阶段，软件开发人员根据用户提出的软件系统的定义，快速地开发一个原型。该原型应该包含目标系统的关键问题和反映目标系统的大致面貌，展示目标系统的全部或部分功能、性能等；目标软件开发阶段，在征求用户对原型的意见后对原型进行修改完善，确认软件系统的需求并达成一致的理解，进一步开发实际系统<sup>[10]</sup>。快速原型模型开发流程示例见图 2.3。

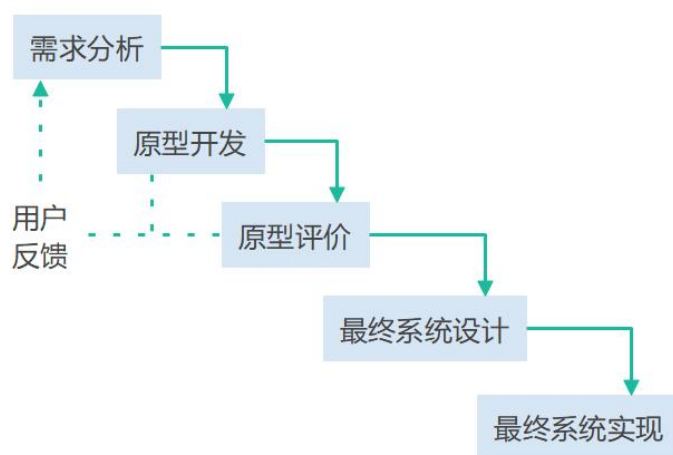


图 2.3 快速原型模型开发流程示例图

### （4）敏捷模型

敏捷模型是 20 世纪 90 年代兴起的一种软件开发管理模型。敏捷模型以对用户需求的快速响应为核心，采用迭代、循序渐进的方法进行软件开发<sup>[11]</sup>。在敏捷模型中，软件项目在构建初期被拆分为多个相互联系而又独立运行的子项目，然后迭代完成各个子项目，开发过程中，各个子项目都要经过开发测试。当客户有需求变更时，敏捷模型能够



迅速地对某个子项目做出修改以满足客户的需求<sup>[11]</sup>。在这个过程中,软件一直处于可使用状态。除了响应需求,敏捷模型还有一个重要的概念,就是迭代,就是不断对产品进行细微、渐进式的改进,每次改进一小部分,如果可行再逐步扩大改进范围。敏捷模型开发流程示例见图 2.4。

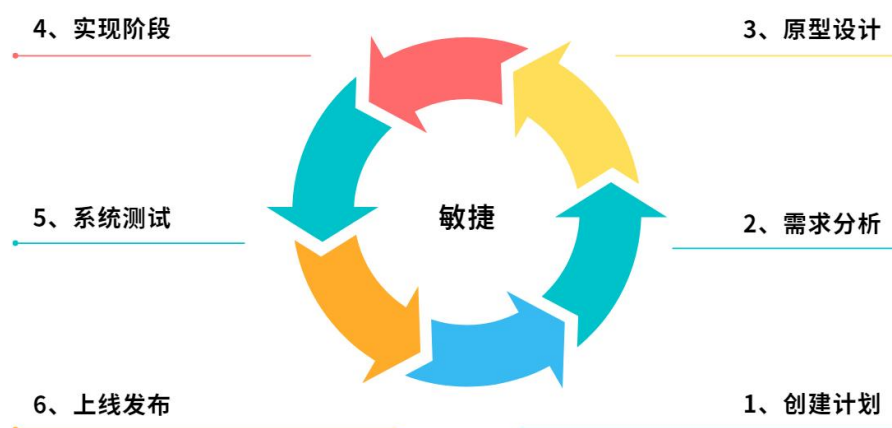


图 2.4 敏捷模型开发流程示例图

## 2.1.4 项目管理常见模型对比分析

### （1）软件项目管理常见模型对比

在对上述四种软件开发管理模型的内涵和开发工作流程分别阐述后,为了更好的理解各个模型的特点和适用场景,本文对这四种模型进行了对比,内容见表 2.2。

表 2.2 软件项目管理模型对比表

特性/模型	瀑布模型	增量模型	快速原型模型	敏捷模型
开发过程	线性顺序执行	阶段性, 逐步增加	重复迭代, 快速原型制作	强调适应性和灵活性
灵活性	低 (难以应对变化)	中等 (部分适应变化)	高 (易于根据反馈调整)	高 (持续适应变化)
客户参与	通常仅在开始和结束阶段	阶段性评估和反馈	持续反馈, 高度参与	持续反馈, 高度参与
优点	流程清晰、有序, 易于管理和测量	适应需求变化, 分阶段交付增加可见性	易于发现需求错误, 降低维护成本	高度灵活、可视化, 强调客户反馈和快速持续交付、拥抱变化
缺点	难以适应需求变化, 后期发现问题成本高	早期版本功能不全可能导致混乱, 需要频繁的管理和调整	可能过分依赖原型, 最终产品质量不一	缺乏对文档的重视; 强调快速反应, 项目可能面临风险

## （2）敏捷模型灵活适应变化

通过对以上几种软件开发项目管理模型进行对比后，可以看到在具体的企业管理实践中，每种模型都有其优缺点和其适用性。因此，公司在推进项目管理体系时，要依据公司业务需求和团队人员状况选择不同的模型<sup>[12]</sup>。

其中敏捷模型，以其对变化的高度适应性和强调客户反馈、强调沟通协作和避免资源浪费的优势<sup>[13]</sup>，比较适合互联网领域的快速和个性化需求，目前已成为当前主流的软件过程改进与交付模式。在项目开发实践过程中，客户可以随时提出需求变更，项目团队能够迅速响应并调整项目计划<sup>[14]</sup>。这种以客户为中心的管理方式，有助于提高客户满意度，降低项目风险。

## 2.2 敏捷开发管理理论

### 2.2.1 敏捷开发的内涵

敏捷开发管理的诞生标志着软件开发领域对于日益增长的项目管理挑战的一种革命性回应。2001年，敏捷联盟的成立和敏捷宣言的发布，由 Jeff Sutherland 等 17 位软件开发领域的领军人物一起在美国犹他州雪鸟滑雪胜地提出敏捷开发这个概念，共同发表了一份《敏捷宣言》。敏捷开发是一种以价值驱动的项目管理模型，就是用高效的方法交付有价值的产品<sup>[15]</sup>。敏捷开发管理的目标是用最短的时间交付最大的业务价值，通过管理者与被充分授权的团队成员共同完成项目目标，并以风险最小及自我组织的方式完成令客户满意的成果。

### 2.2.2 敏捷开发的管理理念

敏捷开发是一种以人为本、迭代和适应性强的软件开发实践，它强调团队合作、客户反馈以及能够快速响应变化的能力，旨在提高软件开发过程的效率和适应性。敏捷开发的核心是用户需求的变化，敏捷模式可以实现工作过程的流程化和灵活化，具备可视和可集成的特征<sup>[16]</sup>。同时，敏捷开发管理模型还强调持续优化，改善延期交付，降低缺陷率，提高项目的交付质量<sup>[17]</sup>。

敏捷宣言包括 4 个价值观和 12 条指导原则，4 个价值观是敏捷方法论的核心内容，而这 12 条原则进一步详述了实现敏捷宣言的实践和方法<sup>[18]</sup>。敏捷方法论的这种结构旨在促进一个更加动态、协作和响应快速变化需求的开发环境，从而提高项目成功率和产品质量。敏捷 4 个价值观内容见表 2.3。

表 2.3 敏捷价值观内容表

序号	敏捷宣言核心内容
1	个体和互动高于流程和工具
2	可工作的软件高于详尽的文档
3	客户合作高于合同谈判
4	响应变化高于遵循计划

敏捷 12 条指导原则内容见表 2.4。

表 2.4 敏捷实践原则内容表

序号	实践原则内容
1	我们的最高目标是通过早期和持续地交付有价值的软件来满足客户的需求。
2	欢迎需求变化，即使在开发后期也能够这样做，敏捷过程利用响应变化作为竞争优势。
3	经常交付可工作的软件，频率从几周到几个月，以较短的周期为优。
4	业务人员和开发者必须每天都工作在一起，直到项目结束。
5	围绕被激励的个体建立项目，给他们创造环境和提供支持，然后完全信任他们能够完成工作。
6	最有效和最具效率的方法是面对面的交流。
7	可工作的软件是进度的主要衡量标准。
8	敏捷过程促进可持续开发，项目相关者、开发者和用户应该能够长期保持恒定的步伐。
9	不断地追求技术卓越和良好的设计增强敏捷性。
10	简洁，要尽量减少不必要的工作。
11	最好的架构、需求和设计来自于自组织的团队。
12	团队定期地反思如何变得更加高效，然后相应地调整和优化自己的行为。

### 2.2.3 敏捷开发的常见管理方法

#### （1）极限编程(XP)

极限编程(XP)作为一种以小中型团队协作、迭代开发和持续改进为核心的敏捷软件开发方法，旨在适应快速变化的需求环境中交付高质量软件<sup>[19]</sup>。XP 在迭代生命期中贯穿了 12 条实践，其中持续集成和测试驱动开发是两个关键技术实践。持续集成要求团队成员频繁地将他们的工作集成到共享代码库中<sup>[20]</sup>，这有助于早期发现错误和缺陷，确保代码的一致性和稳定性。而测试驱动开发则要求在编写实际代码之前先编写测试用例，以测试为导向进行开发，这确保了软件的质量从一开始就得到保障。XP 的 12 个核心实践见表 2.5。

表 2.5 XP 的核心实践内容表

实践内容	内容解释说明
现场客户	XP 同其他的快速开发一样，要求客户在现场持续地参与到项目组中。
策划游戏	以业务优先级和技术估计为基础，决定下一步计划发布的范围。
小版本发布	XP 项目组以非常短的周期按递增的方式发布新版本。
平稳的开发速度	保持每周工作 40 小时，保证每天的工作都是高效的，不能连续两周加班。
系统隐喻	使用类比或比喻简化系统架构和设计的沟通。
简单设计	XP 认为代码设计应该尽可能的简单，只要满足当前功能的要求，不多也不少。
结对编程	两人合作共同编程，一个人编写代码，另一人负责审查以保证代码的正确性和可读性。
编码规范	制定统一的编码标准和规范，确保代码一致性。
测试驱动的开发	先编写测试代码，再编写可使测试通过的功能代码。
持续集成	只要有任务完成，就集成新代码，构造系统，检查系统版本的所有测试工作。
重构	定期优化代码结构，提高代码质量。
代码集体所有	整个团队拥有所有代码，任何人都可以更改任何部分。

## (2) Scrum 方法

敏捷 Scrum 是敏捷开发模型中比较广泛的一种实践，由 Jeff Sutherland 和 Ken Schwaber 于 1995 年共同提出，创立者将 Scrum 设计为能支持开发复杂软件并能持续提高生产力和创造性，帮助交付高价值产品的方法。它采用迭代开发，把项目拆分为更小、更易于管理的任务<sup>[21]</sup>，每个固定周期结束交付一部分功能，使团队能快速交付产品和功能。它鼓励团队适应变化，使其更容易处理需求变更和不确定性，团队可以在每个周期中重新评估和调整项目方向<sup>[22]</sup>。敏捷开发中的 Scrum 方法的各个阶段和活动的的内容见表 2.6。

表 2.6 敏捷 Scrum 方法实践内容表

类别	名称	解释说明
角色	产品负责人	定义产品目标和优先级，管理产品功能列表。
	敏捷教练	帮助团队遵循 Scrum 流程，解决障碍。
	开发团队	负责完成本次 Sprint 开发任务，交付产品。
物件	产品功能列表	由产品负责人管理的所有需求和功能列表。
	迭代任务列表	团队从产品功能列表中选出的某一 Sprint 将要完成的任务。
	燃尽图	跟踪 Sprint 进度的可视化视图和剩余工作量的图表。

续表 2.6 敏捷 Scrum 方法实践内容表

类别	名称	解释说明
仪式	Sprint 计划会议	确定本次 Sprint 目标和迭代任务列表。
	每日站会	团队相互了解彼此的进展，聚焦用户需求的交付共同解决障碍。
	Sprint 评审会	Scrum 团队在会议中向最终用户真实展示工作成果，获取直接反馈。
	Sprint 回顾会	检查并调整，团队停下手上的工作，回顾过去的 Sprint，寻找改进机会。

### （3）看板(Kanban)方法

看板方法属于敏捷开发的一种，也是目前比较受欢迎的敏捷实践方法，因此它具有敏捷开发所具有的一切特性。看板管理源于日本的丰田公司在汽车制造流程中被引入，采用看板管理对生产过程的各个环节进行控制以及可视化的管理，以及促进并实现持续改进<sup>[23]</sup>。在敏捷方法中，看板是一个动态的管理工具，可以显示项目中每项工作的流动性，并且可以识别瓶颈，可以实现质量问题的有效控制<sup>[24]</sup>。任务写在卡片上，从一栏到下一栏，直到任务完成。看板可视化流程示例见图 2.5。

## Kanban Board

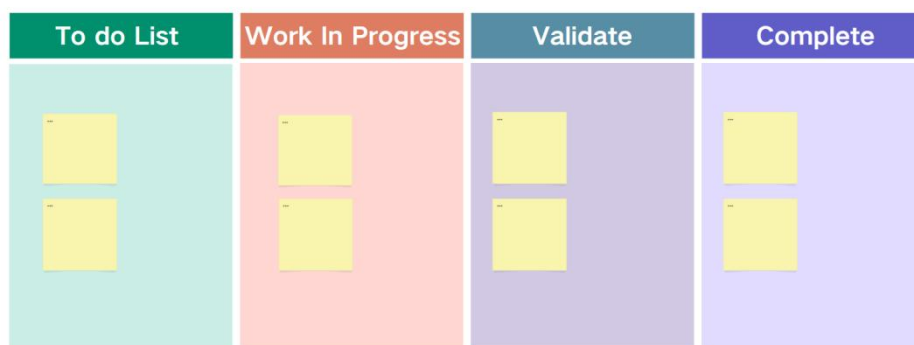


图 2.5 看板可视化流程示例图

看板最核心理念在于减少浪费，它通过使用看板卡片和物理看板这些可视化工具，使管理层能够直观地看到整个生产流程，从而及时、准确地作出供应决策<sup>[25]</sup>。可视化管理最大程度地帮助人们看到整个流程的运转，辅助管理层及时准确地制定供应计划。同时，生产效率和成本控制是产品保持市场竞争力的决定性因素<sup>[26]</sup>。受到了丰田生产系统影响，David J. Anderson 将其进一步适应并推广到 IT 和软件开发领域，以及其他类型的工作环境中。看板(Kanban)也逐渐成为了最新版的《敏捷实践指南》中的一类方法，近几年被广泛应用。看板管理有利于减少生产过程中的浪费，节约生产资源，以需求推动工作任务的推进。看板方法的核心实践流程见表 2.7。

表 2.7 看板核心实践内容表

看板核心实践	解释说明
可视化工作流	使用看板来展示任务的不同阶段（如“待办”、“进行中”、“审查中”、“已完成”），让团队成员清晰地看到每项任务的状态，从而更好地理解工作流程和任务进展，提高工作效率和交付速度。
限制在制品数量	通过设置在制品（WIP）限制，鼓励团队集中精力完成当前任务，避免同时启动多个任务。这有助于减少任务切换成本，提高流程效率。
管理流动	关注监控和优化任务从开始到完成的整个流动过程。通过分析任务的停留时间识别流程瓶颈，并采取措施改进，加速工作流动。
明确的流程规则	明确定义工作流程和规则，包括任务如何移动、新任务的添加条件以及 WIP 限制的设置等，确保团队理解期望和责任，保持流程一致性和可预测性。
改进和演化	鼓励根据反馈和度量指标不断调整和优化流程。采取渐进式的改进方法，帮助团队在变化环境中保持灵活性和竞争力。

### 2.2.4 敏捷开发常见方法对比分析

#### （1）敏捷开发管理方法对比

在对 XP、Scrum 和看板(Kanban)三种敏捷开发方法的具体实践内容进行阐述后，为了更好的理解各个方法的优缺点和适用场景，本文对这三种方法进行了对比，内容见表 2.8。

表 2.8 敏捷常见开发方法对比表

敏捷方法	适用性	优点	缺点
极限编程 (XP)	适合需求频繁变更的小中型团队项目。	强调技术卓越和良好的开发实践；提高产品质量和响应变更的能力；强调客户的持续参与。	对团队成员的技能水平要求较高；对于规模较大的项目可能难以管理。
Scrum	适合需求变化频繁且期望快速迭代开发的中小型项目	清晰的框架和周期性迭代，增强对需求变化的适应性，客户前期满意度高。	需要高度自律的团队沟通协调；忽略文档的重要性；任务优先级不明确，大项目容易遇到瓶颈问题。
看板(Kanban)	适合多种规模的项目，特别是需要持续交付的维护和运营项目，以及对工作流优化有需求的团队。	灵活性极高，可视化；适应于持续交付；鼓励团队协作合作；强调流程可视化和持续改进。	如果成员没有及时更新任务阶段，依赖该任务的后续环节可能受阻，需要人监督控制。

在系统的介绍了三种敏捷方法的实践流程和对其做了简单对比后，本文发现每种方法都有其对应的适用场景、明显的优势和一些可能存在的缺点。XP 通过强调技术卓越

和良好开发实践,非常适合那些需求变更频繁的小至中型项目,对于大型项目或分布式的团队适用性则存在争议<sup>[27]</sup>。

**Scrum** 以其结构清晰、角色分明的框架著称,特别适合需求明确但可能变化的项目,能有效促进团队的自我管理和项目的适应性发展。但是,这种快速迭代敏捷式开发,也不适用于所有的软件项目,对于一些大型、复杂的项目来说,**Scrum** 的周期迭代机制难以覆盖所有内容,另外不太详尽的文档和跟踪可能会有实现误差,这将增加协调和沟通的复杂性<sup>[28]</sup>。

## (2) 看板成为工作流程优化的理想选择

看板(Kanban)以其极高的灵活性和可视化流程的特点,成为了持续交付和工作流程优化的理想选择。一个动态的项目管理工具可以通过数据可视化让项目状态和进度清晰的展示出来<sup>[29]</sup>,防止未来出现与沟通相关的问题,有效的规避风险。看板管理实践中成员可以同时通过看板,了解自己的任务信息,避免了信息技术传递过程中的遗漏<sup>[30]</sup>。因此,选择敏捷看板方法,意味着项目团队会通过循环的回顾和反馈,持续改进工作流程<sup>[31]</sup>。通过这种简化的、可视的、不断改进的工作实践流程,也能够更好地应对挑战,同时保持项目的竞争力和适应性。

敏捷关注业务价值,通过项目分解可以根据需要进行变更,**Scrum** 和看板(Kanban)是目前比较受欢迎的两种敏捷模型管理方法,但是看板(Kanban)在连续流程环境中更具操作性,具有稳定的系统改进能力,在资源分配调度方面优于 **Scrum**<sup>[32]</sup>。并且在敏捷开发实践中,增量迭代开发存在用户需求较多,项目迭代周期较长以及部门间缺乏协作等问题<sup>[33]</sup>,而看板管理可以很好的解决以上问题,在实现商业价值基础上促进团队协作和持续改进。

## 2.3 本章小结

通过对软件项目管理理论的整理和分析,可以看出国内外在这一领域普遍采用的软件项目管理方法包括瀑布、增量、快速原型、敏捷开发等多种模型。领域内学者对于主流软件项目管理方法介绍了定义和实践步骤等。选择合适的项目管理方法可以提高项目交付效率和质量,传统的瀑布方法和敏捷的多种方法也有其不同的适用条件。目前在敏捷开发管理方面,**Scrum** 是使用率比较高的一种方法,但是 **Scrum** 方法实践中的一些挑战,例如缺乏工作可见性、任务优先级、沟通协作、以及迭代周期长难以尽快完整交付项目的问题,使用看板(Kanban)方法可以很好的解决。看板的具体实践相对简单,但是

高效可视化的团队协作可以更好地提高项目效率和质量，提升团队的凝聚力，同时也大大提升了项目交付度和满意度。因此，在项目管理实践中，企业需要选择一套与公司业务发展方向相符的项目管理方法，以达成企业目标。



## 第3章 M公司项目管理现状调研及问题原因分析

### 3.1 M公司概况和项目管理现状

#### 3.1.1 公司基本情况

M公司成立于2015年，在跨境电商领域有着长达8年的行业经验，品牌致力于为全球消费者提供最有魅力的快时尚服饰产品。公司专注于打造服饰专业品牌线，且开始通过自主研发深耕该领域，由此确立了自身清晰的品牌定位，到2018年成为美国较受欢迎的快时尚服饰品牌之一。至2023年，M公司正在积极开拓多品类、新渠道、新市场，丰富多元化的商业版图。M公司由一个始终保持年轻、创新、拼搏精神的团队打造，拥有自建网络销售平台、独立设计和一体化的供应链系统，在整个团队的持续努力奋斗之下，其员工遍布全球，品牌产品也正在向全球各大洲的更多国家销售，并迅速获得客户青睐。M公司客户几乎遍布全球每个大洲和多个国家，积极探索市场和了解世界各地客户的喜好，且公司年销售额不断增长。

#### 3.1.2 公司组织架构体系

目前公司总部在N城市，其他城市设立分公司。在职员工约为500余人，由下图可以看出，公司组织结构是由职能划分的。纵向是职能，横向是架构。公司在董事会下设置独立网站事业部，该事业部主要由创意中心、互联网研发中心、营销运营中心等部门组成并分别管理。各职能部门包含UED设计部、流程部、产品部、研发部、质量部、运维部、算法部、用户运营部等。各部门人员的具体工作是由其部门负责人进行管理和调配。当前M公司的独立网站事业部的组织架构以及职能见图3.1。

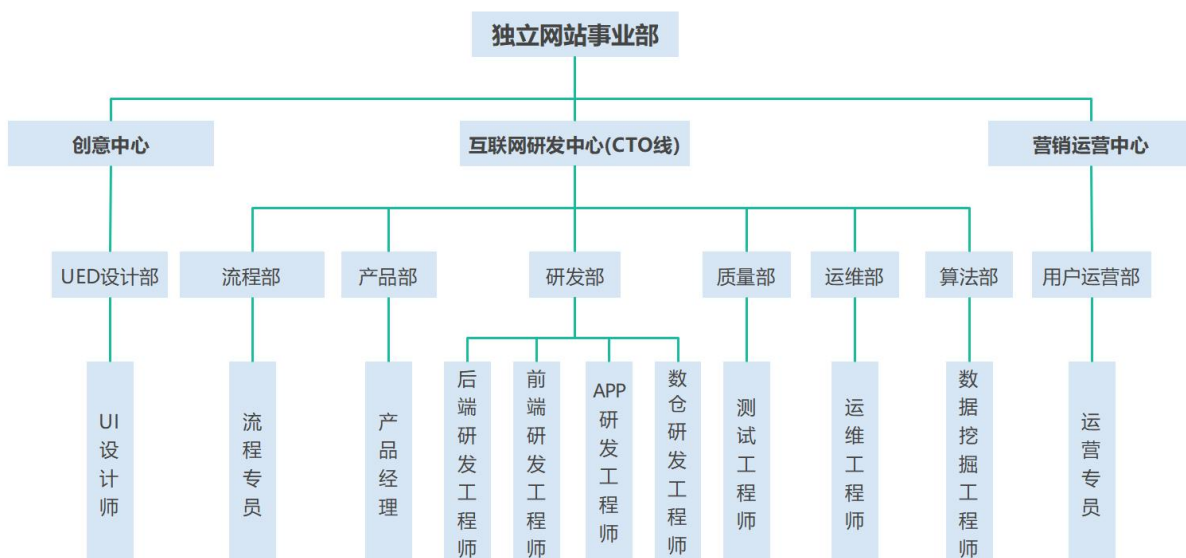


图 3.1 独立网站事业部的组织架构图

互联网研发中心首席技术官(CTO)是公司技术方向和战略的管理者。他领导技术团队，确保技术和研发方向与公司的愿景和战略一致。其职责包括技术创新、技术架构规划和领导研发团队。

产品部负责整体产品策略规划，并统筹产品生命周期管理。产品经理主要职责是作为产品和技术团队之间的纽带，确保研发方向符合产品策略，推进项目按需求规范实施，最终输出满足用户需求的优质产品。

研发部负责研发、设计和维护公司的技术产品和解决方案。研发工程师的主要职责是在限定时间和成本范围内确保产品的质量、性能和安全，根据各自的专业方向，实现需求功能，提供综合技术支持。

UED 设计部负责产品的视觉设计、界面设计和用户体验设计。设计师的主要职责是与产品经理和研发人员合作，确保产品的页面和外观交互满足用户期望，并提高产品的可用性。

质量部负责进行测试和质量控制，确保产品符合质量标准。测试工程师的主要职责是执行各种测试，包括功能测试、性能测试、安全测试等，确保产品在上架之前经过严格的测试和验证。

算法部负责研究和开发新算法、优化现有算法，以提高产品的效率和性能。数据挖掘工程师的主要职责是开发和实施数据挖掘，进行数据分析和算法编写及测试，并与其他团队协作，为业务决策和产品方法提供支撑。

用户运营部负责与用户互动，了解他们的需求和反馈。运营人员的主要职责是促进

用户参与和反馈，以改进产品和服务，回应用户的查询和解决用户遇到的问题。

### 3.1.3 公司项目开发管理流程

M 公司目前采用的软件项目管理方法主要是基于瀑布开发模型。具体的项目实施流程见图 3.2。

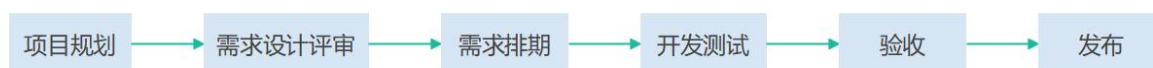


图 3.2 瀑布项目实施流程图

#### （1）项目规划阶段

项目启动后由项目相关团队成员参与，各团队产品经理在完成后召开会议，跟成员明确该版本的目的、需要做的需求等等，但不必须明确需求文档细节，主要讨论需求方案可行性以及正式需求评审前产品经理需要完成的工作。

#### （2）需求设计评审阶段

在需求设计评审阶段，产品经理需要整理需求文档，召开团队成员会议进行评审。需求评审会上团队成员可以进行提问和讨论。如果是功能比较复杂或者规模比较大的版本，在进行需求评审后，团队成员可能会反馈一些问题，需要会后产品和业务方进行重新确认或者修改方案，评审完成后产品经理一定时间内完成需求定稿。

#### （3）需求排期阶段

需求评审通过后，由团队各职能成员根据上线时间对需求进行排期。确认本项目最终需要交付的需求，各需求分配研发和测试人员，明确各成员所需工时以及提测计划。期间会按需对项目的完成情况进行跟踪，进行项目工作的检查和协调。

#### （4）开发测试阶段

在开发测试阶段，研发和测试人员各自进行编码和测试设计工作。前后端研发在提测计划时间节点前完成全部功能提测。测试人员根据需求输出测试用例并组织用例评审会，测试接到提测任务后，部署环境进行功能测试及对应专项测试等，测试人员按照排期完成该需求的测试工作。

#### （5）验收阶段

在测试任务完成后，测试人员会通知对应产品经理和交互设计师对相关需求进行验收，确保需求按照要求来实现并保证用户视觉效果。根据不同项目的需求，有些项目需

求产品经理也会通知业务人员验收，需求上线之前验收工作完成。

#### （6）发布阶段

需求验收通过后，产品需通知相关团队成员并整理上线需求，对应开发人员发起上线申请，测试人员确认满足发布质量要求后同意发布。生产发布完成后，测试人员和产品经理在生产环境对发布需求进行验证，生产环境验证通过后，即项目交付完成。

根据以上项目开发实施流程，M公司将项目执行过程划分为不同的阶段，前一个阶段工作完成可进入下一个阶段。因此在需求评审阶段完成后，后续的一系列开发和测试工作才会顺序展开，所以对产品需求定义确定项目方向的阶段提出了比较高的要求。同时在项目过程中，业务方可能存在需求变化反馈，因此需求变更也是时长会有的情况。另外，关于项目的相关文档和人员的工作任务安排都是在 Jira 项目管理工具上添加编辑和查看，在每个阶段完成任务后，执行人员可操作完成工单并向下个阶段流转。在所有工单完成关闭后，即项目周期结束可以交付成果，如果项目再有优化和迭代就再次重复该流程。

### 3.1.4 公司项目管理流程中问题观察

上节描述的 M 公司研发中心现行的是瀑布开发项目管理流程，在具体的项目实施过程中存在以下问题。

#### （1）需求描述不清晰

缺乏详细需求说明。一些产品经理只是大致描述了想要实现的产品目标，缺乏对各开发团队需要实现的具体内容和实现方式的阐述。这可能导致团队成员对需求理解存在偏差，开发和测试结果与预期不符。还有一些产品经理使用表格分类需求，但表格中的内容过于简略，很容易引起歧义。尤其是公司业务体系相对复杂，简短的需求描述可能无法覆盖所有场景和异常情况，导致后期需求变更频繁，开发和测试人员难以准确评估工作量。

需求文档不及时更新。测试人员发现，有些功能实现并未在需求文档中描述，可能是以往遗留的需求、本次新增的需求，或是开发过程中临时确定的需求。如果需求文档没有及时更新，会导致开发和测试后期增加沟通成本和工作时间，经常有项目延期的情况。

#### （2）项目过程进度不透明

项目排期缺乏透明度。一方面，项目优先级没有明确说明，导致项目一拖再拖，不

知道具体什么时间上线。另一方面，前后端的排期存在不一致，从而影响交付测试的时间，也会导致延期。

项目进度难以跟踪。首先，由于参与人员众多，任务状态在 Jira 上流转不及时，需要反复与每个相关人员确认实际进度。其次，如果开发任务出现延期，测试时间就会被压缩，测试人员只能在有限的时间内覆盖重点功能，很可能导致上线后问题较多。

### （3）沟通协作障碍

业务人员和产品经理之间存在信息不对称。业务人员提出的需求常常是基于个人视角的点状、发散性想法，未经有效评估，产品经理可能也未能正确理解业务部门的需求。因此在项目执行阶段，业务部门会不断提出新的想法，产生大量需求变更，沟通矛盾不断增加。如果业务部门态度强硬，后期变更增加的成本都需要研发中心承担。

产品和开发测试人员之间存在信息不对称。有些人员在获悉一些最新的项目进展或需求变更的信息，没有及时通知项目组其他成员，导致项目执行到后期就是单纯的沟通确认，存在很多误差。Jira 项目管理工具无法直接查看团队成员的工作进度，需要一一询问，沟通耗费大量时间，影响工作效率。

上述问题是研发中心成员的工作中经常遇到，在以往工作会议中也有经常提出的一些问题，而且工作内容、职位角色不同，发现的问题也就不同，看待问题的视角也不一样。同时，也可能存在一些没有被直接提出或反馈的隐性问题，它们同样影响了研发中心的交付效率和质量。本文的研究需要通过专门的调研来研究问题、分析问题，才能得出结论。尤其是要对团队管理层和各个职能角色的骨干成员进行意见和建议的收集分析，才能更全面的识别和解决问题。因此本章在接下来的部分会对其进行专门的调研和分析，对公司的软件项目管理问题做出更详细的分析总结。

## 3.2 问卷调研收集归纳问题

### 3.2.1 调研设计实施与结果分析

#### （1）调研背景信息

两次德尔菲问卷调研进行时间为 2023 年 5 月至 6 月。

本次调研背景信息为公司现行瀑布项目管理流程下，近半年内交付的大小项目执行情况。近半年内公司业务发展表现良好，交付的项目规模既有业务覆盖面较广的大项目，也有短期的小项目，专家们可以对目前项目管理方法的适用性进行综合考虑。

因为对于公司的大型项目，各位专家成员作为各职能部门的管理者和骨干成员也都

有全程深度参与,所以对于项目计划和实际完成情况、交付成果、过程中存在的问题都特别了解;而对于小型项目,则根据专家成员对自己过往参与交付过的小型项目过程进行回顾评价。

## (2) 设计调研问卷主题

为了准确识别 M 公司研发中心在项目开发管理过程中存在的问题,本研究采用德尔菲问卷调研法发放了 2 轮的问卷进行问题收集,调研问卷详见附录 1。问卷设计思路见表 3.1。

表 3.1 德尔菲问卷设计思路表

问卷设计思路	描述
1.明确调研目的	通过该问卷集思广益,让各位专家给出在各自工作流程中发现的问题。使本次调研范围更为全面且不会出现引导性的问答,本次调研为开放性问卷。
2.设计开放性问题	本次问卷题目为“您认为本公司研发中心软件项目管理中存在哪些亟待解决的问题?这些问题的成因您认为是什么?”。
3.引导详细回答	专家填写时可以参考一下几个方面,比如:项目计划进度方面、需求管理方面、项目变更控制方面、研发流程问题、测试质量方面、人员沟通协作方面、业务方面、项目管理模式方面、项目管理工具方面、组织架构方面等等。
4.提示回答方法	您的回答可以从以下三方面给出: 1.识别当前项目管理过程中遇到的主要问题。 2.探索这些问题产生的原因。 3.征集可能的解决方案。
5.预期结果	各位专家基于个人视角和经验,畅所欲言地指出 M 公司软件项目管理过程中的痛点。这样可以扩展本次调研的研究维度和内容,增加本次调研结果的价值性,也让后续的项目管理优化工作在目标和路径上更加清晰。

## (3) 选取专家成员

本次受访的专家成员是选取在本公司研发中心的管理层和各职能角色的骨干员工组成,涉及整个项目开发全流程。这样能够帮助构建出一个与组织目标和管理层预期相一致的优化策略。

需要满足以下条件:一是行业从业经验不低于 5 年且在 M 公司服务为期 3 年以上;二是长期参与公司的项目研发工作,较为熟悉公司项目管理流程和业务内容;三是角色覆盖全,来自部门管理人员、产品、研发、测试等不同岗位,以确保可以收集到来自不同角色视角观测出的问题。

专家团队成员一共 18 人,包括:1 名首席技术官(CTO)、1 名流程专员和 1 名项目

经理能从管理视角提供意见；2 名研发经理，对研发流程和团队协作经验较为丰富；4 名产品经理，能提供业务提需和产品设计维度的考量；5 名开发工程师，是项目开发过程中的执行角色；1 名测试经理和 3 名测试工程师，能体现质量监控的视角。这样的专家团队可以很大程度上代表研发中心成员，能够提出项目运作过程中较为全面且有效的项目管理意见和对应建议。调研问卷邀请专家成员见表 3.2。

表 3.2 问卷邀请专家成员信息表

专家序号	性别	工作年限	学历	职位
1	男	14	本科	首席技术官
2	女	6	研究生	流程专员
3	男	10	本科	项目经理
4	男	9	本科	前端经理
5	男	12	本科	后端经理
6	男	7	硕士	产品经理
7	女	9	本科	产品经理
8	女	10	本科	产品经理
9	男	7	硕士	产品经理
10	男	6	硕士	后端工程师
11	男	9	本科	后端工程师
12	男	10	本科	APP 工程师
13	女	8	本科	前端工程师
14	男	10	本科	前端工程师
15	男	10	本科	测试经理
16	男	8	本科	测试工程师
17	女	6	本科	测试工程师
18	女	7	本科	测试工程师

### 3.2.2 第一次问卷结果收集整理

在第一次调研中，将问卷标注序号通过线上沟通工具发送给 18 位专家，并提醒专家们每次填写问卷的时间为一周左右。调研问卷发送给专家后，在约定的一周左右时间内收到了回复的问卷结果，回复率 100%。笔者在接下来的一周时间内，对问卷回收结果进行整理、归类。

#### （1）问卷答案初步阅读和标注

在对 18 位专家首次问卷回复的答案进行整理后，首先进行初步的阅读和审查，目

的是从这些回复中识别出反复出现的主题和关注点，并将主题给出分类标注。比如：在阅读过程中，一位专家提出“问题1：需求优先级模糊及难以排期”；另一位也提出“需求管理方面，目前需求优先级评估方法已经近乎失效，流于表面。”等内容，这些专家回复答案旁边可以标注“需求优先级不明确”。

## （2）对标注问题进行整理

在对专家问卷答案的阅读和标注后，本文归纳出了M公司瀑布项目管理流程问题清单，见表3.3。

表3.3 第一次问卷收集专家问题清单

标注问题	专家问题描述
1.需求优先级不明确	需求优先级模糊，需要问一圈领导才能拍板确定，难以排期，降低开发效率。
2.提需缺乏中长期规划	需求单一以业务侧输入为依据，需求管理变动较大，无法做中长期安排。
3.需求文档设计不清晰	需求文档过于简单，不明确细节，看文档不知道在描述什么，会理解偏差。
4.需求变更控制难管理	经常需求变更，变更没有得到有效管控，导致项目延期率偏高。
5.项目过程缺乏有效监控	项目缺乏计划和进度监控机制，而且计划也会变，没有人定期确认进度，经常拖到最后加班赶工期甚至是延期。
6.项目工具适用性有限	Jira 不好看排期，很难直观的将任务、人和时间进行量化呈现。
7.流程风险识别控制不足	项目计划和执行人会变，但是计划变更没有同步做风险预估，有风险了也没人报警。
8.质量管理体系不完善	开发代码质量不达标，测试不充分，或者开发测试时间被压缩导致项目存在缺陷。
9.项目成员沟通协作困难	团队内部以及对其他部门沟通不充分或者不及时，导致信息传递错误。
10.组织人员利用率不高	团队人员都是按照方向固化了同事，任务分配后有人忙不过来，有些人空闲。
11.项目管理和职能管理冲突	人员是按照各职能部门排布，职能部门领导权限太大，会降低成员对项目效率关注度。

通过上述对问卷答案阅读和标注进行归纳，本文将上表前4项问题归为需求管理方面问题，第5-7项归为项目过程方面问题、第8-9项归为研发测试方面问题、第10-11项归为组织架构方面问题。

### 3.2.3 第二次问卷结果收集整理

在进行第二次德尔菲问卷调研时，笔者向原来的18位专家再次发送他们第一次填写的问卷，包括题目和专家自己填写的答案。与第一次问卷不同的是，在第一次问卷调研后所有专家提供的答案被归类汇总和问题标注，整理在附件文档中，随第二轮问卷发送给各位专家。调研问卷详见附录1。

专家们在看完其他人的答案后，如对自己第一次填写的问卷答案有新的想法和意见可对自己第一轮问卷填写答案新增修改和删除作为自己第二轮问卷的答案，以便专家们



更准确地表达他们的观点和意见。

问卷发出一周内陆续收到 18 位专家的回复，回复率 100%。其中有部分专家在第一次问卷汇总结论基础上，对自己填写的问卷结果进行了一些修正补充，也产生了一些新的问题。有些专家未做变更，最后对专家组的意见和建议进行分析、归类整理。新增问题清单见表 3.4。

表 3.4 第二次问卷收集专家新增问题清单

标注问题	专家问题描述
12.项目并行导致资源冲突	多个项目间并行，存在时间和资源上的冲突，这种冲突经常没人知道和协调。
13.项目管理方法不被认可	现在采用瀑布式开发再加上管理工具适应性差，管理成本大且项目风险影响面更大。
14.缺少问题复盘归纳	缺少复盘，开发测试和上线之后的一些问题反复出现，消耗很多沟通和修改成本。
15.开发工作量难以评定	公司项目功能间关系复杂，一个工作任务牵扯的模块很多，常出现工作量被低估的情况。
16.成员权责划分不明确	公司缺少专职人员调整项目排期，有时项目阻塞，当前责任人任务完不成也不去协调，不告知下一个环节责任人，觉得推进这个事情不是自己应该做的。

通过上述对专家问卷新增答案阅读和标注归纳，本文将上表 12 和 13 项问题归为项目过程方面问题，第 14 和 15 项归为研发测试问题、第 16 项归为组织架构问题。

### 3.2.4 问卷结果综合归纳

在两次问卷调研问题收集结束后，本文重新梳理了每个问题表现，和其归属的问题方面。结合第二次次问卷调研中专家对问题回复的变更，笔者对两次调研问卷结果进行综合调整和归纳，决定将第一次问卷问题清单的第 6 项归入第二次问卷新增问题清单的第 13 项。

至此，有关 M 公司软件项目管理问题，总结为 4 个方面共 15 个问题。两次问卷结果综合归纳见图 3.3。

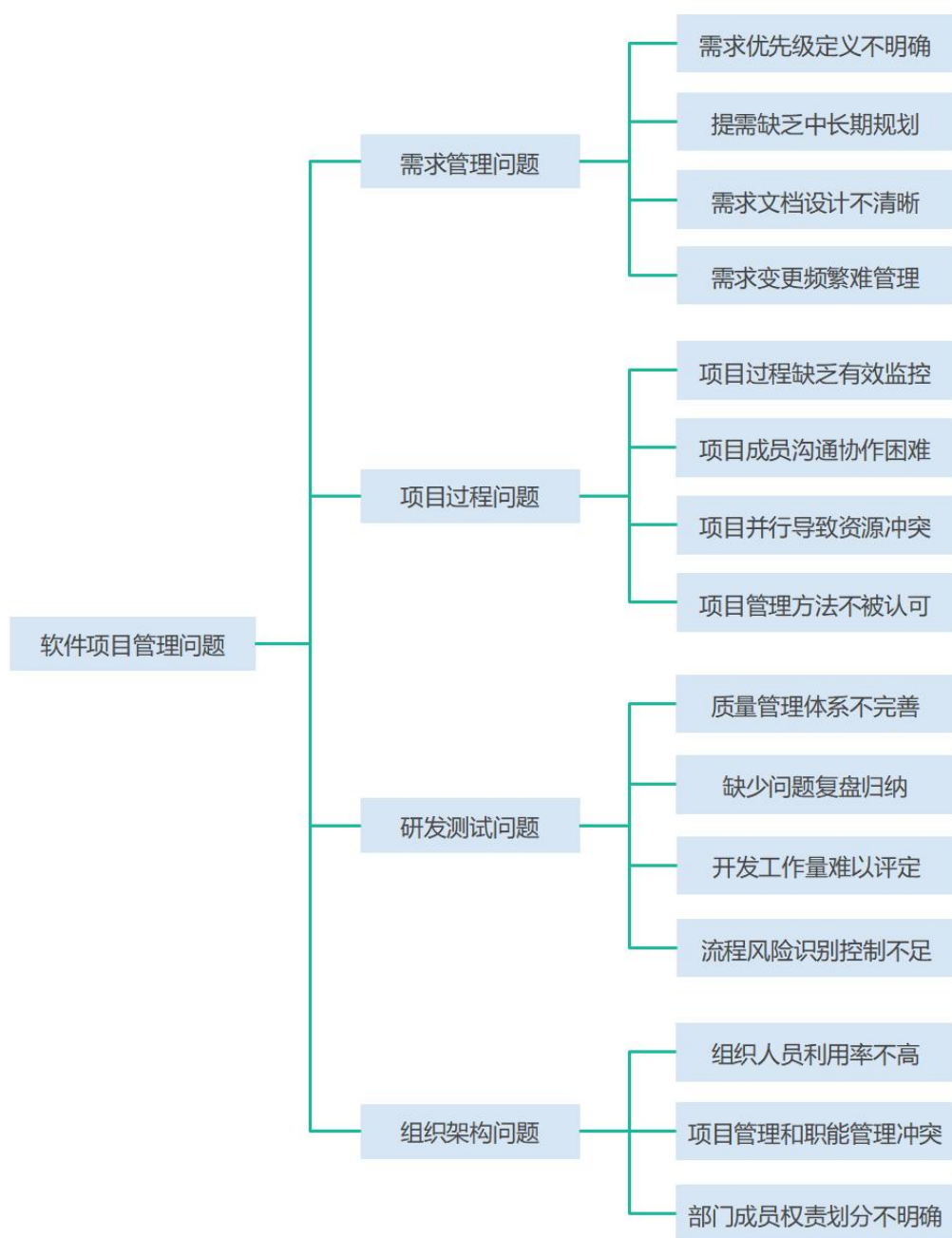


图 3.3 问卷结果综合归纳图

### 3.3 层次分析法确定问题权重

针对上节对两次德尔菲问卷结果的综合归纳，本节会采用层次分析法(AHP)对归纳结果进行量化和比较各个问题的权重。通过这种方法精确地判定各个问题的相对重要性，以及哪些措施可能会产生最大的效果。以此作为判断依据，对公司研发中心项目管理流程优化提出优化方案。

### 3.3.1 构建决策问题判断矩阵

根据对专家问卷结果的综合归纳，可以帮助构建决策问题判断矩阵。首先，决策目标为确定 M 公司软件项目管理过程中存在的问题，即为目标层。然后，归纳的四个问题方面包括需求管理、项目过程、研发测试和组织架构等，即为标准层。最后是方案层，也就是具体的问题表现。

决策问题按照目标层、标准层、方案层形成一个多层次模型。为方便本节对多层次模型进行分析，故对四个问题方面和所属的 15 个问题表现进行编码，编码后的多层次模型见表 3.5。

表 3.5 软件项目管理问题层次结构模型表

目标层	标准层	方案层
M 公司软件项目管理存在问题	需求管理(A)	需求优先级定义不明确(A1)
		提需缺乏中长期规划(A2)
		需求文档设计不清晰(A3)
		需求变更频繁难管理(A4)
	项目过程(B)	项目过程缺乏有效监控(B1)
		项目成员沟通协作困难(B2)
		项目并行导致资源冲突(B3)
		项目管理方法不被认可(B4)
	研发测试(C)	质量管理体系不完善(C1)
		缺少问题复盘归纳(C2)
		开发工作量难以评定(C3)
		流程风险识别控制不足(C4)
	组织架构(D)	组织人员利用率不高(D1)
		项目管理和职能管理冲突(D2)
		部门成员权责划分不明确(D3)

### 3.3.2 调研问卷对矩阵问题打分

#### (1) 本次调研背景信息

本次调研进行时间为 2023 年 7 月初。

基于上一节进行的两次德尔菲问卷调研，为了深入理解并解决在专家问卷中收集的问题，故需要对各个问题的相对重要性进行比较，笔者以此为背景展开此次调研。

调研中引用 T.L.Saaty<sup>1</sup>~9 标度规则，分别对构建矩阵的同层因素进行两两比较，

详细分析每个标准下问题的相对重要性。比较结果通过一致性检验来合成最终的权重并将其整合在一个整体的决策框架中。T.L.Saaty1~9 标度规则见表 3.6。

表 3.6 T.L.Saaty1~9 标度法表

标度绝对值	标度值含义
1	前者因素与后者因素同样重要
3	前者因素比后者因素稍微重要
5	前者因素比后者因素明显重要
7	前者因素比后者因素强烈重要
9	前者因素比后者因素极端重要
2、4、6、8	前者因素比后者因素的重要程度介于上述相邻的中间值
以上数值倒数	两因素的互换位置的比较是原位置比较的倒数

（2）本次调研问卷设计思路

为了计算决策矩阵中的 4 个问题方面和 15 个问题表现的权重排序，本研究再次采用问卷调研法发放 1 次问卷让专家们对问题进行重要性打分，调研问卷详见附录 2。问卷设计思路见表 3.7。

表 3.7 问题评价重要性问卷设计思路表

问卷设计思路	描述
1.明确调研目的	通过该问卷判定各问题方面及对应的问题表现的重要性，以判断优化哪些问题可以获得更好的效果。
2.设计成对比较问题	根据判断矩阵的标准层的问题方面和各自对应的问题表现，将标准层的 4 个问题方面的两两比较设计为 1 个问题，然后将每个问题方面对应的问题表现各设计 1 个问题。
3.明确打分规则	在问卷中提供明确的填写规则和指导包括对标度的解释，包括如何评估各标准层和方案层各因素的相对重要性。
4.介绍回答方法	详细介绍问卷填写的具体步骤，以及如何选择分数反应他们的偏好，可给出具体的示例，帮助专家理解如何有效的进行成对比较。
5.预期结果	各位专家基于个人视角和经验，将 4 个问题方面和 15 个问题表现的给出逻辑一致的评分矩阵，方便笔者可以有效地汇总和分析数据，以确定各问题因素的权重排序，指导决策。

（3）选取专家成员

为了进行本次打分问卷调研，本文从之前进行的两次德尔菲问卷调研中的选出的专家团队中，特别邀请了 10 位成员参与，包括：首席技术官、项目经理、前端经理、后端经理、测试经理、产品经理、APP 工程师、前端工程师、后端工程师、测试工程师各 1 名，以确保本轮问卷评分过程的专业性和可靠性。

### 3.3.3 标准层对目标层的权重计算

通过计算判断矩阵的最大特征值及对应的特征向量，特征向量经归一化处理后可作为各因素的权重。同时需要进行一致性检验，确保成对比较具有逻辑一致性。本文将计算一致性指标(CI)和一致性比率(CR)。如果 CR 小于或等于 0.1，则认为矩阵的一致性可以接受。标准层对目标层矩阵见表 3.8。

表 3.8 标准层对目标层矩阵表

	需求管理(A)	项目过程(B)	研发测试(C)	组织架构(D)
需求管理(A)	1	4/5	5/3	2
项目过程(B)	5/4	1	2	5/2
研发测试(C)	3/4	1/2	1	3/2
组织架构(D)	1/2	2/5	2/3	1

根据标准层对目标层矩阵得出的层次分析结果见表 3.9。

表 3.9 层次分析结果表

项	特征向量	权重值	最大特征值	CI 值	CR 值
需求管理(A)	1.278	29.517%	4.063	0.021	0.024
项目过程(B)	1.581	36.522%			
研发测试(C)	0.866	20.004%			
组织架构(D)	0.604	13.958%			

经计算得出该矩阵 CI 值为 0.021，CR 值为 0.024，因此判断该矩阵一致性检验结果通过。

### 3.3.4 方案层对标准层的权重计算

方案层需求管理问题 A 矩阵见表 3.10。

表 3.10 需求管理方案层对标准层矩阵表

	需求优先级定义不明 确(A1)	提需缺乏中长期规 划(A2)	需求文档设计不清 晰(A3)	需求变更频繁难 管理(A4)
需求优先级定义不明确(A1)	1	2	3/5	3/4
提需缺乏中长期规划(A2)	1/2	1	1/3	2/3
需求文档设计不清晰(A3)	5/3	3	1	2
需求变更频繁难管理(A4)	4/3	3/2	1/2	1

根据方案层需求管理问题 A 矩阵得出的层次分析结果见表 3.11。

表 3.11 A 矩阵层次分析结果表

项	特征向量	权重值	最大特征值	CI 值	CR 值
需求优先级定义不明确(A1)	0.974	22.496%	4.035	0.012	0.013
提需缺乏中长期规划(A2)	0.577	13.335%			
需求文档设计不清晰(A3)	1.778	41.072%			
需求变更频繁难管理(A4)	1.000	23.097%			

经计算得出该矩阵 CI 值为 0.012, CR 值为 0.013, 因此判断该矩阵一致性检验结果通过。

方案层项目过程问题 B 矩阵见表 3.12。

表 3.12 项目过程方案层对标准层矩阵表

	项目过程缺乏有效监 控(B1)	项目成员沟通协作困 难(B2)	项目并行导致资源冲 突(B3)	项目管理方法不 被认可(B4)
项目过程缺乏有效监 控(B1)	1	5/4	4	1/2
项目成员沟通协作困 难(B2)	4/5	1	3/2	2/3
项目并行导致资源冲 突(B3)	1/4	2/3	1	1/3
项目管理方法不被认 可(B4)	2	3/2	3	1

根据方案层项目过程管理问题 B 矩阵得出的层次分析结果见表 3.13。

表 3.13 B 矩阵层次分析结果表

项	特征向量	权重值	最大特征值	CI 值	CR 值
项目过程缺乏有效监控(B1)	1.257	28.444%	4.119	0.040	0.044
项目成员沟通协作困难(B2)	0.946	21.393%			
项目并行导致资源冲突(B3)	0.485	10.982%			
项目管理方法不被认可(B4)	1.732	39.180%			

经计算得出该矩阵 CI 值为 0.040, CR 值为 0.044, 因此判断该矩阵一致性检验结果通过。

方案层研发测试问题 C 矩阵见表 3.14。

表 3.14 研发测试方案层对标准层矩阵表

	质量管理体系不完善 (C1)	缺少问题复盘归纳 (C2)	开发工作量难以评定 (C3)	流程风险识别控制不足 (C4)
质量管理体系不完善 (C1)	1	5/3	5/2	2
缺少问题复盘归纳(C2)	3/5	1	2	3/2
开发工作量难以评定 (C3)	2/5	1/2	1	1/2
流程风险识别控制不足 (C4)	1/2	2/3	2	1

根据方案层研发测试问题 C 矩阵得出的层次分析结果见表 3.15。

表 3.15 C 矩阵层次分析结果表

项	特征向量	权重值	最大特征值	CI 值	CR 值
质量管理体系不完善(C1)	1.699	39.300%	4.032	0.011	0.012
缺少问题复盘归纳(C2)	1.158	26.792%			
开发工作量难以评定(C3)	0.562	13.007%			
流程风险识别控制不足(C4)	0.904	20.901%			

经计算得出该矩阵 CI 值为 0.011, CR 值为 0.012, 因此判断该矩阵一致性检验结果通过。

方案层组织架构问题 D 矩阵表格见表 3.16。

表 3.16 组织架构方案层对标准层矩阵表

	组织人员利用率不高(D1)	项目管理和职能管理冲突 (D2)	部门成员权责划分不明确 (D3)
组织人员利用率不高(D1)	1	3/2	2
项目管理和职能管理冲突 (D2)	2/3	1	2
部门成员权责划分不明确 (D3)	1/2	1/2	1

根据方案层组织架构问题 D 矩阵得出的层次分析结果见表 3.17。

表 3.17 D 矩阵层次分析结果表

项	特征向量	权重值	最大特征值	CI 值	CR 值
组织人员利用率不高(D1)	1.442	45.456%			
项目管理和职能管理冲突(D2)	1.101	34.689%	3.018	0.009	0.018
部门成员权责划分不明确(D3)	0.630	19.855%			

经计算得出该矩阵 CI 值为 0.009, CR 值为 0.018, 因此判断该矩阵一致性检验结果通过。

### 3.3.5 各层因素权重及总排序

在对标准层的因素进行权重计算后, 然后根据标准层的权重对方案层进行加权合成, 最终得到方案层元素相对于总目标的权重, 选择权重较大的方案层因素作为决策优化方向。

根据层次分析法对 M 公司项目管理问题进行定量分析后得到各标准层对目标层的综合影响程度, 以及方案层各因素对其标准层的影响权重, 权重总排序见表 3.18。

表 3.18 权重总排序表

目标层	标准层		方案层			
	因素	权重	因素	权重	综合权重	总排序
软件项目管理存在问题	A	29.517%	A1	22.496%	6.640%	7
			A2	13.335%	3.936%	13
			A3	41.072%	12.123%	2
			A4	23.097%	6.818%	6
	B	36.522%	B1	28.444%	10.389%	3
			B2	21.393%	7.813%	5
			B3	10.982%	4.011%	12
			B4	39.180%	14.310%	1
	C	20.004%	C1	39.300%	7.862%	4
			C2	26.792%	5.360%	9
			C3	13.007%	2.602%	15
			C4	20.901%	4.181%	11
	D	13.958%	D1	45.456%	6.345%	8
			D2	34.689%	4.842%	10
			D3	19.855%	2.771%	14



在采用德尔菲问卷统一专家意见后，对收集的所有问题进行层次分析，得出了 M 公司要重点解决的问题方向。权重较高的标准层在整个决策过程中影响最大，优化这些准则下对应的方案可以获得更好的效果，由上述的各项权重计算可知在 M 公司项目管理问题中，标准层对目标层的重要性权重见图 3.4。

标准层权重

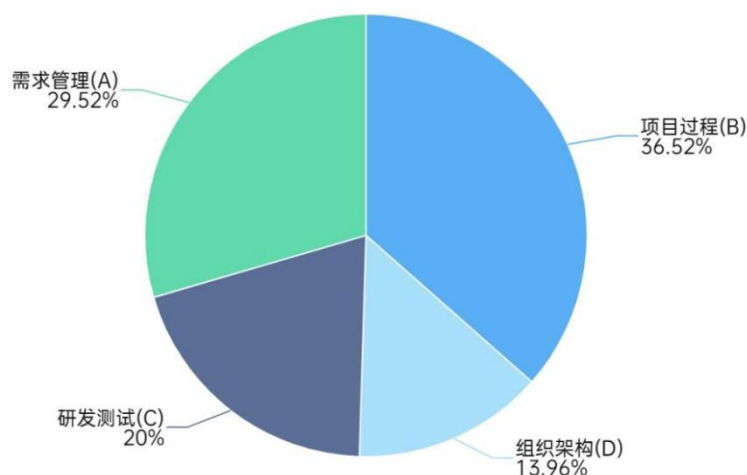


图 3.4 标准层对目标层重要性权重图

经由权重排序总表计算本文得出了方案层对标准层的 15 个问题因素重要性排序，但是考虑到 M 公司时间和客观条件的限制和软件开发项目管理现实决策的需要，本文会将一部分问题因素纳入重点考虑范畴，但是对于一些权重相对不高但是仍有改进潜力的准则及其方案也会适当关注以节省成本。总的来说决策方案会根据层次分析的结果，通过主次结合、平衡兼顾的策略制定出一个有效度高和可操作性强的优化方案。根据成熟度提升(Maturity Improvement)理论，即通过测评判断项目管理过程的成熟水平，找到优化空间，制定对策提升项目和组织的管理能力，以达到行业优秀水平。根据本文的研究与实践经验，成熟度提升重点解决方案的最佳选择区间为问题总数的 50%-70%。

因此本文将方案层权重排序的前 10 个问题因素纳入重点考虑范畴，而后 5 个问题因素会在后面的优化方案设计中适当关注。即方案层重点考虑问题因素包括：需求优先级定义不明确(A1)、需求文档设计不清晰(A3)、需求变更频繁难管理(A4)、项目过程缺乏有效监控(B1)、项目成员沟通协作困难(B2)、项目管理方法不被认可(B4)、质量管理体系不完善(C1)、缺少问题复盘归纳(C2)、组织人员利用率不高(D1)、项目管理和

职能管理冲突(D2)。前10位问题因素具体权重值见图3.5。

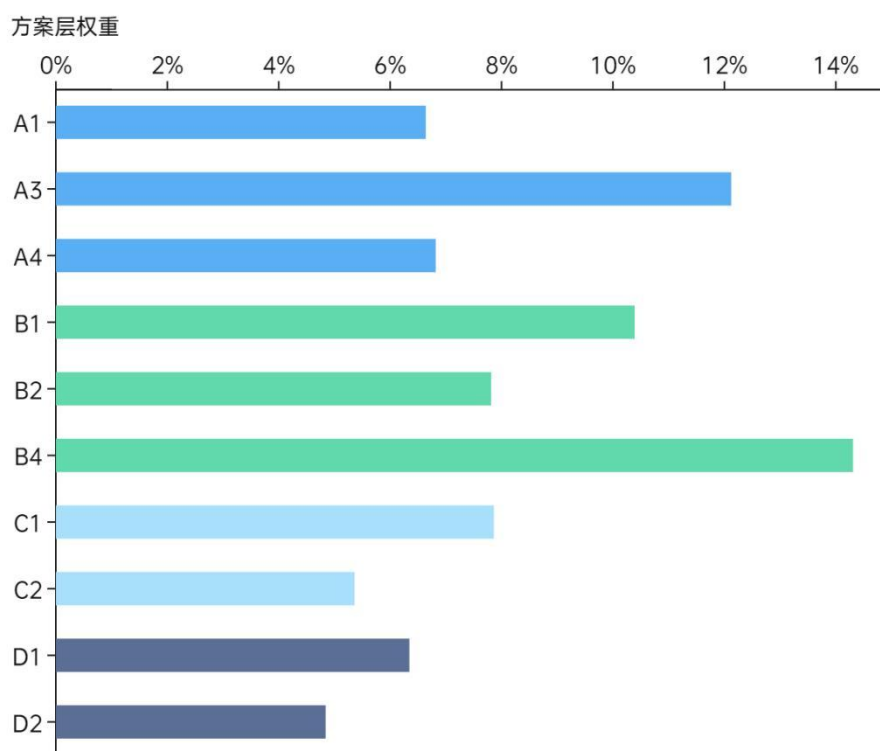


图 3.5 方案层重点考虑因素权重值图

注 a: 本图展示权重分布按照矩阵的4个问题方面进行分类。展示顺序并非排序结果，各问题因素已根据先前分析在权重总排序表中。

### 3.4 核心问题总结和原因分析

首先，在需求管理(A)方面，需求优先级定义不明确(A1)、需求文档设计不清晰(A3)、需求变更频繁难管理(A4)这三个问题表现所占权重相对较高，经笔者对问卷数据的分析总结，这三个关键因素综合导致了M公司在需求管理方面存在偏差，需要采取有效措施来优化改进。

第二，在项目过程方面，项目过程缺乏有效监控(B1)、项目成员沟通协作困难(B2)、项目管理方法不被认可(B4)这三个问题表现所占权重相对较高，经笔者对问卷数据的分析总结，这三个关键因素综合导致了M公司项目过程控制效率有待提高，需要采取有效措施来优化改进。

第三，在研发测试方面，质量管理体系不完善(C1)、缺少问题复盘归纳(C2)这两个问题表现所占权重相对较高，经笔者对问卷数据的分析总结，这两个关键因素综合导致了M公司研发测试工作质量需要改善，需要采取有效措施来优化改进。

最后,在组织架构方面,组织人员利用率不高(D1)、项目管理和职能管理冲突(D2)这两个问题表现所占权重相对较高,经笔者对问卷数据的分析总结,这两个关键因素凸显了M公司组织协调机制存在不足,需要采取有效措施来改进。

综上,本文通过对问题的详细拆解,总结出M公司存在业务需求管理存在偏差、项目过程控制效率有待提高、研发测试工作质量需要改善、组织协同机制存在不足等4个待优化的方向。但是关于优化方案的设计,本文需要对这4个方向15个问题进行原因识别,并针对性给出具体的优化方案,旨在系统解决M公司软件项目管理问题。

### 3.4.1 业务需求管理存在偏差

在需求优先级定义不明确(A1)、需求文档设计不清晰(A3)、需求变更频繁难管理(A4)方面,专家们对于这三个问题的原因分析见表3.19。

表 3.19 业务需求管理存在偏差原因总结表

问题因素	对应问题原因分析
需求优先级定义不明确(A1)	没有切实有效的优先级评估方法
	业务部门提出需求较分散缺乏规划
	不同业务团队优先级不同存在矛盾
需求文档设计不清晰(A3)	需求文档输出不完整或太简单
	产品经理和业务人员沟通不充分
	没有严格的需求编写规格要求
	产品经理文档表达能力差异
需求变更频繁难管理(A4)	业务方前期对需求考虑不充分
	存在其他临时插入需求
	需求评审时各方对细节关注不够
	开发人员对需求理解偏差

综上,关于公司业务需求管理存在偏差的问题,本文的优化思路为:对于需求优先级要根据需求的价值判断,价值更高的优先做。产品部门需要平衡需求资源竞争,避免各业务部门主观确定需求优先级和抢占资源<sup>[34]</sup>;制定需求基线标准、提高产品经理文档表达能力,建议产品部门内部交叉审和预审来改进需求文档设计不清晰的问题;对于需求变更需要建立变更管理流程,并通过工具落地,不管是内外还是外部的变更都要通过流程体现,通过项目管理人员审核确认,变更带来的影响要尽可能量化留档。

### 3.4.2 项目过程控制效率有待提高

在项目过程缺乏有效监控(B1)、项目成员沟通协作困难(B2)、项目管理方法不被认可(B4)方面,专家们对于这三个问题的原因分析见表 3.20。

表 3.20 项目过程控制效率有待提高原因分析表

问题因素	对应问题原因分析
项目过程缺乏有效监控(B1)	Jira 难以观测项目流程进度
	项目管理流程和工具培训欠缺
	缺少专职人员协调项目计划和执行差异
项目成员沟通协作困难(B2)	业务、产品和开发之间的工作期望矛盾
	缺乏团队建设和培训
	缺少有效的沟通渠道
项目管理方法不被认可(B4)	成员沟通协作能力不足
	现行项目管理方法成本大
	缺少合适的项目管理工具

综上,关于公司项目过程控制效率有待提高的问题,本文的优化思路为:引入敏捷开发管理改进工作流程,结合可视化看板工具进行项目全过程平台管理,同时搭配敏捷项目经理完成“人+看板”的可视管理方式。明确成员沟通机制,比如定期的项目会议和培训<sup>[35]</sup>、对齐成员的沟通内容,通过对项目整体过程的优化提升团队协作能力。

### 3.4.3 研发测试工作质量需要改善

在质量管理体系不完善(C1)、缺少问题复盘归纳(C2)方面,专家们对于这两个问题的原因分析见表 3.21。

表 3.21 研发测试工作质量需要改善原因总结表

问题因素	对应问题原因分析
质量管理体系不完善(C1)	代码和测试用例评审缺失
	开发自测不到位
	没有按需召开技术交流会
	开发测试排期时间被压缩
缺少问题复盘归纳(C2)	问题反复出现但没有规避措施
	项目上线后的用户反馈知晓不多
	没有形成成熟的知识库文档

综上，关于公司研发测试工作质量需要改善的问题，本文的优化思路为：在项目管理过程可以得到优化的基础上，研发过程中需要提升代码质量，重视代码评审和提高团队协作能力。测试过程中有效执行测试计划，提升测试自动化运行能力来分担一些重复性的测试工作，提升测试效率<sup>[36]</sup>。另外对于开发和测试人员都需要进行定期培训和问题复盘归纳，问题分类针对性提出解决方案，到后期要看出问题减少的趋势，并形成技能知识库文档，方便成员后期回顾和能力提升。

### 3.4.4 组织协同机制存在不足

在组织人员利用率不高(D1)和项目管理和职能管理冲突(D2)方面，专家们对于这两个问题原因分析见表 3.22。

表 3.22 组织协同机制存在不足原因分析表

问题因素	对应问题原因分析
组织人员利用率不高(D1)	业务提出需求缺少统一收口角色
	项目组成员职责未充分调动
	项目重大决策缺少角色承担
	职能角色固定无法按需流动
	缺少专职项目管理人员
项目管理和职能管理冲突(D2)	职能管理决策权限过大
	没有针对项目设置绩效考核
	项目与职能部门资源分配冲突

综上，关于公司组织协同机制存在不足的问题，本文的优化思路为：明确项目管理规范，可以成立项目管理部门来管理协调项目经理，为项目管理过程制定标准和规范<sup>[37]</sup>。并建立职能部门人力资源池为各项目协调人员，在此基础上从项目管理和职能管理两个角度双向考虑项目经理和其他员工的绩效激励机制，以提高人效。

## 第4章 M公司敏捷项目管理流程优化方案设计

根据第三章的数据分析结果和问题原因总结,本文得出了业务需求管理存在偏差、项目过程控制效率有待提高、研发测试工作质量需要改善、组织协同机制存在不足等四个待优化的方向。

为有效提高软件开发效率及用户满意度,本章将运用敏捷开发模型看板(Kanban)方法,结合M公司软件交付的目标和实际状况,从组织协同机制入手完善部门架构和人员配置,然后定制项目过程控制规范、优化需求管理办法,最后全面提升研发测试工作质量。最终以这样的实施逻辑顺序来制定一个整体的优化方案,以此确保解决方案的连贯性和可行性。

公司一直以来用Jira作为项目管理工具,但Jira的可视化交互体验不佳,导致团队成员难以了解和把控项目整体进度和其他诸多问题。因此,基于对看板管理的实际需求和成本考量,公司决定采用飞书看板作为流程优化后使用的可视化项目管理工具,以此构建适合M公司的软件开发项目管理流程,促进组织项目管理能力的整体提升。

### 4.1 组织协同机制优化方案设计

#### 4.1.1 组织架构敏捷化

##### (1) 成立项目管理办公室(PMO)

由于M公司在项目管理和职能管理方面存在一些冲突,导致项目管理权限相对较小,各项目推进过程出现问题时无人进行统筹,而是要与各职能管理人员协调,沟通效率低并且人员配合度不高,因此需成立项目管理办公室来承担统筹工作,可以较好的解决两者管理冲突问题。

项目管理办公室负责对组织内部所有的项目进行持续治理和监控,关于项目重大事项的决策机制,会由项目管理办公室负责制定标准和规范<sup>[38]</sup>。包括及时发现偏差、启动纠正措施、终止失去价值的项目。项目管理办公室还需促进项目之间进行经验教训分享,使项目团队少走弯路,提高组织整体效率。具体包括输出项目管理模板、沉淀项目数据库、项目绩效评定、项目经理的工作指导、定期进行项目经理复盘和经验分享。

##### (2) 引入敏捷项目经理角色

目前M公司在具体的研发项目缺少专职项目管理人员,导致一些项目重大决策缺少角色承担,增加项目人员的沟通成本。因此,可在项目管理办公室对项目概况进行评

估后，为该项目配置一名项目经理。

项目经理可从团队领导和骨干成员中酌情选择，项目经理人员的安排由项目管理办公室负责指定和评估绩效。产研中心会划拨一定的项目激励经费，用来奖励优秀的项目经理或者项目成员。项目经理在项目中负责规划、执行、监控、控制和收尾项目，需要组织和领导项目团队，并负责项目过程的整体管理，以确保项目能按照预期的目标、范围、时间和质量要求完成。

#### 4.1.2 构建敏捷看板人力资源池

M 公司通过建立一个统一的人力资源池，并通过飞书可视化看板实施资源分配可以构建一个相对高效的项目管理动态资源调配机制<sup>[39]</sup>。研发中心目前的人员都是按照职能来设置的，每个职能部门都形成了一个资源池。在有项目的时候会进行资源匹配，以此实现跨部门、跨项目的资源动态配置，同时需要赋予项目管理者在资源调度与优先级确定方面的主导权力。首先，职能部门资源管理者需要在项目规划阶段有效参与，为项目提供关于资源需求、时间线安排和目标设定的重要输入。

与此同时，职能部门管理者需要考虑到员工之间的沟通成本问题，开发团队按照后端、前端、移动端等技术线进行分割。产品和测试人员则可以基于业务方向进行相对固定的分组，在有大型项目的时候或者特殊情况再从职能部门抽调。这样可以避免人员频繁更换，降低沟通成本。由于成员相对固定，每个人的工作习惯和能力也可以相互了解和适应，协调沟通更加顺畅。即使偶尔有人员变动，新人也可以更快融入团队。当然具体的分组方式和人员安排，还是需要结合实际情况酌情考虑。

通过在飞书看板管理工具上创建组织级的共享资源池，项目团队和职能部门能够实现人力和其他关键资源的可视化管理和动态调配。此外，看板作为一个实施资源共享和项目协调的有效工具，提供了一个平台，使项目管理者能够实时监控资源状态和团队工作负荷，同时促进了项目团队与职能部门之间的沟通和协作。通过在平台上进行任务分配、进度跟踪和实时沟通，团队成员可以更加高效地共享信息和协调工作，进一步提升项目的执行效率。

比如在一个优先级较高的项目中，因为变更或其他原因需要增加开发人员或需要别的部门成员来协助完成。项目经理通过看板管理平台发现，另一个项目组 B 的一个关键阶段已经顺利完成，该项目中的一名具备所需技能的开发人员即将有更多的可用时间。项目经理 A 与项目经理 B 和职能部门进行沟通，一致同意暂时将这名开发人员从项目 B

调配到项目 A 中，以帮助解决人力短缺问题。这名开发人员的工作负荷和任务分配在看板项目管理平台上进行更新，确保所有相关人员都能看到最新的资源分配情况。这种调配不仅解决了项目 A 的需求，也保证了人力资源的高效利用，避免了在项目之间出现资源浪费或闲置的情况。项目完成后资源可以根据新的项目需求和优先级再次进行动态调配。通过这样的资源调配机制，公司能够灵活应对项目需求的变化，优化资源利用，加快项目交付速度，从而提高整个组织的工作效率和项目成功率。

4.1.3 引入敏捷项目绩效考评机制

为了在 M 公司内提升项目团队与职能部门之间的协调与合作，同时确保组织资源的优化配置与业务目标的顺利实现，研发中心需要完善与项目成果紧密挂钩的组织目标和考核体系。首先，需要制定与项目成果直接相关的考核指标体系。研发中心需要明确设定能够反映项目质量、进度和效率的关键绩效指标，并将这些指标与项目团队及项目经理的绩效考核直接挂钩。通过形成对项目目标的共识，确保所有团队成员都明白自己的工作如何与组织的最终目标相联系，这样不仅可以提高团队的目标导向性，还能促进成员之间的协作和责任感。

具体来说，项目管理办公室可以引入项目团队与项目经理双重绩效考核机制，能够更细致地评估团队协作和个人贡献度。这种双重考核机制旨在鼓励团队成员积极参与项目工作，提升他们对目标的贡献度<sup>[40]</sup>。因此组织需要优化现有的激励考核体系，确保考核指标既能反映出部门的整体目标，也能与项目的具体目标相匹配。项目绩效评估由负责该项目的流程部责任人发起，评估对象包括项目本身、项目经理以及项目成员，项目绩效评估维度包括：项目交付准期率、项目范围确定性、过程管理能力、项目质量和难度等，具体内容见表 4.1。

表 4.1 项目绩效评估表

指标	说明	计算方式	权重
交付准期率	立项之后的时间点是否按时上线	5 分：按时上线或者提前上线；4 分：延期上线；3 分：其他。	30%
项目范围确定性	是否发生过变更，变更的量是否可控	5 分：项目未发生变更；4 分：发现在可控范围内问题，对上线无影响；3 分：发现未识别到的内容，对交付产生较大影响。	10%
过程管理能力	所有节点是否正常流转，业务部门是否知晓项目的进度	5 分：所有节点按时流转；4 分：节点流转略延迟；3 分：其他。	15%



续表 4.1 项目绩效评估表

指标	说明	计算方式	权重
项目质量	项目规模和缺陷数量的关系	5 分：快速决策，当天问题当天解决；4 分：依赖他人决策，或者 48 小时内决策完成；3 分：无决策问题，或者决策效率较低。	15%
项目难度	总评分：参与人员数量	5 分：100 人天以上，参与人员大于 15 人；4 分：60 人天以上，参与人员大于 10 人；3 分：其他。	30%

同时对项目经理实施 360 评分方案，考察领导力、团队合作、沟通能力、专业技能和工作态度等方面。这种全面的评估机制确保了评价的客观性和公正性，为项目经理和团队成员提供了全方位的反馈，帮助他们明确优势和改进方向。项目干系人、项目经理和项目成员的参与，保证了评估结果能够全面反映项目的实际情况和团队成员的实际表现。360 评分方案见表 4.2。

表 4.2 360 评分方案表

评分维度	评估标准
领导力	项目经理对项目的整体管理和掌控能力，包括对项目目标的清晰规划、团队的协调和沟通、问题的解决能力等。
团队合作	项目经理与团队成员的合作关系，包括对团队成员的指导和支持、团队合作的氛围和成效等。
沟通能力	项目经理的沟通能力，包括与项目相关者、团队成员、上级领导之间的有效沟通和理解能力。
专业技能	项目经理的专业技能和经验，包括对项目管理流程和方法的掌握、对项目风险和问题的识别和应对能力等。
工作态度	项目经理的工作态度和责任心，包括对工作的认真负责、对团队成员的关心和支持、对项目目标的执着追求等。

最后，项目绩效奖励机制可以较好地激励成员积极性。对于在项目中表现优异的团队和个人，项目管理办公室应给予明确的奖励，包括但不限于金钱奖励、职业发展机会或其他。项目经理的晋升和加薪会根据他们负责过的项目的规模、难度以及项目绩效来进行考评。这种激励办法是要奖励能成功管理复杂项目、带领项目团队解决问题和较好的实现项目目标的项目经理。

同时项目管理办公室会对应给出一定的项目激励经费，专门用于奖励表现优秀的项目经理和项目团队成员。项目经理可以通过这个奖励机制来激励项目组成员，提高团队

的整体表现和项目的交付质量。M 公司通过实施这种激励方案，可以有效地提升职能部门人员的工作积极性，提高项目团队与职能部门之间的协作效率，促进业务目标的达成。

4.2 项目过程控制优化方案设计

M 公司业务部门现阶段对于很多运营转化类业务需求，更倾向于整体迅速交付并且团队成员可以一目了然看到项目执行过程，而且在此过程中要积极应对变化以快速实现转化目标，因此希望构建一个更具有灵活性项目管理工作流程<sup>[41]</sup>。为了更好地适应这些需求，本文的优化方案是引入可视化看板，强调项目时间弹性规划。对于上线时间根据任务量预估确定，如果因为其他事情插入导致延期，由其他人协作分担或者提前告知延期即可。这一改进充分利用了看板的灵活性和适应性，使团队能够快速响应紧急或高优先级的需求。

方案设计旨在满足业务部门对于提高效率和质量的期望，核心目的是赋予项目团队管理工作量的能力、定期交付业务价值、有效解决工作瓶颈，以及最大限度地提高项目的交付质量和满意度。通过这种方式使产研团队能够更好地平衡大小型业务需求的多元性、复杂性与面对变化需求的快速反应能力，进而提升整个项目管理的灵活性和效率。

4.2.1 定制敏捷看板工作流程

根据项目需求用敏捷看板划分阶段并定制工作项<sup>[42]</sup>，包括创建项目、创建需求、任务分解、记录缺陷、上线发布等，以适配 M 公司的项目管理过程。项目-需求-任务层级关系见图 4.1。

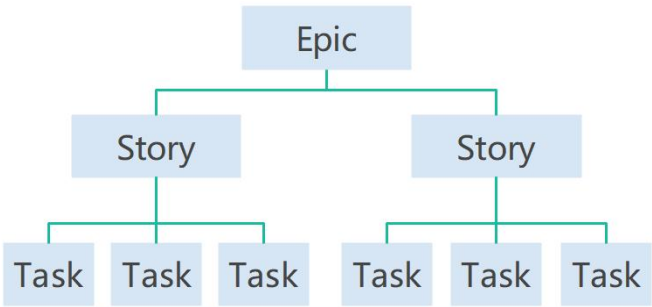


图 4.1 项目-需求-任务层级关系图

(1) 看板创建项目(Epic)

项目即为项目目标，通常需要拆分为多个需求来具体实施。业务需求分常规优化类需求和需要立项的需求两个类别，且支持取消和驳回。立项需求流程主要面向对业务影

响大、需要重点推进的核心需求。通过这种方式,团队可以保持灵活性,适应需求变化,同时确保项目的可管理性;流程关注可行性分析、立项管理、项目执行控制等关键环节。总体目的是全面评估需求实施的可行性,有效管控项目风险,确保需求顺利上线。常规需求流程主要关注需求工作量评估和研发交付节奏,由提需人、部门负责人、产品等共同参与需求处理即可,无需进行复杂的项目化管理。项目立项流程节点见图 4.2。

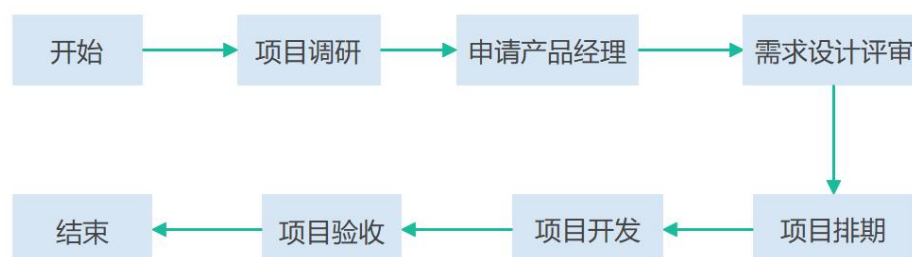


图 4.2 项目立项流程节点图

## (2) 看板创建需求(Story)

需求即为用户故事,通常分产品需求和技术需求,由项目分解产生,且支持取消和驳回。工作流程为创建需求、需求和交互设计评审、开发和用例评审、测试、验收发布。需求评审和 UI 交互稿评审同期进行,在需求确认后,可以准备技术方案评审,开发收尾阶段进行测试用例评审等等。配置此工作流程的目的是在节约时间的同时,也同样重视过程管控,从而有效平衡进度、质量与风险。分阶段充分利用时间实施各项工作内容,防止后期返工和修复,确保顺利上线。

## (3) 看板任务分解(Task)

任务需要分解至最细的工作项,由需求分解产生,分角色拆分为开发和测试任务。工作流程为创建任务、任务待办、任务处理中、任务完成。期间记录开始和结束时间以及是否延期及延期时长和任务间依赖,预估工作量和实际投入工作量并归档任务交付物。该流程的核心在于将任务分解至可操作的最小工作单元,以提高掌控度,同时设置状态流转机制,实时跟踪任务进展情况<sup>[43]</sup>。配套的数据收集可以发现问题和改进方向。任务交付物的归档也能够重复利用,任务顺利完成的同时提高数据化管控水平。

## (4) 看板记录缺陷(Bug)

测试人员记录发现的问题,并根据发现环境区分线上缺陷和线下缺陷。工作流程为创建缺陷、处理缺陷、回归缺陷、关闭缺陷。该流程专注于测试缺陷的生命周期管理,通过连贯的缺陷处理流程,可以高效解决测试过程中发现的问题。而数据的收集可以量

化工作流程的执行效率、质量状况，并关联至前端需求形成闭环。目前制定缺陷等级判断标准为：严重、重要、一般、优化四个等级。除缺陷处理外,还会收集项目各维度数据，并反馈至项目相关者和质量部经理，通过测试结果汇总分析促进产品质量提升。测试 Bug 处理流程见图 4.3。

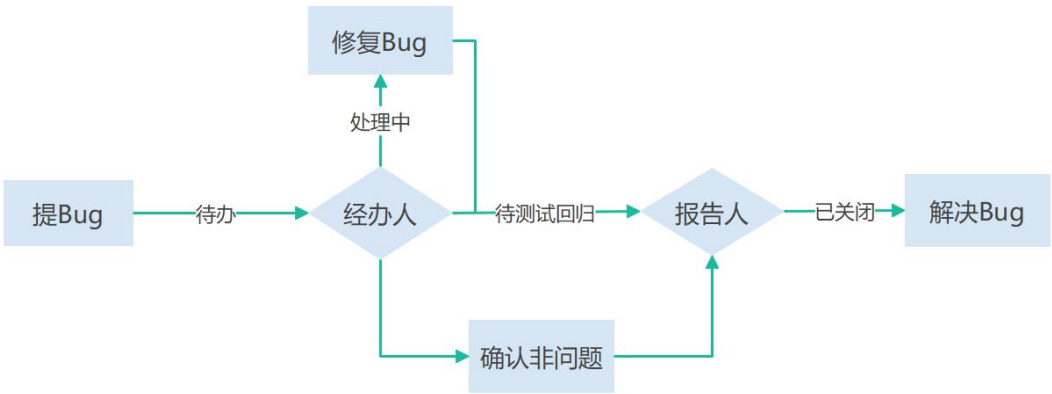


图 4.3 Bug 处理流程

（5）看板上线发布

上线发布用于上线流程跟踪。上线申请工作流程为创建上线申请、项目相关者确认、发布、关闭。上线申请聚焦于征求干系人意见，结合实际发布执行时创建的发布单，两者共同完成产品上线。上线申请与发布单的配合流程,明确了职责划分。申请侧重确定发布合理性,发布单保证技术交付的正确性。流程数据的收集可以分析资源投入与过程效率,而失败统计则可资源作为改进参考，目的在于使产品顺利上线。项目上线发布流程见图 4.4。

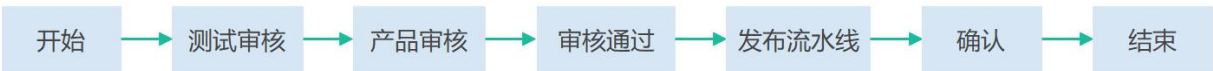


图 4.4 项目上线发布流程图

4.2.2 敏捷看板实现进度可视化

（1）看板查看项目状态

飞书看板能直观展示 M 公司产品研发中心不同阶段的项目状态，可以建立包括项目数据统计和项目状态看板、进展甘特图等展示面板。故事墙通过卡片化任务分解步骤，项目数据统计则从全局角度观察剩余工作量。两者相互配合，使团队成员能够清晰看到自己和他人的工作进展，并主动发现与进度相关的问题。与此同时，设计、开发、测试等角色之间的工作互为影响。通过看板项目人员可以观察彼此进度，有利于推动相关任

务尽快完成，从而整体提升工作效率。

相比之前过于依赖 Jira 面板的管理方式，飞书看板的直观性与灵活性，可以有效激励团队成员主动参与项目管理。本文希望通过这种可视化方法，使项目管理更高效<sup>[44]</sup>。

项目看板示例见图 4.5。

□	🔒 项目名称	👤 总负责人	👥 成员	🎯 目标	🕒 状态
▼	进行中	2 条记录			
1	优化产品功能	👤 于小宁	👥 黄泡泡, 于小宁	功能留存提升 4%	进行中
2	创新功能探索	👤 雷小达	👥 雷小达, 邹小杉	功能渗透率达 8%	进行中
▼	未启动	1 条记录			
1	完成用户拉新	👤 王小铭	👥 王小铭, 万小奇	日新增用户数提升 2 万	未启动
▼	已完成	1 条记录			
1	提升用户体验	👤 周北北	👥 周北北, 沈小茜	用户满意度提升 20%	已完成

图 4.5 项目看板示例图

注 b: 图片来源-飞书项目看板

## (2) 看板展示不同任务进度

看板视图展示不同任务的进度，如“待处理”、“进行中”、“已完成”等。团队成员可以直观地看到每个任务的当前状态，可以拖拽卡片在不同状态之间移动，以更新任务进度。看板视图提高了项目的可视化管理效果，使得工作流程更加透明，便于团队成员实时了解和更新工作进展。

甘特视图显示了每个任务的详细信息，包括描述、负责人、排期等。成员可以根据需要定制列，筛选特定的任务（如按优先级、状态、负责人等），并对任务进行分组和排序。它可以清晰地显示任务间的时间关系和依赖性，帮助项目管理者更好地规划和调整时间表。甘特视图适合于进行时间管理和进度规划，特别是在需要理解任务间复杂依赖关系和时间约束的情况下。

## (3) 看板多维度视角分类

M 公司看板可分为业务视角、产研中心视角、产研职能部门视角、个人视角。比如，业务视角可查看对应的需求及进展情况；研发中心视角可查看：所有需求的状态及进展，风险项，人员投入情况；职能部门视角可查看：人员投入情况、人员排期情况、工作项数量及处理情况、延期及变更事件；个人视角：待办工作项、到期前预警、延期告警，

参与的项目及评分情况。看板个人视角任务示例见图 4.6。



图 4.6 看板个人视角任务示例图

注 c：图片来源-飞书项目看板

4.2.3 明确敏捷项目经理职责

项目经理负责确保 M 公司产品研发中心敏捷实践的正确实施，包括监控工作流程和进度、协调团队成员、明确任务的优先级和维护敏捷文化。项目经理是项目成功交付的第一责任人，是项目中负责规划、执行、监控、控制和收尾项目的人员。项目经理需要组织和领导项目团队, 并负责项目过程的整体管理, 以确保项目能按照预期的目标、范畴、时间、成本和质量要求完成<sup>[45]</sup>。当然，项目过程中出现一些变化在所难免，需要交付的功能可能在过程中重新安排优先次序，这需要项目经理具备更大的灵活性和反应能力。

M 公司产品研发中心目前的人员都是按照职能来设置的，每个职能部门都形成了一个资源池。在实际工作中，会根据项目实际情况来匹配合适的项目经理。项目经理工作流程见表 4.3。

表 4.3 项目经理工作流程表

序号	工作流程	备注
1	委派项目经理	产品方案或技术方式评审通过后，由项目管理部门成员委派项目经理，并上报公司决策。
2	需求跟踪	项目经理按需申请产品经理，在产品经理完成需求后粗排任务，并将任务分配到具体的同事，之后组织需求评审会议。



续表 4.3 项目经理工作流程表

序号	工作流程	备注
3	立项启动	项目经理与流程部确认立项通过后，发出立项通知，组织项目成员进行项目启动会议，并精排任务。
4	项目执行	项目经理需定期跟进项目进度，确保任务按计划执行和协助处理延迟问题，并进行周度项目进展汇报。
5	项目验收发布	项目经理在测试任务完成后组织项目验收并输出验收报告，之后跟进发布上线流程。
6	结项汇报	项目完成后可视情况组织项目复盘会议，并交付结项汇报。汇报内容包括但不限于项目过程执行概况、出现的变更、重大问题和项目成果评估等内容。

#### 4.2.4 促进团队成员沟通协作

##### （1）明确沟通机制

实施明确的沟通机制及对应的责任分配。这包括沟通工具、方式方法、统一且易于理解的语言描述等。需要明确由谁负责明确和传达信息、何时传达以及通过什么渠道传达，并确保所有团队成员都了解并遵守这一流程。同时，需要为信息传达建立明确的责任分配机制，每位团队成员都将清楚自己在信息链中的角色和职责，在哪个流程中出现沟通信息差需明确由谁牵头与责任人沟通确认汇总结果，避免后续出现责任推诿和问题遗留，这样可以较好的降低项目过程中的沟通信息不对称的情况。

定期或不定期的会议。这可以使各职能和项目团队间交流业务进展、存在的问题以及需要协助的事项，确保所有员工都参与讨论并分享自己的观点和建议，在项目目标分配和进度管理等方面有共同的理解和协同<sup>[46]</sup>。一般指定项目经理或者职能部门领导作为会议责任人形成一个有效的沟通桥梁，可以帮助解决项目团队和职能部门之间的疑问和冲突。另外最重要的一点是明确沟通的结果，要达到沟通的目的，比如项目复盘会议过程的会议纪要，在会后需要由项目经理及时整理至对应知识库或发送相关人员会议邮件，并督促会议形成结果的执行，比如会议决议里提出需要开发人员组织技术方案评审以进一步跟相关人员明确产品开发方向，这需要项目经理进行会后跟踪执行。

非正式交流。因为正式会议耗费时间，有时候成员时间很难对齐，所以日常的非正式交流可以通过飞书即时消息工具或者简单的面对面交谈来实现<sup>[47]</sup>，因为环境相对宽松，沟通双方心理负担小所以解决问题相对比较快。线上双方沟通过程留痕也比较方便后期追溯，另外建立相应的项目沟通群，如果项目过程中有需要消息同步可在消息群公布确保相关人员知悉。但是因为每个团队成员理解能力不同，线上的消息和文档内容仍

然可能存在理解偏差,所以有些情况下线下面对面演示或者沟通效率可能会更好。

### (2) 明确沟通的议题

沟通议题需要同步给所有团队相关人员,这样可以确保每个人都对沟通的目的和预期结果有明确的理解。发起沟通的人需要明确指出沟通的目的即希望解决的具体问题,这有助于使沟通更加目标导向和高效。在完成有效沟通后,确保所有讨论的问题已经得到明确的解答,且相应的解决方案已经制定<sup>[48]</sup>,接下来的关键动作是确认和记录沟通成果。

这样的做法确保了沟通的清晰性和后续行动的可追溯性。通过留下详细的沟通记录,团队为未来可能出现的相似情况或需要回顾的场景提供了参考资料。这些记录不仅是对当前沟通成果的总结,也是以后沟通和决策的重要依据,有助于保持团队内部信息的一致性和项目进展的连续性。

### (3) 对齐所有沟通信息

在确认沟通目的后需要对齐背景信息。即针对当前要沟通的事情,比如产品需求变更就要确保所有参与沟通的人是在同一个起点,并且找到可以依据的材料,比如会议记录、文档、之前沟通的聊天记录等等。针对项目开发中常见的几个场景,当涉及到产品与研发的实现细节时,评审过的产品需求文档和交互设计图应作为沟通的首要前提。如果有任何口头上的需求变更都应及时记录下来并更新到相关文档中,以避免信息的遗漏。

同样在产品经理、开发人员与测试人员的测试细节沟通中,测试用例也要充分考虑,用例评审时最好通过场景化的描述来完善用例,确保所有参与评审的人员对用例有深入的理解。对于技术设计与代码实现的细节沟通,架构设计和技术详细设计文档是重要的依据。这些文档应清晰定义实现的框架和技术细节,为团队提供了共同的理解基础。但是文档维护可能难以跟上需求和开发的快速变化,这就要求沟通者在沟通时必须仔细描述问题,确保所有参与方对问题有一致的理解。

## 4.3 业务需求管理优化方案设计

### 4.3.1 需求管理敏捷化

#### (1) 敏捷看板明确需求优先级

解决需求优先级不明确问题需要制定明确的优先级设定标准,产品和业务部门在制定标准时,需要创建一个统一的框架来明确哪些因素会影响需求的优先级,例如业务价



值、业务紧急程度、资源可用性等。最后，项目优先级评定需要有产品部门统一收口审核完成后记录需求看板。

在需求管理中，为了有效地管理工作负荷并确保关键任务得到及时处理，可以采用 P0, P1, P2 等级制对其进行优先级排序，然后体现在可视化看板上，帮助团队成员明确工作的优先顺序和资源分配。P0 代表最高优先级的任务，通常关联到对项目成功至关重要的功能或紧急问题，如关键功能实现或重大漏洞修复。这类任务需要立即关注并优先完成。然后是 P1 级别，表示高优先级但不如 P0 紧急的任务，涵盖了重要功能或较严重问题的处理。这些任务应在所有 P0 级别的任务处理完毕后着手解决。最后，P2 级别的任务是中等优先级，包括一般功能改进或非关键错误的修复，通常在处理完 P0 和 P1 级别的任务后进行。看板需求优先级列表示例见图 4.7。



名称	进行中节点	需求状态	优先级	开始时间	PM
1 新建页交互优化	● 运营 待完成	开发中 1个月	P2	2023-04-14	李磊
2 暂停与离职账号成员适配方案	● 完成「进行中」	产品评审 1个月	P0	2021-11-28	刘梓辛
3 权限配置化配套改造	● 运营 待完成	开始 1个月	P1	2023-05-05	Shalia Kumar
4 新建页面支持选择不同流程类型	● 产品设计 待完成	产品评审 1个月	P2	2023-05-01	陈浩
5 新版交付物解决方案	● 产品设计 待完成	开始 1个月	P0	2023-04-20	陈一卓
6 看板支持自定义字段	● 2项待完成	待产品评审 4天	P1	2023-04-26	陈浩
7 工作台支持展示不同待办信息...	● 产品设计 待完成	开始 1个月	P2	2023-04-18	陈浩
8 需求终止提供单独goapi, 前...	● 2项待完成	产品评审 1个月	P1	2023-05-04	Olivia Holt
9 SearchUserExactly接口增加...	● 产品设计 待完成	待产品评审 4天	P1	2023-05-01	李磊
10 列表页查询级联字段分组排序...	● 4项待完成	待产品评审 4天	P2	2023-04-16	Olivia Holt
11 增加前端包封禁能力	● 产品设计 待完成	产品评审 1个月	P1	2023-04-18	Naomi Hill

图 4.7 需求优先级列表示例图

注 d: 图片来源-飞书项目看板

如果存在相同级别的多个需求时，由产品和业务团队综合对业务价值、需求的重要性和紧迫性进行衡量，并充分考虑团队人员的存量工作任务，来设定需求优先级。最后需确保优先级设定与公司的整体业务目标一致，期间也可以根据项目进展和条件变化灵活调整优先级。通过这种方法结合可视化看板展示需求优先级列表，可以更有效地管理各级别需求，确保项目资源的最优利用<sup>[49]</sup>。这种优先级排序可以为团队提供了清晰的工作指导，也可以帮助项目经理合理分配资源，确保最关键的任务得到优先处理。通过这种方式，团队能够更有效地应对多任务挑战，保持项目进度的有序进行。

此外，为了确保需求优先级的透明度和一致性，通过明确和及时的沟通使团队中成

员和都能理解当前的优先级设定及其背后的原因。比如可以定期会议回顾和更新需求。这个过程应兼顾产品经理、项目经理和业务人员等各方的需求都能得到平衡考虑。职能部门通过定期召开任务回顾会议，团队可以审查和调整成员任务列表，确保每个成员当前阶段可以根据需求优先级顺利完成自己的任务项。这种动态的需求管理方法有助于适应项目进展过程中的变化。另外，看板还可以通过功能性标签或颜色编码区分不同优先级的需求，这些方法不仅有助于保持团队对优先级的关注，还促进了更高效的资源分配和任务管理。

(2) 制定需求基线

制定需求基线主要包括创建标准化的需求提交模板和文档规范，进行需求分析和评审，以及定义变更控制权限。首先需要为业务产品部门制定一个详细的需求提交模板，确保所有需求包含关键信息如需求背景和目标，具体包括要解决的问题、名词解释、目标必须是可衡量的<sup>[50]</sup>，能帮助团队事后判断是否实现了目标；需求优先级、对上下游系统的依赖和详细描述。

需求通过具体和清晰的文档方便项目人员准确理解和分类每个需求，这些需求经过详细评审后，成为需求分析规格的一部分，并连同评审报告进入基线库作为项目资料保存。项目经理在项目计划和配置管理计划中明确变更控制权限，确保团队成员根据授权进行操作。制定需求基线的目的在于确保所有项目成员对项目需求有一致的理解，避免重要信息遗漏或产生歧义。并且可以提供一种有效的变更管理机制，帮助项目团队适应需求变化，同时避免项目范围无序扩张。这一过程有助于提高项目效率，确保项目质量，减少不必要的重复工作，从而提升整体项目管理的系统性和效率。需求基线包含内容见表 4.4。

表 4.4 需求基线内容表

序号	内容	描述
1	创建需求提交模板	设计标准化模板，确保每个需求包含完整信息，包括背景、目标、问题描述和名词解释。
2	确定文档规范和结构	规范统一的文档格式、结构和内容要求，保证信息传递的一致性和易于理解。
3	需求的详细描述	需求文档需提供详细的描述，包括具体的功能点、性能要求、界面设计等。
4	需求依赖和接口定义	确定需求之间互相依赖关系和对上下游系统的接口要求，确保需求实现的可行性和完整性。
5	编制需求用户操作手册	提供详细的用户操作手册，包括系统使用、功能实现内容，以指导运营人员和用户。

### （3）双阶段需求评审步骤

需求评审是确保需求质量和适应性的关键步骤，需求评审优化方案可以分为预评审和正式评审两个阶段。预评审阶段由产品团队、业务团队和交互设计团队进行互评，主要目的是提前识别并解决需求文档中可能存在的问题。在这个阶段，团队成员通过相互检查和提出建议，共同努力提升需求的质量<sup>[51]</sup>。这一过程有助于提前发现和纠正潜错误，也能促进团队成员之间的协作和沟通，促进更深入的需求理解。

进入正式评审阶段，项目团队会正式举行评审会议，邀请项目相关成员参与。在这一阶段会对需求内容进行更广泛和深入的审查和讨论。通过汇集不同角色的观点和反馈，评审过程可兼顾需求的全面性和准确性。收集和整合各方的意见和建议有助于进一步完善和细化需求，确保后期项目的顺利进行。

### 4.3.2 提升需求设计能力

#### （1）员工培训与能力提升

产品部门需要对员工制定定期的学习培训计划，促使他们熟悉需求管理的方法和工具提高工作能力和效率。可以使用项目管理知识库统一维护需求，包括需求详情和跟踪，版本控制和变更记录。

项目管理知识库统一分类维护各产品线需求，可以确保项目相关人员查看一致的文档内容和需求变更信息。通过这样的知识库，团队成员能够追溯每个需求的变化情况，及时调整和更新自己的工作内容。另外，培训还包括提升员工对所涉及业务的理解能力，使他们能够更好地理解公司的业务方向和分析业务需求。这不仅需要对相关的业务背景有深入的了解，也需要能够从用户和市场的视角出发，准确把握需求的定位。

培训还需关注提高员工的文档输出能力。使他们能够清晰准确地描述自己的需求目标和需求内容，形成详细且易于理解的文档。清晰准确的文档便于团队成员之间的理解沟通，也能降低与其他团队成员对需求信息理解产生偏差的几率。

最后，对外沟通能力的提升也十分重要。产品经理需要具备良好的沟通能力和技巧，能够在不同的部门和不同成员之间进行有效的信息传递和协调，确保项目需求可以顺利推进。

#### （2）培养需求分析师

在目前人力资源相对有限的情况下，产品部门可由各个产品经理和业务人员轮流担任需求分析师，可以更好地利用现有资源来提升需求的整体质量。产品经理在承担需求

分析师职责时，需要深入理解和分析用户及业务需求<sup>[52]</sup>。在需求收集和文档化过程中不仅要关注产品的整体方向和策略，还需站在第三方用户的视角来审视需求的细节，确保需求的清晰性和可理解性。这样的角色轮值有助于确保需求符合业务目标的同时也具有实际可操作性。

为了使产品经理和业务人员能够做好需求分析这项工作，优化需求评审过程很重要。在需求前期规划环节，需求分析师需要对需求内容进行有效评估分析，以确保它们与业务目标相符。在内部预评审和正式评审环节，需求分析师作为一个关键参与者，负责促进需求的完善和提升需求质量。这包括在内部预评审中提前识别和消除问题，并在正式评审会议中与其他项目团队成员再次共同讨论和完善需求。通过对人力资源的高效利用和流程的优化，可以有效提高项目需求的质量，降低后期需求变更次数和开发测试的成本。

4.3.3 制定需求变更流程

项目经理负责监控和管理所有变更请求，利用看板项目管理工具记录、分析并追踪每个变更的影响。需求变更流程见图 4.8。



图 4.8 需求变更流程图

(1) 发起变更

产品经理发起需求变更的场景可以分为以下两种：原有项目范围发生变化（立项后）、项目评审排期后原有需求范围发生变化。发起时需提供的内容包括：变更发起方是业务或产品、变更发起人、变更时间、变更对象是项目还是某个需求、变更理由、变更内容包括变更前和变更后内容。

(2) 影响分析

项目经理负责对变更内容进行影响分析，包括对成本的影响、对进度的影响、对资源的影响。并且需要登记信息包括：变更涉及方包括开发测试的范畴，执行变更人员、变更需要的总工作量等。

(3) CCB 审核

CCB（全称为 Change Control Board，即变更控制委员会）需进行审核变更。审核

内容包括：评估变更的必要性和紧急性、分析变更对项目的影响、风险评估等。CCB 根据情况审核后给出同意和拒绝，如果拒绝变更需要说明理由。CCB 的审批决策需要全面记录并通知相关方，尤其是变更请求发起人。项目经理根据 CCB 决策更新项目文件和计划，以反映已批准采纳的变更内容。

#### （4）变更实施

首先项目经理负责通知所有项目相关者有关变更的批准和实施事宜；第二是变更内容更新，需要调整与变更相关的项目所属需求文档、及子任务等，与批准的变更点保持一致；第三是新增变更内容跟踪，将新引入的需求及子任务等新增变更内容录入跟踪表单中；第四是实施跟踪，项目经理负责跟踪新引入变更内容在实际开发及实施中的进展情况。

#### （5）变更展示

在看板项目工作项详情页，变更管理菜单下能看到所有关联的变更。确保实时反映最新批准生效的变更状态，保证多方访问同一信息。可以查看和核对每次变更前后的对比，审阅变更差异。通过“变更管理”菜单实现标准化、透明的变更展示，提升多方变更信息访问的完整性与一致性，是变更管理的重要一环。需求变更看板示例见图 4.9。

影响类型	变更工作量
交付时间 +1	3
暂无	暂无
交付时间 +1	6
人力成本	5

图 4.9 需求变更操作看板图

注 e: 图片来源-飞书项目看板

## 4.4 研发测试工作优化方案设计

### 4.4.1 开发质量提升敏捷化

#### (1) 开发代码质量稳定提升

有效执行开发工作流程。从开始概要设计阶段设定了项目的目标和范围，这一步骤明确了团队的工作方向，减少人员工作出现偏差的可能性。随后的详细设计阶段将大方向的规划细化为具体实施步骤，这是一个质量控制的环节，可以提升设计的完整性和可行性。最后验收和上线方法的制定需要确保生产发布流程规范，避免出现功能丢失和发布过程中用户使用受影响的问题。开发工作流程见图 4.10。

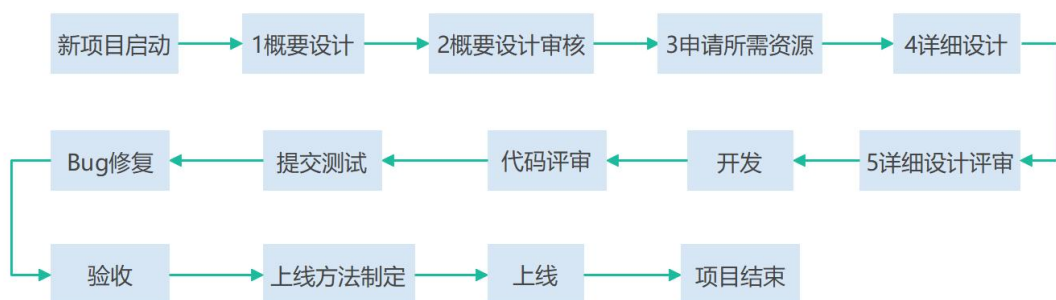


图 4.10 开发工作流程图

重视技术架构设计。需求评审通过后，项目经理可在预开发阶段组织技术设计会议，提前让研发介入需求，讨论合理的技术选型与方案，期间就可以就技术框架设计与产品经理、交互设计师、测试等人员进一步明晰代码实现逻辑。这一过程涉及到确保选定的技术框架稳固无误，以及进行准确的开发人员工作量评估，同时考虑到实现逻辑的合理性、效率和性能。这有助于规避后期因技术选型不当或评估工时不准确而造成延期的问题。

执行代码质量评审检测。为了进一步确保代码的质量，需要严格实施代码评审，尽可能在这一环节发现问题。让开发者复述所实现需求的设计思路是否与需求一致，重点关注代码是否依照概要设计与详细文档的规划编写，是否遵循开发规范和设计原则，是否有性能和安全漏洞等。代码评审环节进一步确保了开发质量，通过同行评审，可以在代码合并到主分支之前发现和修复错误，基于评审人的经验判断代码是否合理，由部门经理判断是否通过，也可以促进团队内部的知识分享。此外关于代码架构的质量问题，可以通过引入代码自动化分析工具对代码进行规范检测<sup>[53]</sup>，自动识别代码中的缺陷、安全漏洞以及不符合编码标准的地方，帮助开发人员及时修正问题。

#### (2) 打造协作与学习型开发团队

建立一个良好的团队协作和知识管理机制是促进团队成员之间交流和学习的關鍵。



研发中心需要对开发人员进行持续的培训，确保开发人员能够掌握最新的技术和工具，是提升开发质量和加速项目进度的基础。这种培训不仅包括新技术的学习，还包括前期项目中最佳实践、工具的有效使用方法以及软件开发原则，以促进开发人员的全面成长。

此外，通过定期的技术分享会和复盘会可以加强团队成员之间的沟通，分享彼此的知识和经验。这不仅有助于构建一个学习型团队，也能提升团队解决复杂问题的能力，最终提高软件开发的质量。并将学习协作成果落实到绩效考核里面，设置专项奖励。通过团队投票或者部门领导评定的方式，关键结果明确，有绩效考核成员就会有动力去实现技能成长，最终促成团队良性发展。

### （3）技术创新与研发文档完善

研发中心要坚持技术创新和完善研发工作文档，技术创新不仅是采纳新工具或新技术，更在于鼓励团队思考如何用创新的方法解决旧问题，以及如何在现有的技术框架内寻找效率更高、性能更强的解决方案。同时，完善的研发文档可以为其他开发人员，包括产品和测试人员都提供了重要的参考资料，也保证了开发过程的透明度和后续维护的便捷性，可以促进团队内部的信息共享和知识积累。通过将这些文档保持最新状态，不仅能够为团队成员提供清晰的指引，减少误解和重复工作，还能够确保产品和测试人员对项目的完整理解和正确实施。此外，良好的文档管理还有助于新团队成员快速熟悉项目，加速他们的融入和贡献。

### （4）业务数据分析与运维

完善线上业务数据的定期巡检，可以更及时发现系统的潜在问题和性能瓶颈，实现问题的快速响应和解决。这一过程是指对服务端运行状态的持续监控以及对产生的问题进行迅速反馈和处理，从而确保服务的连续性和可靠性。首先需要梳理对应的巡检场景，一般包括业务资损场景、研发规范场景、历史故障场景等，场景梳理完成后创建对应的测试任务，分析评审通过后可创建线上的巡检任务并通过自动化的方式来触发执行。每个自动化巡检任务都会有相应的校验点来验证结果是否符合预期。经过阶段性的执行和数据分析会形成一个线上系统的质量评估趋势。

如果巡检过程中出现异常或者其他不符合预期的情况，则负责对应业务的开发和测试人员会进行故障响应，及时处理故障以确保线上业务稳定运行。在此基础上各团队需根据线上问题的重要性和严重程度，通过定期和不定期的线上业务数据复盘和改良逐渐沉淀运维经验来提高服务的稳定性，也能进一步优化用户的服务体验。

4.4.2 测试质量提升敏捷化

(1) 规范编写测试用例标准

测试用例是用来验证开发出来的软件功能满足产品设计要求，良好的测试用例需要详细指出需要测试的每个功能，并且程序错误尽量都能被发现和修正。因此，编写的测试用例的过程应遵循严格的规范，以确保测试的全面性、一致性和可重复性。编写测试用例标准见表 4.5。

表 4.5 测试用例编写标准表

序号	编写测试用例标准
1	测试用例本身的描述是否清晰准确；是否存在歧义。
2	测试用例内容是否完整，是否清晰的包含前提条件、输入和预期输出的结果；测试步骤是否清晰。
3	测试用例中使用的测试方法、测试数据是否恰当合规。
4	测试用例是否是否能指导测试工程师通过测试用例发现更多的缺陷，而不是限制他们的思维。
5	是否考虑到测试用例执行的效率，或者测试用例的设计是否存在冗余性等。
6	验证测试用例是否完全覆盖了需求，异常场景是否覆盖，要验证测试用例的覆盖性。

(2) 有效执行测试用例评审

测试用例评审对验证测试用例的正确性、有效性、测试覆盖度等十分必要，而且可以有有效的保障测试实施，并且对测试用例改善工作也至关重要。

在评审之前，测试人员需要将即将评审的测试用例文档提前发送给项目相关的人员；让他们有时间提前阅读，以便于在测试用例评审会议上直接向测试人员提出问题；在评审期间测试人员需要做好详细的记录，需要对有关的疑问和问题及时进行回复和补充。会议期间，测试人员要督促相关人员集中精力在测试用例审查上，以免有疏漏的情况，后期避免出现线上问题；在评审会议结束之后，测试人员要及时修正完善测试用例，在项目相关人员认可后，上传用例管理知识库。

最后需要实施反馈循环和持续改进机制。这包括测试用例执行的状态和缺陷管理，还有对测试流程本身的评估和优化。通过定期的部门会议回顾和分析测试活动，质量团队能够识别改进机会，持续提升测试流程的质量和效率。较完善的执行测试用例能够确保测试覆盖到项目的绝大部分并能够更详细地评估测试过程的质量。

(3) 提升自动化测试能力

目前公司业务日渐复杂多元，且对软件交付速度要求也很高，中后台业务采用接口自动化测试工具和框架可以有效提升测试效率和覆盖率。这可以减少对手工测试的依赖



且降低错误率，尤其是在处理复杂的业务链路场景和耗时的逻辑测试时十分必要。通过集成多环境运行能力，即测试环境和线上环境的自动化测试用例运行能够无缝切换或配置切换，可以为扩大自动化使用场景并提高投资回报率奠定基础。

具体到接口自动化测试，采用动态配置管理和数据驱动的测试方法，能够实现对不同测试环境的快速适应和高效管理测试数据，进一步提升测试的灵活性和适应性。此外，通过模块化设计的测试脚本，不仅能够覆盖复杂的交易流程，还能通过模块的组合和串联来高效处理复杂且耗时的业务逻辑，同时保持测试脚本的可维护性和可复用性。

目前用户端监控薄弱，线上环境缺少有效的用户端监控，不能提前发现问题导致出现问题后比较被动，可能会造成损失。针对线上环境可以做用户端自动化测试提供一种线上巡检机制检查核心页面的展示和可用情况，通过模拟用户操作，更贴近实际使用场景。通过选择合适的自动化工具和框架可以提高测试的执行效率，也扩大了测试的覆盖范围。

#### （4）打造协作与学习型测试团队

质量团队的专业素质和技能水平直接影响测试效率。因此，质量团队中应该包含多样化的技能组合和专业分工成员，如接口测试、性能测试、自动化测试、安全测试等覆盖软件测试的多个方面。质量部门负责人要善于发现团队成员的优势和专长，把正确的人放到合适的位置上。分工完成后，确定每个人对应的职责后同每一位团队成员进行明确说明，个人任务完成情况与绩效考核挂钩，和开发团队绩效考核一致。

同时，持续提升测试团队技能和促进成员协作分享也可以很好提高测试效率，测试团队需要定期组织参加专业培训、技术研讨会等。鼓励成员了解最新的测试技术和工具，不断学习和探索新的测试领域以提高自身的专业素质和技能水平。结合成员自己的意愿，制定发展发向，如接口测试专项、性能测试专项、UI 自动化专项等等。

#### 4.4.3 回顾和持续改进

产品研发中心定期回顾和反思能够帮助团队及时调整自身方向、提升团队学习能力，是 M 公司敏捷实践当中必不可少的环节。

##### （1）定期会议回顾

立项会议。目前项目经理会在项目启动后召开参与人员的立项会议，会议内容包括项目起止时间、项目背景、业务价值、项目相关者和利益点描述、核心功能清单、关键里程碑和阶段性成果、约束条件、验收标准、风险管理等。

项目例会。项目进展过程按周或者里程碑节点会召开参与人员例会，会议内容包括项目当前进度、项目有无变动内容、有无进度风险以及后续核心工作等。

复盘会议。项目交付后流程部成员会根据各项目的最终实现情况决定是否召开项目复盘会议，会议内容包括：回顾项目计划和进度、需求文档和变更记录、交付物和成果物、缺陷和问题跟踪记录、业务反馈和满意度调查、回顾项目执行环节等<sup>[54]</sup>。

### （2）可视化看板监控和调整

看板的功能为M公司产品研发中心项目管理带来了更大的灵活性和实时性，通过使用可视化工具，项目经理和项目成员可以清晰地看到项目所有的任务分配和进度。这种可视化管理使得任何任务分配的调整都变得直观易懂。项目经理也可以根据需要迅速调整任务调整，应对突发事件或变更需求。这种实时更新和同步确保了团队成员能够及时看到最新的任务和排期情况，从而快速适应项目需求的变化。

### （3）持续改进流程

在项目进展过程中，有效的资源分配和人员排期至关重要。这不仅涉及到根据改进计划和团队反馈来调整任务，还可以利用看板来提高这一过程的灵活性和效率。首先针对需要改进的目标和团队反馈问题，项目经理需要深入分析团队在回顾会议中提出的改进意见，综合考虑团队成员对当前任务分配和时间安排的看法，并根据项目的最新需求和优先级进行人员排期和任务重新分配。在这个过程中需要了解成员个人的工作任务进展和排期，以避免个别成员任务过量无法如期完成。完成这些调整后，项目经理应与团队成员进行有效沟通，确保每个人都清楚自己新的责任范畴，确保成员对调整后计划知情且同意。

综上所述，通过结合对看板工具的灵活运用和定期会议回顾进行持续改进调优，可以显著提升开发和测试过程中资源分配的效率和团队的工作交付质量。这不仅能提高项目管理的透明度，还可以增强项目成员对项目进展的控制感，从而提升整个项目团队的工作积极性和满意度<sup>[55]</sup>。

## 第5章 优化方案的实施与效果跟踪

### 5.1 试点项目背景介绍

M 公司在对研发中心项目管理存在问题进行调整和优化后,虽然很多流程优化工作还存在一定程度缺陷,但是从目前的项目实践状况来看,优化工作也颇有成效。

本章选取了 M 公司研发中心在 2023 年 10 月初确认立项的 S 项目作为优化方案效果验证的试点项目非常合适,也为后期的相关研究提供更多的反馈和探索。

S 项目的业务目标为新增站内信触达渠道,通过推送通知和促销信息,和用户产生链接,并激发顾客的购买欲望,最终达到提高网站转化的目标;用户通过站内信及时获取平台信息,通过阅读和了解平台信息,实现用户利益最大化。业务部门和研发中心预估该项目的交付周期约 1 个月左右,由于该项目需求内容和投入人员规模适中,团队成员技术和协作能力较为成熟,业务部门也希望可以按照计划安排实施,且整体人员进度和风险可控,因此本项目适用敏捷看板管理工作流程。

### 5.2 优化方案实施步骤

#### 5.2.1 组织协同机制优化方案实施

##### (1) 按需组织项目成员参加会议加强沟通

S 项目过程中有关于沟通方面,需要团队成员共同参与的讨论和决策的一般通过线下会议形式,包括项目立项会议,需求、技术、用例评审会议,周会,复盘会议等等。会议结束后由项目经理或会议发起人整理会议纪要至知识库,会议期间的沟通内容,项目经理或会议发起人会指定对应责任人跟进沟通结果。日常的开发测试过程中有很多细节问题一般会进行当面沟通,或者在项目沟通群发布消息,这样其他成员也可以查看消息记录同步其沟通内容,当然如果非正式沟通有涉及需求或者开发的变更内容也需要更新到知识库,并确认相关责任人知悉且同意。

##### (2) 发布和实施项目绩效考评机制

据项目和个人双重绩效考核机制,由项目管理办公室发起。关于评估维度,参评成员通过项目准期率、项目范围确定性、过程管理能力、决策速度、项目难度等几项指标在项目上线后为 S 项目实施绩效考核,为该项目提供了相对客观公正的绩效反馈。并在后续的复盘会议中指出其评分说明,回顾项目执行过程中的优缺点、以及收集到的用户

反馈，并给出后续的改进建议持续优化。在之后的职能部门会议中，部门经理整理了各项目的复盘会议总结对部门成员提出工作要求，并根据项目和成员的绩效评分发放对应的项目和成员的奖励。

5.2.2 项目过程控制优化方案实施

(1) 看板创建项目 and 需求

S 项目在看板上创建项目后，根据优化方案设计的流程执行，由项目经理负责组织和领导项目团队，并负责项目过程的整体管理。首先项目经理在项目启动后召开相关人员的立项会议，立项通过后由项目经理和产品经理依次创建项目和拆分需求，然后对项目进行综合评估并组织所需人员参加需求评审会议，评审通过后对各团队成员进行任务分解并评估工时，项目经理据此制定预计上线发布时间。创建项目(Epic)和其所属需求(Story)示例见表 5.1。

表 5.1 项目所属需求示例表

Epic 名称						
序号	Story 名称	Story 状态	负责人	需求排期	预计上线时间	实际上线时间
1	运营后管	已结束	A	10.17-10.24	2023.10.31	2023.10.31
2	C 端-后端	进行中	B	10.18-10.27	2023.10.31	2023.10.31
3	C 端-web	进行中	C	10.18-10.31	2023.10.31	2023.10.31
4	C 端-APP	未开始	D	10.25-11.10	2023.11.15	2023.11.15

(2) 看板跟踪成员任务完成进度

开发和测试工作开展前，项目经理组织召开技术评审会议进一步与成员同步完成任务细节，之后项目经理组织项目周会与成员同步项目当前进度、项目有无变动内容、有无进度风险以及后续工作等。开发和测试任务完成之后交由产品、设计师和业务人员进行验收，验收完成后发布上线，各成员完成各自流程任务后更新看板状态，期间项目经理和各团队成员都可以通过可视化看板查看项目工作进程。任务分解看板示例见表 5.2。

表 5.2 任务分解看板示例表

Story	Todo		Doing		Done
用户故事 1	任务	任务	任务	任务	任务
	任务	任务			
用户故事 2	任务	任务	任务	任务	
	任务	任务			

续表 5.2 任务分解看板示例表

Story	Todo		Doing	Done
用户故事 3	任务	任务	任务	任务
	任务	任务		

### （3）按需在项目完成后组织复盘总结

项目经理可根据工作流程执行状况按需组织项目复盘会议，回顾项目过程了解业务反馈，确定有无可以改进的环节，项目系列会议由项目经理整理并形成会议报告发送项目相关者。

## 5.2.3 业务需求管理优化方案实施

### （1）将需求重要性和优先级录入看板

由于研发中心同一时间内各项资源有限，业务和产品部门根据项目价值和紧急程度评估 S 项目为 P1 级别，即优先级较高但是不紧急的项目。S 项目的产品经理将项目所属需求和其对应的优先级录入看板，确保该项目会优先投入人力安排、优先处理交付，满足业务部门需求。之后，各项目成员也可以在看板以及项目和职能部门会议上确认自己参与项目需求的优先级。

### （2）根据需求基线要求输出需求相关文档

关于需求文档，产品经理根据产品部门统一制定的需求基线要求编写需求文档和原型图及附件文档，完成 S 项目的产品详细设计说明书，并通知 UI 设计师协助根据需求完成 UI 设计图。S 项目需求文档包括：需求背景目标、功能框架设计图、名词解释、原型图、功能描述、附 UI 稿和对应展示说明、所涉及接口设计要求、用户操作手册等内容。其中也包括需求版本管理即初版和新增补充及变更删除等信息，方便给项目组其他成员查看需求实时变动内容。

### （3）组织项目成员进行需求评审会议

在需求编写完成后首先是在产品、业务和 UI 团队内部评审检查有无明显问题，之后再组织项目有关成员进行需求评审会议，评审过程中产品经理和需求分析师要充分收集和记录各成员提问和意见建议，待会议结束后进一步确认和完善需求直至需求定稿，通知所有项目成员后续开发和测试工作可以相继开展。最后项目相关文档会统一收录至产品知识库归档，方便后期查看追溯。

### （4）根据项目实际情况确认是否操作变更

S 项目在开发过程中有因为业务人员和产品人员沟通理解偏差问题还是有发生需求内容变更,但是范围不大,内容只是稍作调整。产品经理和项目经理经与相关人员会议讨论后确认有需要变更,并确认变更内容对项目造成的影响不大之后,即发起变更流程,流程审批完成后实施变更,所有涉及变更内容在看板和需求文档完成更新并通知相关成员后,成员即可查看变更后最新的需求信息。

#### 5.2.4 研发测试工作优化方案实施

##### (1) 按需组织技术评审或完成技术文档

在 S 项目需求评审通过后,开发人员在预开发阶段进行应用架构设计并形成技术设计文档,之后由项目经理组织项目相关者召开技术评审会议。评审的首要任务是确保架构设计的完整性和合理性。这涉及到评估系统组件的配置,包括它们的相互作用是否符合逻辑,数据流、服务和接口设计是否满足项目需求;所选技术是否适合项目目标,包括考虑技术的成熟度、性能指标、成本效益以及团队的熟悉度。

另外,架构设计必须能够满足性能要求并支持未来的增长,考量架构设计是否能够处理预期的负载,包括用户并发数、数据处理速度等;同时,评审会关注架构设计的可扩展性,确保它能够适应业务增长带来的压力,技术评审完成后需进一步完善技术文档以方便其他成员查看追溯。

##### (2) 项目上线前组织代码评审和质量扫描

项目经理在团队开发任务完成后准备提测之前,组织开发团队相关成员进行代码评审以及在开发环境通过静态代码扫描工具对各自代码质量进行检查。检查是否有明显的逻辑错误和异常场景、是否落实代码规范、检查代码的可读性和可维护性是否良好等内容,并将报出的代码质量问题进行修改,同时做好代码评审的记录,记录时间、问题的位置、问题的修改方案。同时通过代码审查也可以实现较好的知识共享,帮助其他开发者了解代码的设计思路和实现方式,代码审查记录可以作为参考文档,帮助其他人理解代码、查找问题。

##### (3) 项目上线后按需配置业务巡检监控

业务系统上线后为保障系统稳定运行,设置每日巡检值班人,由值班人员负责完成对系统的巡检和技术支持工作,其他人员协助配合其完成工作。巡检内容包括查看监控系统整体运行情况,如有异常会即时在巡检工作群发出告警消息提醒值班人员,值班人员响应处理完成后回复该消息处理结果,以备核查是否有遗漏告警。

#### （4）严格执行测试工作流程保障项目质量

在 S 项目开发阶段项目测试负责人制定了项目测试计划，对参与测试人员任务进行分解，明确所需的测试资源。期间各测试成员编写测试用例，开发人员开发任务完成后进行自测。项目团队人员随时关注项目进展状况，及时响应变化和调整。用例评审通过后，测试过程中 Bug 提交至平台并督促开发人员及时处理并验证通过，然后在测试任务完成后在看板更新任务状态通知产品、设计师和业务人员验收。在项目上线后的一周内测试成员利用测试空档期，将 S 项目所涉及到的业务功能场景，编写接口自动化用例提交运行。自动化测试优先覆盖核心业务，对于核心业务的正常和异常场景都要覆盖。

对于生产缺陷，如果识别到是测试场景遗漏，优先考虑通过自动化测试来覆盖遗漏场景和类似场景，可以在自动化项目里单独创建一类补充型目录，放置此类场景的自动化测试用例，每次回归测试时批量运行，避免再次出现问题。自动化用例执行通过持续集成和交付工具触发任务，也可手动执行任务用于回归测试。页面自动化用例可由团队成员在其他限定时间内完成，用于回归测试或者实施线上用户端页面巡检。

#### （5）组织项目工作会议总结和改进

定期的项目工作周会中讨论当前阶段需要完成的目标、在进行中的任务、已完成的任务以及目前遇到的影响项目进度的问题和对应的解决方法。项目经理可按照时间定期召开。项目结束后，项目经理可视情况召开复盘会议，周会和复盘会议的主旨在于问题回顾总结和持续改进。项目工作会议示例见表 5.3。

表 5.3 项目工作会议示例表

节奏	频率	目的
立项会议	项目启动后	立项会议的目的是对项目的整体规划、范围、进度、成果、风险等内容进行明确和沟通，确保项目团队和利益相关方对项目有一个整体而清晰的认识，以推进后续的顺利实施。
项目周会	项目过程中的每周	实时跟踪项目的最新进展，基于回顾的结果规划以后的工作方向，包括确定新的目标和改进措施；确定资源重新分配的需求，如人员排期和任务调整；通过周期性更新，可以及时响应变化，减小风险对项目的影响，保证如期交付结果。
复盘会议	项目结束后	确定项目管理过程中可以改进的环节,总结经验教训；形成回顾报告和提升方案，防止类似问题在后续项目中发生，通过定期回顾，不断优化项目管理机制，提高项目管控能力和资源利用效率。

## 5.3 优化方案实施效果跟踪

在对 S 项目进行的项目流程优化实施全程跟踪后,研发中心总体认为改进效果较为显著。在优化方案实施之前,管理层对研发中心成员进行流程实施培训,并对看板平台各个角色的操作提供了近两个月的试用期,在试用期内员工可以逐步适应对看板的项目流程推进和协作。在试用期结束后,团队成员正式投入敏捷看板工作流程实践中,结果证明相比于之前使用 Jira 项目管理工具进行的瀑布开发流程,敏捷看板在管理开发工作流程方面的表现更具优越性,同时业务部门对项目交付速度和质量反馈也比之前更高。

为了深入的评估项目管理优化效果,本文选取了 M 公司在流程优化前交付的 T 项目作为对照。考虑到 T 项目的规模和参与人员以及业务部门对该项目的关注度与 S 项目相当,因此两个项目具有可比性。接下来本文从以下四个方面来阐述敏捷项目流程优化效果。

### 5.3.1 成员沟通协作成本降低

通过实施促进团队成员之间沟通协作和加强组织协同管理的方案,我们显著提升了团队协作的顺畅度。明确了成员间的沟通机制和提升了沟通的准确度和效率,团队成员能够更有效地交流和协调工作,减少了误解和冲突,加速了决策过程。

首先,在人员利用率上有良好的提升,说明动态人力资源池调配机制的有效性。另外由沟通不当引起的错误率也有显著降低,这一改进效果说明了通过优化沟通机制,有效减少了因误解或者信息传递不准确导致的错误,从而提高项目执行的准确率和效率。

项目有关的知识库文档完成度也有很大的提升,这意味着当前项目的一系列文档资料相比之前更为完善,这为后续项目的回溯和迭代提供参考,强化了组织的知识管理和积累。

T 项目(优化前)和 S 项目(优化后)的沟通协作成本降低比率效果见图 5.1。



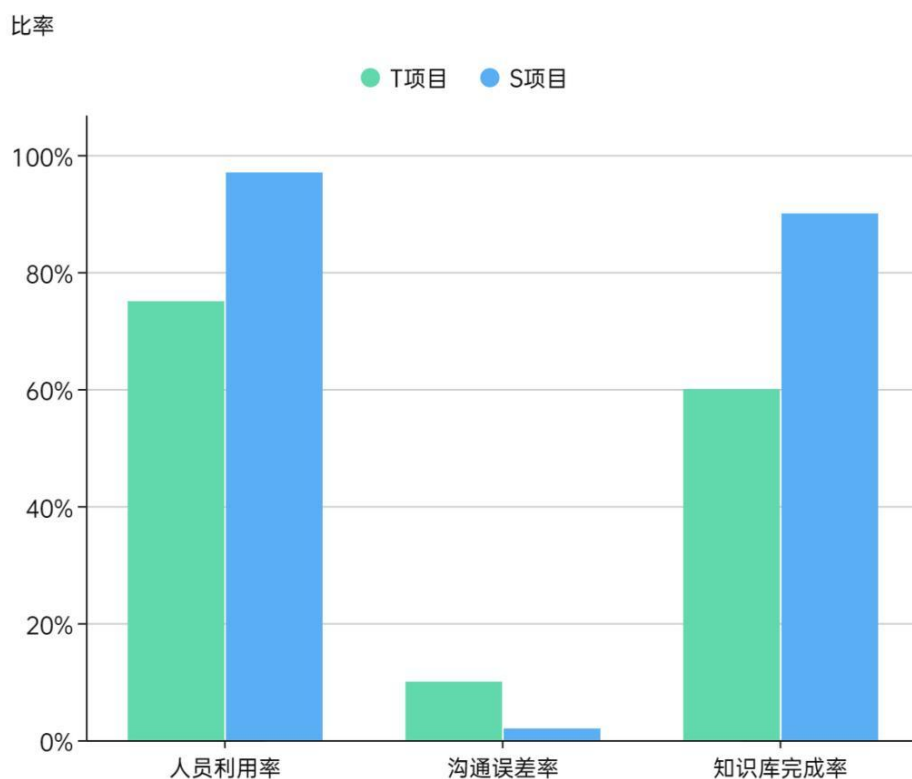


图 5.1 沟通协作成本降低比率效果图

### 5.3.2 项目交付准期率提升

通过实施敏捷工作流程改进、使用看板项目管理工具以及加强项目经理的监控职责，我们的项目过程控制能力得到了显著提升。管理过程更加灵活和透明，同时确保了信息的即时和准确传递。敏捷工作流程的改进增强了团队的适应性和响应速度，还加快了问题解决的过程，提高了项目交付的效率和质量。

在项目实践中，各个部门的需求交付准期率作为项目的重要过程节点，其准期完成情况能够直接反应项目是否能够按照计划推进，是项目成功的关键指标。根据优化方案实施前的 T 项目和优化方案实施后的 S 项目的交付情况，从大数据部门和中台部门以及用户端部门分别对于完成需求的交付准期率进行对比。

T 项目（优化前）和 S 项目（优化后）的项目交付准期率对比见图 5.2。

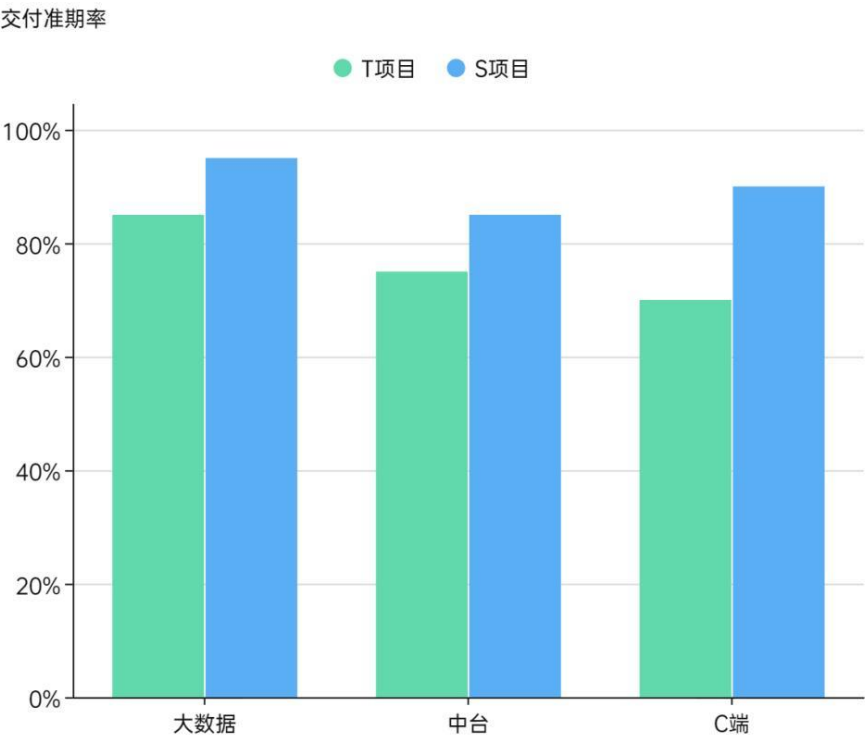


图 5.2 项目交付准期率对比图

5.3.3 业务需求满足度提高

通过实施需求设计管理体系的优化方案，包括需求管理体系的优化、需求管理与分析能力的提升，以及变更管理流程的严格执行，经业务部门反馈和项目复盘会议总结证明，需求满足度有显著提高。

首先是运营人员的满意度在 T 项目时期反馈一般，满意度评级为 B，在实施了针对性改进措施后，S 项目收到的业务反馈满意度评级为 A，改善效果较为明显，业务人员对研发中心投诉也有所减少。此外需求变更次数明显降低，需求在开发过程中因为各种原因造成的变更，对项目进度、人力和时间成本的影响也大幅缩小。

T 项目（优化前）和 S 项目（优化后）业务需求满足度评估见表 5.4。

表 5.4 业务需求满足度评估表

指标	T 项目	S 项目	改善效果
运营人员满意度	业务反馈一般，满意度评级为 B	业务反馈积极，满意度评级为 A	业务满意度显著提升，对产研部门投诉减少
需求内容变更次数	5-7 次	1-2 次	需求在开发过程中变更次数减少，节约各项成本

5.3.4 项目总体交付质量提高

通过深入实施质量管理体系优化，我们在开发和测试流程中取得了显著成效。优化后的数据显示 S 项目四个等级的线下缺陷数量都有明显降低，体现了代码提测质量的大幅提升。

此外，S 项目线上缺陷数量仅出现了一个普通问题，在上线发布之后测试人员也及时发现，并没有对线上业务造成影响，及时确保了测试的全面性和有效性，进一步证明了测试流程质量的显著提升。

T 项目（优化前）和 S 项目（优化后）线下和线上的缺陷数量对比见图 5.3 和图 5.4。

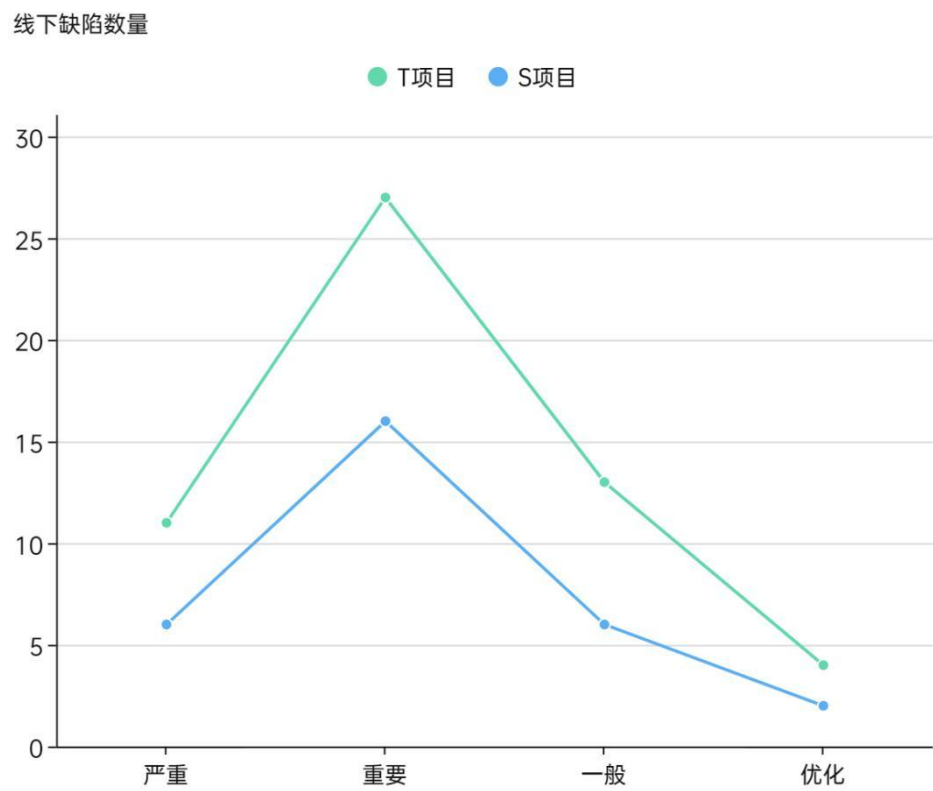


图 5.3 线下缺陷数量对比图

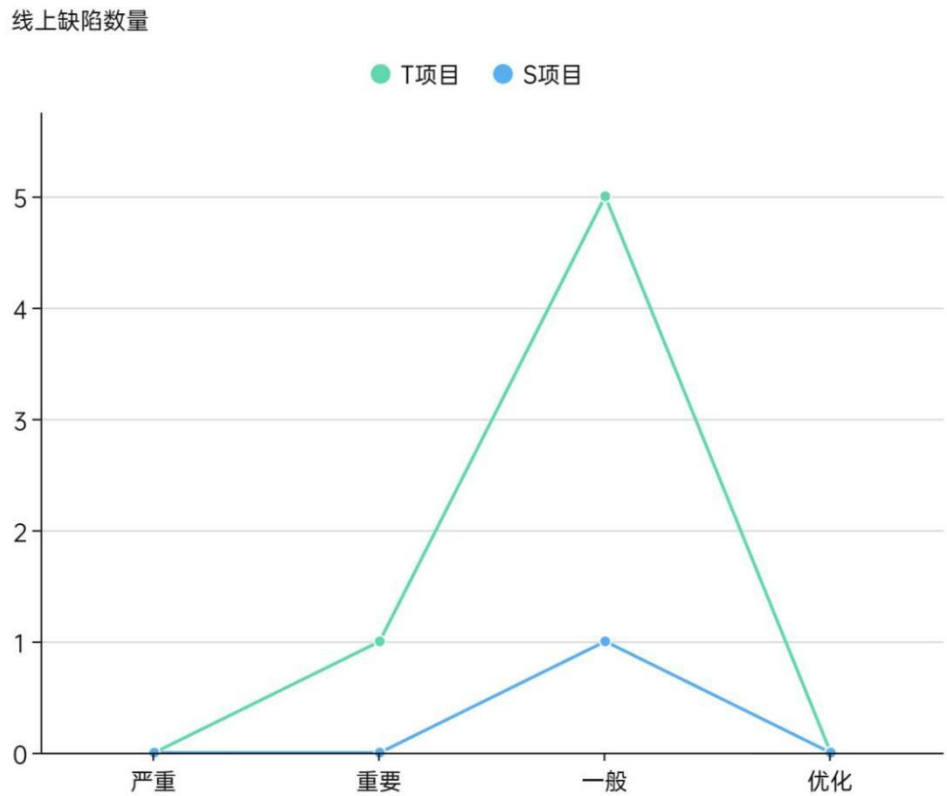


图 5.4 线上缺陷数量对比图

5.4 本章小结

本章节主要是对上一章节提出的优化方案进行实施和验证。首先选取了方案实施后的 S 项目开发作为试点项目进行验证，结合 S 项目的项目实施过程，对实施效果从成员沟通协作成本、项目交付准期率、业务需求满足度及项目总体交付质量等方面相较优化方案实施之前的 T 项目进行对比。S 项目经业务部门反馈、项目复盘结论和项目管理办法评估：综合评级为 A。项目的用户体验和价值远超出预期，从而得出本文优化方案的有效性。在后续的实施过程中，M 公司也会根据产品研发中心的实际情况和成员反馈对优化方案进行持续调整和完善，以确保方案能够实现长期价值。

## 第 6 章 结论与展望

### 6.1 研究结论

本文以 M 公司现有的项目开发管理过程为研究对象, 分析现行瀑布模式中存在的问题, 并基于敏捷模型看板(Kanban)管理方法提出优化方案, 最终得出结论如下:

(1) 从相关理论文献分析的结果来看, 敏捷看板管理方法属于一种更高效的敏捷开发实践, 能够保证开发过程的透明性和及时性的基础上, 实现项目的持续交付。

(2) 根据德尔菲问卷调研的结果, 经分析和梳理, 确定了公司项目管理过程中存在 4 个方面 15 个问题。之后采用层次分析法, 开展问题权重值分析, 经过一致性校验, 权重值排序后, 将方案层权重排序的前 10 个问题因素纳入重点考虑范畴, 而后 5 个问题因素在优化方案设计中会进行适当关注。对这些问题进行原因分析总结后, 得出业务需求管理存在偏差、项目过程控制效率有待提高、研发测试工作质量需要改善、组织协同机制存在不足是 M 公司项目管理过程中的待优化方向。

(3) 本文基于敏捷模型看板管理方法提出各项优化方案设计和选取 S 项目跟踪观测优化方案实施过程, 将 S 项目(优化后)与 T 项目(优化前)进行对比参考。经评估总体效果较为显著, 在实施敏捷模型看板管理方法以后, 成员沟通协作成本降低、项目交付准期率提升、业务需求满足度提高、项目总体交付质量提高。

### 6.2 未来展望

项目管理流程优化是一个持续改进和适应的过程, 它需要经过一定的时间和大量的流程实践进行反思总结和不断探索。本文虽然针对软件项目管理流程优化提出了一些具体措施, 但是由于研究视野和深度仍有局限, 所以管理启示还存在很多可以扩展的维度, 为了增强研究结果的指导意义和应用价值, 未来需要考虑不同企业规模和所处行业等因素, 对管理启示和优化措施进行有效分类, 以提高对不同类型的企业的理论指导和实践参考。

敏捷开发管理模型作为一种以人为核心和适应性强的管理方法, 随着技术的快速发展也会不断地改进, 以更有效地满足复杂项目需求和提高团队工作效率。流程优化作为敏捷管理中一个不断追求的目标, 强调简化流程和加强沟通协作, 希望将项目管理实践提升更高效和灵活的水平。对于 M 公司在敏捷项目管理流程优化方面, 可以从以下几个方面进行探索。

（1）更加灵活的方法论融合。目前敏捷主流的几种实践方法之间的界限变得越来越模糊，未来的软件项目管理将更加注重可视化、灵活性和适应性，M公司在之后的管理实践中可以考虑融合其他的方法论取长补短，在不同的业务状态中各自发挥作用，以适应项目的具体需求。

（2）管理方法和工具的深度融合。项目管理方法和工具可以更深一步融合，看板项目管理工具可以实时访问全量的项目数据，可以有效提高项目管理的有效性，简化流程并减少人为错误。M公司可以将项目管理工具的看板、即时沟通、会议日程、绩效等系列产品以及公司内部其他的数字工具和平台进行有效集成，可以提升信息快速共享能力，这也是数字化转型的关键因素之一。

（3）数据驱动的决策过程。通过看板项目管理工具、员工绩效信息和对公司知识库的整理归纳，来分析项目数据（如进度、成本、质量等指标），以后项目管理会趋向于参考数据分析结果来制定决策。M公司可以建立数据和分析的机制，使用可视化工具展示关键指标，以后可充分利用收集到的数据进行预测分析，进行优化资源配置、风险管理和提高决策准确度。

## 参考文献

- [1] 张涛.软件项目管理问题与优化策略研究[J].中国管理信息化,2023,26(18):150-152.
- [2] 姜文凭.提高软件产品质量的方法研究[J].企业改革与管理,2019,(21):214-215.
- [3] 朱丽娜.计算机软工工程项目管理方法探析[J].科技风,2020,(02):100.
- [4] 谭琨,刘春.计算机软工工程项目管理方法探究[J].数字技术与应用,2021,39(09):222-224.
- [5] 张鹏.基于软件开发项目管理信息系统的几点思考[J].科技风,2022,(12):58-60.
- [6] 刘鹏,周霞,张毅等.浅谈软件项目管理[J].信息系统工程,2022,(04):81-84.
- [7] 唐铭晨,鲍珊珊,齐飞等.并行交互开发模型的设计及其应用[J].信息技术与标准化,2023,(11):74-78+85.
- [8] Saravanos A, Curinga X M. Simulating the Software Development Lifecycle: The Waterfall Model[J]. Applied System Innovation, 2023, 6(6).
- [9] Ibrahim M I. Iterative and Incremental Development Analysis Study of Vocational Career Information Systems[J]. International Journal of Software Engineering amp; Applications, 2020, 11(5):13-24.
- [10] 张良轩,姜潇,鲁丁等.快速原型法在信创项目中的应用研究[J].信息系统工程,2022,(06):145-148.
- [11] Jason L, Kiu Y M, Onalethata B, et al. Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability[J]. Sustainability, 2023, 15(2):1121-1121.
- [12] 张越.瀑布模型、快速原型模型和增量模型的对比[J].电子技术与软件工程,2019,(03):32.
- [13] 任禹洁.敏捷方法在项目管理中的应用[J].中国总会计师,2021,(10):116-117.
- [14] Alok M, Ibrahim Y A. Structured software development versus agile software development: a comparative analysis[J]. International Journal of System Assurance Engineering and Management, 2023, 14(4):1504-1522.
- [15] 钱平康.软件研发项目适用敏捷评估方法改进[J].中外企业文化,2023,(02):120-123.
- [16] 覃琳,杨程,孙卫宁等.基于敏捷开发下的协同创新管理服务平台能力提升[J].企业科技与发展,2019,(02):9-10.
- [17] 潘文卿,周善政,张鲁兵.敏捷开发过程中的质量提升[J].内燃机与配

- 件,2023,(11):93-95.
- [18] Gheorghe A, Gheorghe D I, Iatan L I. Agile Software Development[J]. Informatica Economica, 2020, 24(2):90-100.
- [19] Anchit S, Isha J, Nandita K, et al. A Systematic Review on Extreme Programming[J]. Journal of Physics: Conference Series, 2021, 1969(1).
- [20] 郭峰.极限编程中的关联性分析[J].数字技术与应用,2020,38(12):166-168.
- [21] 何晶.Scrum 敏捷方法在软件项目管理中的应用[J].数字技术与应用,2021,39(03):87-89.
- [22] Mario B, Salvatore B, Riccardo C, et al. Scrum for safety: an agile methodology for safety-critical software systems[J]. Software Quality Journal, 2022, 30(4):1067-1088.
- [23] Ahmad O M, Dennehy D, Conboy K, et al. Kanban in software engineering: A systematic mapping study[J]. The Journal of Systems Software, 2018, 13796-113.
- [24] 罗保发.看板管理系统在样机试制中的应用[J].时代汽车,2023,(14):128-130.
- [25] 叶至东.精益思想在汽车装配产线中的应用[J].时代汽车,2023,(19):25-27.
- [26] 张健,于婷婷.精益管理在生产中的运用探析[J].中小企业管理与科技(下旬刊),2020,(06):40-41.
- [27] Ibrahim M, Aftab S, Ahmad M, et al. Presenting and Evaluating Scaled Extreme Programming Process Model[J]. International Journal of Advanced Computer Science and Applications (IJACSA), 2020, 11(11).
- [28] Basit H, Rohaida R, Malina Z. Identifying components existing in Agile software development for achieving “light but sufficient” documentation[J]. Journal of Engineering and Applied Science, 2023, 70(1).
- [29] Colleoni J C, Josiane K, Dubugras D R, et al. Extending the Project Management Body of Knowledge (PMBOK) for Data Visualization in Software Project Management[J]. SN Computer Science, 2022, 3(4).
- [30] 张玉洁,胡海南,张帅等.看板管理应用的研究[J].大陆桥视野,2021,(03):78-79.
- [31] Mali S, L. M D. Systems Thinking Approach to Implementing Kanban: A case study[J]. Journal of Software: Evolution and Process, 2020, 33(4).
- [32] J. G, R. V, I. F, et al. Agile application approach with Scrum, Lean and Kanban [Enfoque de aplicación ágil con Scrum, Lean y Kanban][J]. Ingeniare, 2021, 29(1):141-157.
- [33] 汪厚俊.基于看板管理方法的敏捷软件开发[J].电子技术与软件工程,2016,(17):60.
- [34] 严华亮.论运营商基于敏捷开发的 IT 需求优先级管理[J].中国新通信,2017,19(15):15-16.



- [35] Popova O. Adaptation of flexible project management models based on Scrum and Kanban technologies[J]. Technology audit and production reserves, 2019, 4(2):4-10.
- [36] Shimza B, Rehman U S K, Shahid H, et al. A conceptual model supporting decision-making for test automation in Agile-based Software Development[J]. Data Knowledge Engineering, 2023, 144.
- [37] Yogeshwar S, Rashina H, Robert A. The role of the project manager in agile software development projects[J]. The Journal of Systems Software, 2021, 173110871.
- [38] Abozar K Z, Abbas A S, Shakiba K. The Effect of Innovation on the Company's Performance in Small and Medium-Sized Businesses with the Mediating Role of Lean: Agile Project Management Office (LAPMO)[J]. Complexity, 2023, 2023.
- [39] 蒋梦云.看板模型在敏捷软件开发流程中的应用[J].计算机与网络, 2019(12):19-21.
- [40] 傅永康.《项目管理知识体系指南》(第7版)前瞻[J].项目管理评论,2021(01): 46-52.
- [41] Cielo M G, Luise P, Ingo W, et al. Uses of business process modeling in agile software development projects[J]. Information and Software Technology, 2022, 152.
- [42] 蒋梦云.看板模型在敏捷软件开发流程中的应用[J].计算机与网络, 2019(12):19-21.
- [43] Eric W, A. C M, V. I R. An influence diagram approach to automating lead time estimation in Agile Kanban project management[J]. Expert Systems With Applications, 2022, 187.
- [44] 郭帅.基于敏捷方法的H公司软件开发项目管理优化研究[D].中国矿业大学,2023.
- [45] Gandomani J T, Tavakoli Z, Zulzalil H, et al. The Role of Project Manager in Agile Software Teams: A Systematic Literature Review[J]. IEEE Access, 2020, 81-1.
- [46] 郭睿.基于精益思想的互联网软件开发项目管理优化研究[D].北京邮电大学,2021.
- [47] 侯成.项目管理中下行沟通常见问题及对策[J].投资与合作,2022,(03):137-139.
- [48] 王斌.敏捷开发模式在软件工程项目中的应用[J].电子技术,2022,51(03):288-289.
- [49] Diebold P, Theobald S, Wahl J, et al. Stepwise transition to agile: From three agile practices to Kanban adaptation[J]. Journal of Software: Evolution and Process, 2019, 31(5):n/a-n/a.
- [50] BasitH, Rohaida R, Malina Z. Identifying components existing in Agile software development for achieving “light but sufficient” documentation[J]. Journal of Engineering and Applied Science, 2023, 70(1).
- [51] 张晓博.软件项目需求管理常见问题[J].合作经济与科技,2023,(20):133-135.
- [52] Elshandidy H, Mazen S, Hassanein E, et al. Using Behaviour-driven Requirements Engineering for Establishing and Managing Agile Product Lines[J]. International Journal

- of Advanced Computer Science and Applications (IJACSA), 2021, 12(2).
- [53] 沈春宝.浅谈应用软件的质量管理[J].金融电子化,2021,(05):86-87.
- [54] Dipendra G, Stuart C. The Impact of Agile Development Practices on Project Outcomes[J]. Software, 2022, 1(3):265-275.
- [55] Veronika H, Francisca N, Manuel W, et al. Job satisfaction in agile information systems development: A stakeholder perspective[J]. Information and Software Technology, 2023, 163.

## 附录 1 德尔菲专家问卷

尊敬的先生/女士您好：

非常感谢您在工作之余参加本次问卷调研。本次问卷采用匿名形式，调研结果将用于调研者论文写作，请各位专家坦率分享。请您结合自身的工作经验、专业知识来对本次问卷进行认真填写，衷心感谢您的帮助！（感谢支持，建议一周内完成）

### 1、背景信息

现行项目管理实践：M 公司目前现行的是瀑布开发项目管理流程，请将公司近半年内交付的大小项目纳入考虑范畴。

### 2、调研目的

识别问题、收集建议、理解管理层的期望。希望调研后可以收集关于目前软件开发项目管理过程中的问题及预期优化建议。

### 3、问卷内容

为了不出现引导性的问答，本次问卷采用开放式调查问卷。通过问卷集思广益，让各位专家给出在各自工作中发现的问题。

您的性别：\_\_\_\_\_

您的学历：\_\_\_\_\_

您的职位：\_\_\_\_\_

工作年限：\_\_\_\_\_

问题：您认为本公司研发中心软件项目管理中存在哪些问题需要优化？请指出问题成因是什么？

问题回复：

---

Tips:

比如：项目计划进度方面、需求管理方面、项目变更控制方面、研发流程问题、测

试质量方面、人员沟通协作方面、业务方面、项目管理模式方面、项目管理工具方面、组织架构方面等等。

您的回答可以从以下三方面给出：

- 1、识别当前项目管理过程中遇到的主要问题。
- 2、探索这些问题产生的原因。
- 3、征集可能的解决方案。

## 附录 2 问题评价重要性问卷

尊敬的先生/女士您好：

非常感谢您在工作之余参加本次问卷调研。本次问卷采用匿名形式，调研结果将用于调研者论文写作，请各位专家坦率分享。请您结合自身的工作经验、专业知识来对本次问卷进行认真填写，衷心感谢您的帮助！（感谢支持，建议一周内完成）

您的性别：\_\_\_\_\_

您的学历：\_\_\_\_\_

您的职位：\_\_\_\_\_

工作年限：\_\_\_\_\_

背景信息：

为了深入理解并解决在前期的专家问卷中收集到的问题，并比较各个问题的相对重要性。故引用下表 1~9 标度法判断，标度值含义如下表所示。

标度绝对值	标度值含义
1	前者因素与后者因素同样重要
3	前者因素比后者因素稍微重要
5	前者因素比后者因素明显重要
7	前者因素比后者因素强烈重要
9	前者因素比后者因素极端重要
2、4、6、8	前者因素比后者因素的重要程度介于上述相邻的中间值
以上数值倒数	两因素的互换位置的比较是原位置比较的倒数

Tips:

在使用 1~9 标度法进行评价时，请各位专家注意，您需要对决策矩阵中的因素进行两两比较，这包括横行和纵列中的每个因素，具体步骤如下。

两两比较：对于决策矩阵中的每一对因素（即每行与每列的交叉点），请根据这两个因素在决策过程中的相对重要性，使用 1 至 9 的标度给出您的评价。标度的选择区

间为从 1（两者同等重要）到 9（一方极端重要）。

填写标度值：每次两两比较确定的相对重要性标度值，请填写在对应的矩阵位置。如果因素 A 相对于因素 B 的重要性为 3，那么因素 B 相对于因素 A 的重要性为 1/3。在进行问题因素比较时，请尽可能保持评价的逻辑一致性。

在评价过程中，请结合自身的工作经验和专业知识进行判断。如果在评价过程中有疑问或需要进一步的说明，可以随时提出，衷心感谢各位专家的配合。

专家问题回复：

问题 1：以下四个因素分别进行横行和纵列因素两两对比，请比较每个因素对公司开发项目管理流程的影响程度。

因素	需求管理	项目过程	研发测试	组织架构
需求管理	1			
项目过程		1		
研发测试			1	
组织架构				1

问题 2：以下四个因素分别进行横行和纵列因素两两对比，请比较每个因素对需求管理的影响程度。

因素	需求优先级定义不明确	提需缺乏中长期规划	需求文档设计不清晰	需求变更频繁难管理
需求优先级定义不明确	1			
提需缺乏中长期规划		1		
需求文档设计不清晰			1	
需求变更频繁难管理				1

问题 3：以下四个因素分别进行横行和纵列因素两两对比，请比较每个因素对项目过程的影响程度。

因素	项目过程缺乏有效监 控	项目成员沟通协作困 难	项目并行导致资源冲 突	项目管理方法不被认 可
项目过程缺乏有效监 控	1			
项目成员沟通协作困 难		1		
项目并行导致资源冲 突			1	
项目管理方法不被认 可				1

问题 4：以下四个因素分别进行横行和纵列因素两两对比，请比较每个因素对研发测试的影响程度。

因素	质量管理体系不完善	缺少问题复盘归纳	开发工作量难以评定	流程风险识别控制 不足
质量管理体系不完善	1			
缺少问题复盘归纳		1		
开发工作量难以评定			1	
流程风险识别控制不 足				1

问题 5：以下三个因素分别进行横行和纵列因素两两对比，请比较每个因素对组织架构的影响程度。

因素	组织人员利用率不高	项目管理和职能管理冲突	部门成员权责划分不明确
组织人员利用率不高	1		
项目管理和职能管理冲突		1	
部门成员权责划分不明确			1