

分类号：

U D C：

密级：

学号：410203020068

南 昌 大 学 专 业 学 位 硕 士 研 究 生

## 学 位 论 文

**基于敏捷开发的 T 公司软件项目管理优化研究**

**Research on Optimization of Software Project Management  
of T Company Based on Agile Development**

涂剑英

培养单位（院、系）：经济管理学院

指导教师姓名、职称：吴伟伟 副教授

专业学位种类：工商管理硕士

专业领域名称：

论文答辩日期：2023 年 5 月 20 日

答辩委员会主席：\_\_\_\_\_ 谌丰毅 \_\_\_\_\_

评阅人：\_\_\_\_\_

2023 年 5 月 26 日

## 摘 要

互联网时代科学和技术日新月异飞速发展,变化才是永恒不变的,自从国外 17 位软件开发领域的领军人物提出敏捷开发的概念,我国很多软件企业开始将敏捷开发应用到软件项目管理中,希望通过敏捷开发来解决软件项目面临的挑战,这些挑战包括开发效率低、产品质量差、需求变化快、用户不满意等。敏捷开发在我国软件企业中已经被证明可以有效缓解软件项目中的一些挑战,并且已经成为高效开发团队的制胜法宝。然而敏捷开发不是万能的,它并不能解决软件项目所面临的所有挑战,因为敏捷开发不是一套行之有效的方法,当然也不存在十全十美的方法,生搬硬套往往达不到好的效果,如何找到适合企业自身的方法成了目前软件企业最为关注的问题。

国内关于敏捷开发的研究有很多,但是大部分都集中于将敏捷开发应用到企业软件项目中的研究,少有文献对已经应用了敏捷开发的项目,如何进一步优化研究。T 公司是一家为电网及能源行业客户提供软件信息化产品和服务的软件企业,本论文以 T 公司的软件项目作为研究对象,主要思路是针对该研究对象,按照找出问题、分析问题、提出优化方案、实施优化方案、实施效果评价的思路来进行研究和实践的。

首先,通过阅读大量的文献资料,对项目管理、软件项目管理和敏捷开发的基础理论进行了概述,为后续的研究提供了理论基础。其次,对 T 公司已经应用敏捷开发的软件项目管理进行了研究,先通过访谈的形式找出当前 T 公司存在的问题,再利用层次分析找到主要问题,然后通过原因鱼骨图分析出这些主要问题出现的根本原因。继而,根据这些主要问题的根本原因,提出了针对 T 公司在需求管理、质量管理和成员间沟通不畅这三个主要问题的优化方案,改进 T 公司软件项目管理流程。最后,以 T 公司 D 项目为案例,跟踪优化方案在该项目的实施过程,并进行效果分析。

本论文希望可以通过对已经应用了敏捷开发的 T 公司的软件项目进行分析,总结出一些对 T 公司软件项目管理实用有效的优化方案,解决该公司的实际问题,提升研发效率和提高产品质量,同时也为同行业的其他企业提供一些参考借鉴。

**关键词：**项目管理；软件项目管理；敏捷开发；Scrum；DevOps

## ABSTRACT

Science and technology are developing rapidly in the Internet era, so there is nothing permanent except change. Since 17 foreign leading figures in the field of software development put forward the concept of agile development, many software enterprises in China have begun to apply agile development to software project management and they hoped to solve the challenges of the software projects by agile development. These challenges include low development efficiency, poor product quality, rapid demand change, dissatisfied users and so on. Agile development has been proved to alleviate some challenges effectively in software projects in China's software enterprises, so it has become a magic weapon for success of efficient development teams. However, agile development is not everything, it cannot solve all the challenges of software projects because agile development is not an effective method. Of course, there is no perfect method. We should not apply mechanically the method of agile development. From the enterprise itself, how to find a suitable method has become the most concerned problem of the software enterprises at present.

There are many researches on agile development in China, but most of them are focused on the application of agile development to enterprise software projects. Few literatures have studied how to further optimize the projects applied the agile development. T company is a software enterprise, which provides software information products and services for customers in the power grid and energy industry. This thesis takes the software project of T company as the research object. The main idea is to find out problems, analyze problems, propose optimization schemes, implement optimization schemes, and evaluate the implementation effect.

First of all, this thesis summarizes the basic theories of project management, software project management and agile development through reading a large number of literature, which provides a theoretical basis for subsequent research. Secondly, the software project management of T company applied agile development has been studied. I acquire the existing problems of T company by interviews; I gain the main problems by using the analytic hierarchy process; I obtain the basic reason of these

## ABSTRACT

---

main problems by fishbone diagram. Then, I put forward optimization solutions according to the basic causes: demand management, quality management and poor communication among members. These optimization schemes improve the software project management process of T company. Finally, taking the D project of T company as a case, I track the implementation process and analyze the implementation effect.

This thesis hopes to summarize some practical and effective optimization schemes for the software project management of T Company through the research and analysis of the software projects of T Company applied agile development. It can solve the actual problems of the company; it can improve the development efficiency and the product quality; it also can provide some reference for other enterprises in the same industry.

**Key Words:** project management; software project management; agile development; Scrum; DevOps

# 目 录

第 1 章 导论.....	1
1.1 选题背景及意义.....	1
1.1.1 选题背景.....	1
1.1.2 选题意义.....	3
1.2 研究的内容和框架.....	3
1.2.1 研究内容.....	3
1.2.2 研究框架.....	3
1.3 研究的思路和方法.....	5
1.3.1 研究思路.....	5
1.3.2 研究方法.....	5
1.4 国内外研究现状.....	5
1.4.1 国外研究现状.....	5
1.4.2 国内研究现状.....	7
1.4.3 文献评述.....	8
第 2 章 相关理论概述 .....	10
2.1 项目管理概述.....	10
2.1.1 项目与项目管理.....	10
2.1.2 项目管理知识体系.....	11
2.2 软件项目管理概述.....	13
2.2.1 软件项目.....	13
2.2.2 软件项目管理.....	13
2.3 敏捷开发概述.....	17
2.3.1 敏捷开发的起源.....	18
2.3.2 敏捷开发简介.....	18
2.3.3 敏捷开发优势.....	19
2.3.4 敏捷开发常用方法.....	20
第 3 章 基于敏捷开发的 T 公司软件项目管理现状与存在的问题....	25
3.1 T 公司简介 .....	25
3.1.1 T 公司基本情况 .....	25
3.1.2 T 公司组织架构 .....	25
3.2 基于敏捷开发的 T 公司软件项目管理现状.....	26
3.2.1 基于敏捷开发的软件项目管理流程.....	26
3.2.2 基于 Azure DevOps 工具管理软件项目.....	28
3.3 基于敏捷开发的 T 公司软件项目管理存在的问题.....	31

3.3.1 企业相关员工访谈.....	31
3.3.2 基于层次分析法定量分析.....	32
3.3.3 各层因素权重及总排序.....	37
3.4 基于敏捷开发的 T 公司软件项目管理问题原因分析.....	38
3.4.1 基于原因鱼骨图对主要问题的原因分析 .....	38
3.4.2 问题原因总结.....	42
第 4 章 基于敏捷开发的 T 公司软件项目管理优化方案.....	44
4.1 项目需求管理优化方案.....	44
4.1.1 需求管理流程优化.....	44
4.1.2 提升需求人员能力.....	47
4.1.3 建立需求跟踪流程.....	47
4.2 项目质量管理优化方案.....	48
4.2.1 质量管理流程优化.....	48
4.2.2 增加测试内容.....	50
4.2.3 增加需求验收文档.....	51
4.3 成员之间沟通优化方案.....	52
4.3.1 沟通管理流程优化.....	52
4.3.2 培养沟通意识.....	54
4.3.3 提升沟通能力.....	55
第 5 章 优化方案的实施与预期效果 .....	56
5.1 T 公司 D 项目介绍 .....	56
5.2 优化方案启动.....	56
5.2.1 确定团队共同目标.....	56
5.2.2 沟通专项会议.....	57
5.2.3 管理制度的贯彻.....	57
5.3 优化方案实施.....	59
5.3.1 现场人员工作优化.....	60
5.3.2 产品经理工作优化.....	61
5.3.3 需求人员工作优化.....	62
5.3.4 研发人员工作优化.....	65
5.3.5 测试人员工作优化.....	66
5.4 预期实施效果.....	68
5.4.1 产品质量提升.....	68
5.4.2 团队沟通良好.....	68
5.4.3 员工抱怨减少.....	69
5.4.4 团队工作效率提高.....	69
5.4.5 用户满意度提升.....	69
第 6 章 结论与展望 .....	71

6.1 结论.....	71
6.2 展望.....	72
参考文献.....	73
附录 A Excel 第二层次分析法计算公式 .....	76



## 插图清单

图 1.1 Standish Group 的 CHAOS Reports 不同年份统计 .....	2
图 1.2 研究框架 .....	4
图 2.1 瀑布开发 .....	14
图 2.2 增量开发 .....	15
图 2.3 快速原型开发 .....	15
图 2.4 螺旋开发 .....	16
图 2.5 敏捷开发 .....	16
图 3.1 T 公司的组织结构 .....	25
图 3.2 T 公司软件项目管理流程 .....	26
图 3.3 T 公司使用 Azure DevOps 工具进行敏捷开发管理流程 .....	28
图 3.4 任务面板示意图 .....	30
图 3.5 T 公司软件项目管理问题层次结构模型 .....	33
图 3.6 需求管理问题的鱼骨图 .....	39
图 3.7 质量管理问题的鱼骨图 .....	40
图 3.8 沟通不畅的鱼骨图 .....	41
图 3.9 T 公司需求的传递关系 .....	42
图 4.1 需求分析基本原则 .....	45
图 5.1 T 公司优化后软件项目管理流程 .....	60
图 5.2 Azure DevOps 工具的原始需求 .....	61
图 5.3 需求分析流程 .....	63
图 5.4 产品原型 .....	64
图 5.5 Azure DevOps 工具的用户情景 .....	65
图 5.6 用户情景拆分子任务 .....	66
图 5.7 测试环境部署变化 .....	67
图 5.8 产品版本问题单统计 .....	68

## 附表清单

表 2.1 PMBOK 的 5 大过程组 .....	11
表 2.2 PMBOK 的 10 大知识领域 .....	11
表 2.3 PMBOK 的 5 大过程组、10 大知识领域、49 个模块之间的关系 .....	12
表 2.4 软件开发模型比较 .....	17
表 2.5 敏捷开发 4 个价值观 .....	18
表 2.6 敏捷开发 12 条原则 .....	19
表 2.7 Scrum 框架的“33355” .....	21
表 2.8 精益开发 7 个原则 .....	21
表 2.9 极限编程 4 个价值、5 个原则、12 个核心实践 .....	22
表 2.10 DevOps 关键实践 .....	23
表 2.11 敏捷开发方法比较 .....	24
表 3.1 Azure DevOps 的 5 个主要模块 .....	28
表 3.2 访谈对象清单 .....	31
表 3.3 访谈收集软件项目管理问题清单 .....	31
表 3.4 T. L. Saaty 1~9 标度法 .....	32
表 3.5 第二层矩阵 .....	34
表 3.6 第三层需求管理问题 A 判断矩阵 .....	34
表 3.7 第三层质量管理问题 B 判断矩阵 .....	34
表 3.8 第三层沟通管理问题 C 判断矩阵 .....	34
表 3.9 第三层进度管理问题 D 判断矩阵 .....	34
表 3.10 组织管理问题 E 判断矩阵 .....	35
表 3.11 随机一致性指标 RI 取值表 .....	35
表 3.12 第二层判断矩阵权重计算结果 .....	36
表 3.13 第三层需求管理问题 A 判断矩阵权重计算结果 .....	36
表 3.14 第三层质量管理问题 B 判断矩阵权重计算结果 .....	36
表 3.15 第三层沟通管理问题 C 判断矩阵权重计算结果 .....	36
表 3.16 第三层进度管理问题 D 判断矩阵权重计算结果 .....	36

表 3.17 第三层组织管理问题 E 判断矩阵权重计算结果.....	37
表 3.18 权重总排序表 .....	37
表 4.1 敏捷冲刺代办列表 .....	46
表 4.2 需求跟踪管理表的具体描述 .....	47
表 4.3 项目规范作用 .....	50
表 4.4 项目规范内容 .....	50
表 5.1 公司敏捷团队定期沟通会议 .....	58
表 5.2 在线需求跟踪矩阵 .....	58
表 5.3 在线问题跟踪记录 .....	58
表 5.4 在线会议沟通跟踪记录 .....	59
表 5.5 在线培训跟踪记录 .....	59
表 5.6 T 公司 D 项目需求优先级排序 .....	62
表 5.7 敏捷团队成员抱怨减少的内容项 .....	69
表 5.8 用户满意度得分 .....	70

## 第1章 导论

### 1.1 选题背景及意义

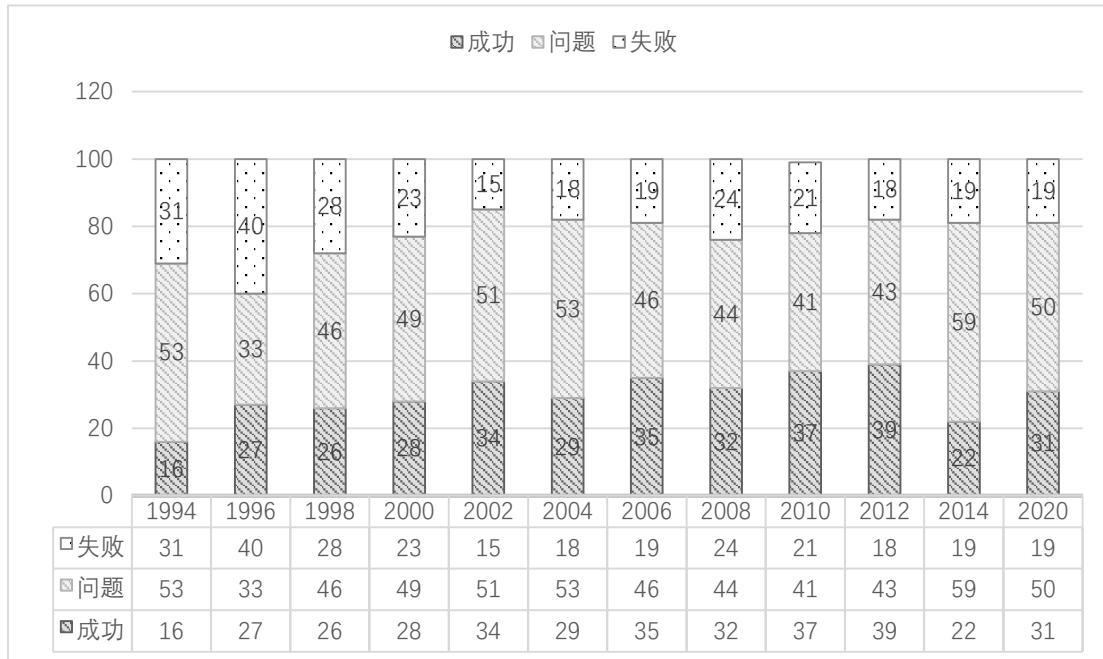
#### 1.1.1 选题背景

21 世纪互联网行业 and 信息技术日新月异飞速发展，变化才是永恒不变的，我国软件市场在这种环境下迅猛发展，企业之间的竞争也进入白热化阶段，为了能在软件市场取得一席之地，提高企业自身核心竞争力，软件企业必须快速响应客户需求、降低生产成本、提高生产效率和交付有市场价值的产品。2001 年 17 位软件开发领域的领军人物提出敏捷开发的概念，敏捷开发立马席卷全球，受到大中小软件企业的青睐，掀起应用敏捷开发到软件行业的热潮。在数字化转型公司（digital.ai）发布的《2021 年敏捷年度状态报告》一文中指出，各行各业的敏捷采用率都出现了大幅度增长，软件开发领域从 2020 年的 37% 增长到 2021 年的 86%，甚至财务、人力资源、市场营销等的敏捷采用率也翻了一番。

T 公司是一家为电网及能源行业客户提供软件信息化产品和服务的软件企业，企业的软件项目主要是使用敏捷开发来进行管理的。敏捷开发以尽快交付可用工作的软件作为目标、拥抱变化、面对面沟通、减少不必要的工作和迭代增量交付。该企业自从引进敏捷开发来进行软件项目管理后，在产品发布时间、开发效率、产品质量、客户满意度方面都有明显提升，但是在使用敏捷开发的过程中还是存在一些问题，如需求质量不高、交付的产品存在质量问题、定制化项目开发导致功能无法重用、团队成员之间沟通存在问题导致工作出现偏差等问题。因此，解决企业项目管理中的问题，优化敏捷开发流程，是当前企业首当其冲的工作。

敏捷开发不是万能的，并不能解决软件项目管理的所有问题。禅道在《2021 年 IT 行业项目管理调查报告》中指出项目管理最常遇到的挑战是需求变更，其次是人力不足、交流沟通不畅和时间不足。根据 Standish Group 发布的 2020 年 CHAOS Reports 显示，能够按期交付的成功项目比例仅为 31%，有问题的项目比例为 50%，失败的项目比例为 19%。图 1.1 所示为 Standish Group 不同年份项

目统计的 CHAOS Reports。



资料来源：Standish Group 的 CHAOS Reports。

图 1.1 Standish Group 的 CHAOS Reports 不同年份统计

Standish Group 发布的 2020 年 CHAOS Reports 显示还分别对敏捷项目和瀑布项目进行了分别统计，敏捷项目的成功比例为 42%、问题比例为 47%、失败比例为 11%，瀑布项目的成功比例为 13%、问题比例为 59%、失败比例为 28%。我们可以看出敏捷项目成功率是瀑布项目的 3 倍，失败率不到瀑布项目的一半，可见应用敏捷开发的确帮助项目提升了成功的概率。尽管如此，仍有超过一半的敏捷开发项目是有问题的、乃至失败的，这是什么原因导致的呢？

敏捷开发的确是一种软件开发领域比较优秀的实践，能够帮助企业提高交付效率，缩短交付周期，发布更好的产品。但是敏捷开发并不是万能的，在使用敏捷开发的过程中，项目团队需要根据项目实际情况及实际遇到的问题，不断优化改进，找到适合自己团队的敏捷开发，才能达到提高企业软件开发效率的目的，满足用户不断变化的需求，提高产品质量，交付用户满意的产品。

因此，本论文将针对已经应用了敏捷开发的 T 公司存在的项目管理问题进行分析，并给出优化方案。

### 1.1.2 选题意义

本论文选题的意义在于通过分析 T 公司使用敏捷开发进行软件项目管理的现状以及存在的问题，将敏捷开发与软件项目管理各阶段管理相结合，从中找到有效的改进方案，从而提高项目团队的研发效率，进一步缩短产品的交付周期，提升企业研发效率，增强企业竞争力。希望通过对 T 公司敏捷开发在软件项目管理的优化，解决该公司的实际问题，提升研发效率和产品质量，同时也为同行业的其他企业提供一些参考借鉴。

## 1.2 研究的内容和框架

### 1.2.1 研究内容

本论文以 T 公司软件项目作为研究对象，针对其使用敏捷开发来进行软件项目管理现状，结合项目管理、软件项目管理和敏捷开发的基本理论，利用文献研究法、访谈法、层次分析法和原因鱼骨图分析法找到 T 公司软件项目管理中应用敏捷开发存在的问题并对其进行原因分析，根据问题根本原因找到解决问题优化方案，最后通过优化方案来改进 T 公司的软件项目管理流程。

### 1.2.2 研究框架

本论文是基于敏捷开发的 T 公司软件项目管理的优化研究，由六章组成。

第一章，导论，主要介绍本论文的选题背景和意义，研究内容和框架，总结分析国内外相关文献，并在此基础上，给出本论文的研究思路和研究方法。

第二章，相关理论概述，介绍了项目管理、软件项目管理和敏捷开发相关理论。

第三章，基于敏捷开发的 T 公司软件项目管理现状与存在问题分析，本章介绍了 T 公司，总结了 T 公司使用敏捷开发进行软件项目管理现状，就存在的问题进行分析。

第四章，基于敏捷开发的 T 公司软件项目管理优化方案，本章针对 T 公司使用敏捷开发进行软件项目管理存在的问题，从软件项目管理的需求管理、质量

管理、成员间沟通不畅三个方面进行改进优化。

第五章，优化方案的实施与预期效果，本章跟踪 T 公司 D 项目实施方案以及实施效果评价。

第六章，结论与展望，这一章是总结本文研究的结论和存在的不足，并展望未来研究方向。

根据本论文研究内容，研究框架如图 1.2 所示。

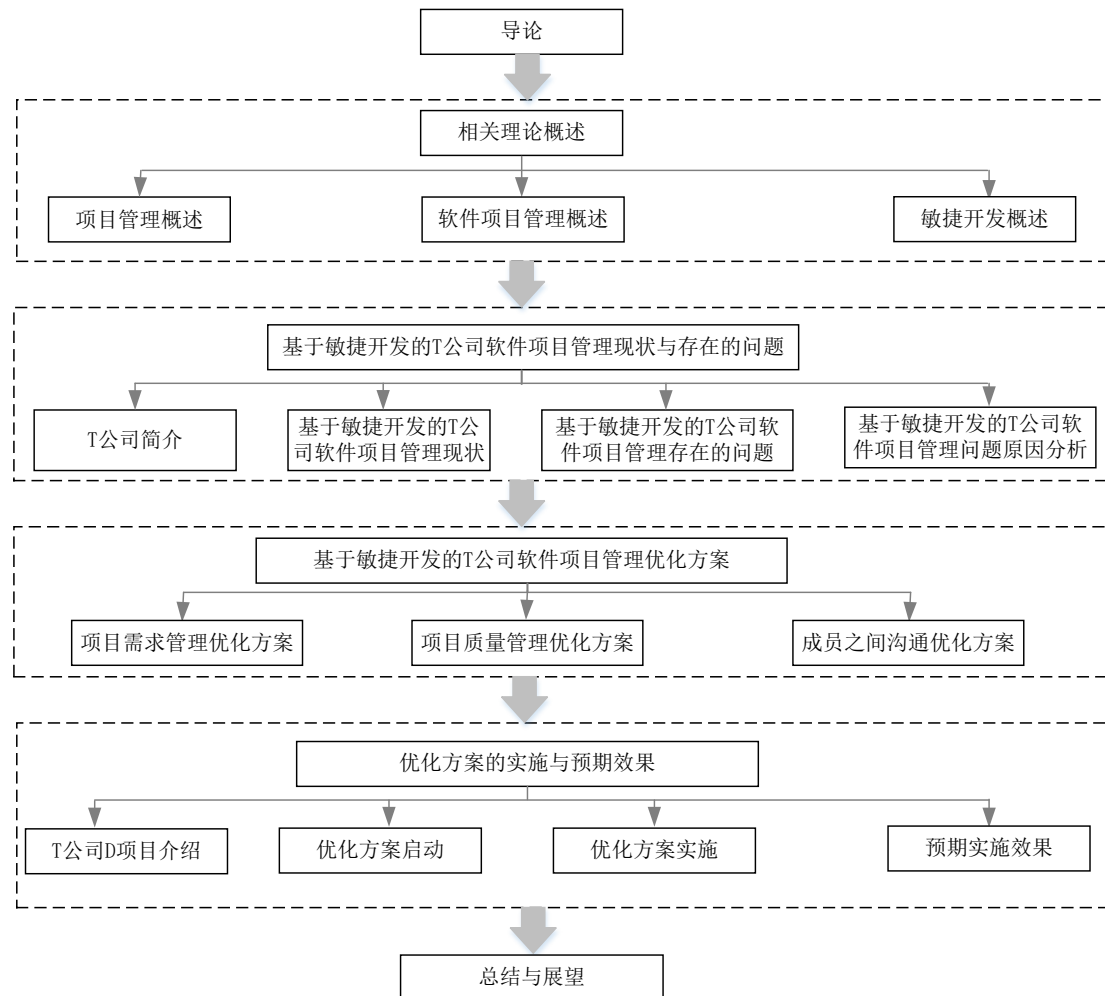


图 1.2 研究框架

## 1.3 研究的思路和方法

### 1.3.1 研究思路

本论文的研究思路是先收集和整理国内外关于软件项目管理、敏捷开发、敏捷开发在软件项目管理中的应用和优化的文献和资料，总结敏捷开发在软件项目管理中应用的研究成果，然后分析 T 公司使用敏捷开发过程中存在的问题及其根本原因，再通过参考整理的敏捷开发相关的文献和基础理论，找出适合于 T 公司敏捷开发的优化方案，最后对优化方案进行预实施及效果分析。

### 1.3.2 研究方法

本论文采用的研究方法包括：

（1）层次分析法。通过层次分析法建立判断矩阵并计算问题严重程度的权重，从中找出 T 公司存在的主要问题。

（2）原因鱼骨图分析法。通过原因鱼骨图分析法找到 T 公司软件项目管理中使用敏捷开发遇到的主要问题的根本原因。

## 1.4 国内外研究现状

我国很多软件企业采用敏捷开发来进行软件项目管理，并且在快速响应客户需求和快速的交付产品方面都有了显著提升，但在使用过程中也遇到各种问题。如何更好地将敏捷开发应用到软件企业的软件项目管理中，识别软件项目管理中成功的关键因素，优化敏捷开发实践，以用最小的成本和时间开发出高质量的软件，从而提升企业的竞争力，是每个软件企业的目标，也是本论文研究的目的。国内外众多学者在使用敏捷开发应用到软件项目管理中进行了不少研究，为本论文研究提供了有益思路。

### 1.4.1 国外研究现状

国外很多学者通过对正在实施敏捷开发的团队进行研究，发现人对于敏捷开发的影响非常重大。Gren et al.（2020）通过研究发现敏捷团队成熟度与敏捷



开发的两个核心实践迭代开发和迭代回顾有较强的关联性，敏捷团队的成熟度级别越高，就可以更好地使用迭代开发和迭代回顾<sup>[1]</sup>。Amorim et al. (2020) 主张一个快乐的工作环境会让敏捷团队成员更积极乐观，工作更加满意，作者通过调查、访谈和交叉分析的方法，提出了一些影响敏捷团队成员的幸福因素，包括有效的沟通、主动热情的成员、成员之间的协作及团队领导<sup>[2]</sup>。Tam et al. (2020) 通过研究一些成功的敏捷开发软件项目的人为因素，把成本、时间和客户满意度作为成功的衡量标准，发现团队能力和客户参与协作是敏捷开发项目成功的主要因素<sup>[3]</sup>。Cram (2019) 从敏捷采用、敏捷剪裁、敏捷使用三个敏捷不同阶段提出了很多敏捷实践经验，目的在于给即将或正在使用敏捷开发的团队提供指南<sup>[4]</sup>。

很多学者对于敏捷开发中的需求管理和质量管理依旧非常关注，如需求如何管理，需求的优先级如何排序，研发质量如何提升等都是学者们在不停探索的问题。Kuengjai et al. (2021) 指出需求优先级对于软件质量和软件开发的成功是至关重要的，对需求优先级的方法进行了比较，提出累计投票、二叉搜索树、层次分析法是最准确最可靠的技术，作者还分析出了一些有助于指导需求优先级排序因素，这些因素包括利益相关者、利益相关者的关注程度、组织文化等<sup>[5]</sup>。Shim et al. (2019) 指出尽管敏捷开发已成为主流的开发方法，但是敏捷开发中的需求如何管理依旧是一个很复杂的问题，作者通过用户影响地图和进度看板这两个可视化的工具来进行需求管理，首先通过用户影响地图识别用户的需求和痛点，然后在进度看板来跟踪需求研发进度，并在每次迭代后，观察用户对产品的反应，及时验证和更新用户影响地图<sup>[6]</sup>。Arcos-Medina et al. (2019) 提出质量在软件工程中的重要性今天已得到广泛关注，敏捷开发在软件开发过程中结合了质量实践，以快速实现高质量的软件，并与快速变化的客户需求保持一致<sup>[7]</sup>。

还有学者对敏捷开发中的是否需要文档进行了分析研究。Mathews et al. (2020) 指出敏捷宣言没有给出任何关于敏捷开发中文档的具体指导方法，并且敏捷宣言的第二个原则“工作的软件高于详尽的文档”被大家误解为敏捷开发是不需要文档。而在实际敏捷软件开发过程中，屡遭诟病的没有文档或者文档不足，严重影响了软件质量，作者提出尽管敏捷文档的完全标准化是不可能的，但是每个敏捷团队可以制定适合自己团队文档，以确保项目的宝贵经验不会流失掉，同时也不影响软件质量<sup>[8]</sup>。

### 1.4.2 国内研究现状

由于敏捷开发是从国外引入到我国的软件项目管理中的,我国研究学者大部分的研究是如何将敏捷开发应用到我国的企业中的。丁慧等(2021)将北京京航计算通讯研究所作为研究对象,强调了敏捷思想的核心价值观和敏捷开发的12条原则,总结了敏捷开发在项目管理中的一些实践经验,包括迭代计划会议、每日站立会议、可视化管理、迭代评审会、迭代反思会等<sup>[9]</sup>。何晶(2021)提出了管理团队的构建、对开发流程重新规划、利用 Scrum 工件管理、计划会议机制和项目管理辅助工具如 JIRA 等基于敏捷开发的软件项目管理方案<sup>[10]</sup>。王斌(2022)指出对于中小软件企业,只有保持稳定的开发步骤,保持稳定的开发团队,避免人员流动,团队成员彼此默契,敏捷开发才能产出高效生产率<sup>[11]</sup>。

敏捷开发在我国应用一段时间后,有一些学者开始提出了一些改进,怎么样才能让敏捷开发更适应我国的软件企业呢。杨帆(2015)针对敏捷开发过程中出现的问题进行了研究,提出了一套应对敏捷开发难以实施的对策,包括扁平化组织结构、加强员工技能培训、减少敏捷团队成员的干扰因素、并提供加薪奖金升职机会,来保障敏捷开发的有效实施<sup>[12]</sup>。王丹(2018)以 DD 公司企业为例,作者将公司从瀑布式开发模式转变到敏捷开发模式的过程中遇到问题通过运用层次分析法,建立判断矩阵并计算权重的方式,找到了实施敏捷开发问题的严重程度级别,再在此基础上提出针对性的优化策略<sup>[13]</sup>。黄欣(2018)提出了一套相对完整的行动框架,先敏捷开发实践、再对实践过程进行数据度量分析,然后摸索新过程改进的方法,再循环继续实践摸索出的新方法,最后通过这种反复实践,找到适合项目团队的敏捷开发实践<sup>[14]</sup>。张东红等(2021)提出了改进后的 Scrum 模型,即在冲刺计划会议上除了构建定时任务外,还要构建创新任务<sup>[15]</sup>。

除了敏捷开发应用到企业项目中的实践研究,还有很多学者提出将敏捷开发与其他方法结合在一起使用,效果更佳。汪厚俊(2016)主张将看板管理和敏捷开发结合的方案设计,看板流程让工作更透明,更适合跟踪需求变更和用户故事调整,以及进度的跟踪<sup>[16]</sup>。黄晨晨(2019)提出通过精益思想发现和消除浪费,达到价值的最大化,再结合看板方法和敏捷开发方法优化现有管理流程,建立学习型组织团队,改善团队协作,提高工作效率<sup>[17]</sup>。闫鹏斌等(2020)指出了敏捷软件开发只适合小型敏捷团队迭代增量开发,缺乏对整个软件组织的持续改进和优化,而 CMM/CMMI 能力成熟度模型可以作为敏捷开发的补充,因为其具备

指导软件开发过程的改进和评估软件开发能力，以促进软件团队整体能力的提升<sup>[18]</sup>。黄敏珍（2020）主张将 CMMI、敏捷开发和 DevOps 进行融合，充分发挥各自所长，为软件企业量身定制特色管理体系<sup>[19]</sup>。龚桂芬等（2022）提出引入 DevOps 软件开发模型，采用敏捷开发对项目进行管理，使用自动化工具进行持续集成和持续部署，对开发、测试、系统运维等进行全过程化管理<sup>[20]</sup>。

很早就有学者提出了软件项目管理中的需求管理、质量管理和沟通管理是非常重要的。王海军（2015）强调了需求管理在软件项目管理中的重要地位，需求管理的好坏直接影响到软件项目的成败<sup>[21]</sup>。杨东（2021）指出质量管理也是软件项目成功的重要指标，产品质量是软件企业的竞争力和生命力的源泉<sup>[22]</sup>。王蕾（2021）以某市管理平台的建设为例，提出了良好的沟通管理对软件项目的成败起着关键作用，不少失败的项目案例都是因为沟通不畅、信息不对称而导致失败的<sup>[23]</sup>。

如何在敏捷开发中做好软件项目管理的需求管理、质量管理和沟通管理，我国学者也做了一些研究，但是并不多。杨俊（2021）提出了敏捷需求管理的基本原则“不是看怎么做，而是看为什么做”<sup>[24]</sup>。彭昊（2021）主张通过标签化的方式进行需求管理，基于马斯洛需求层次理论及卡诺模型，通过量化的方式确定需求的优先级<sup>[25]</sup>。严华亮（2017）提出了将市场价值、损失、工作量、安全性作为敏捷开发需求优先级判断的指标<sup>[26]</sup>。孙子谦等（2016）提出将戴明环（PDCA）应用到基于敏捷开发的软件质量管理中<sup>[27]</sup>。束勉等（2018）提出了一些有效的质量管理措施，包括各项目任务全员参与、QA 角色由检查者转变为指导者、引入自动化测试、收集反馈不断总结改进、灰色发布等，从而提高产品质量<sup>[28]</sup>。黄凌凡（2008）强调了沟通过程透明性，并且需要随时记录沟通过程和结果，及时将项目进展通报给相关干系人，营造良好的项目团队沟通氛围<sup>[29]</sup>。

### 1.4.3 文献评述

综上所述，国内的敏捷开发发展虽然呈现了稳步发展的态势，但是与国外水平相比存在明显差距。

国外大部分企业都应用了敏捷开发，所以国外学者们研究的方式是通过不同企业，不同团队进行调研和分析，得出敏捷团队是有成熟度级别的，敏捷团队处于不同成熟度阶段使用的敏捷实践经验是不同的，敏捷开发项目成功的因

素是敏捷团队能力和用户参与，快乐的工作环境会让敏捷团队成员更幸福。除此之外，还将需求优先级的各种方法进行了比较，得出累计投票、二叉搜索树、层次分析法是最准确最可靠的方法，需求可视化管理可以帮助我们对需求较直观的管理，强调了质量的重要并给出了很多敏捷开发的质量实践，同时提出了敏捷开发中比较有争议的文档的问题，写不写文档，文档写到什么程度还是要根据每个团队自己决定，并没有统一的规范。本论文也将根据 T 公司的实际情况，增加必要的文档。

尽管敏捷开发的概念提了这么多年，我国国内很多企业还处在将敏捷开发应用到软件企业项目管理的阶段，即从传统瀑布式开发转变到敏捷开发，因此我国学者研究较多的是如何应用敏捷开发到不同行业的软件企业中，还有将敏捷开发与精益开发、看板管理、CMM/CMMI、DevOps 结合在一起使用的敏捷实践研究，而关于已经应用敏捷开发的软件企业，如何进行优化的研究文献还比较少。所以本论文将以已经应用了敏捷开发的 T 公司作为研究对象，并对该公司的软件项目管理进行优化研究。

## 第2章 相关理论概述

### 2.1 项目管理概述

#### 2.1.1 项目与项目管理

项目是为创建一个独特的产品，一种独特的服务，一项独特的成果，一个或多个产品、服务或成果的独特组合而进行的临时性的工作<sup>[30]</sup>。项目和一般的组织活动是有区别的，项目具备以下基本特征<sup>[30]</sup>：

（1）目标性。每一项目都具有强烈的目标性，目标可能是产品、服务或者成果。

（2）相关性。每一个项目都有很多相互关联的任务，有些任务需要串行来完成，有些任务可以并行开展。

（3）临时性。每一个项目都具备明确的开始时间和结束时间，项目在有限的时间内完成，不会无限延续下去，是一个临时性的任务。

（4）独特性。每一个项目都是唯一的、一次性的。

（5）资源约束性。每一个项目都需要运用各种资源来开展工作，同时资源是有限的。

（6）不确定性。每一个项目在开展的过程中都会存在一定的风险，以及很多不确定的因素让项目发生一些变化。

项目的管理的研究者们把 20 世纪 80 年代之前称为传统项目管理阶段，20 世纪 80 年代之后称为现代项目管理阶段，在该阶段，随着项目管理不断地应用到各行各业的工程和高科技项目中，持续地促进着项目管理理论和方法的优化和发展<sup>[31]</sup>。

项目管理就是在项目中运用知识、技能、工具与技术，在规定的时间内、成本和质量目标范围内完成项目的各项工作<sup>[30]</sup>。项目管理具有以下特征<sup>[32]</sup>：

（1）项目管理具有全过程、全方位的系统的思维方法论的特征。

（2）项目管理具有多目标及多约束的特征。

（3）项目管理的实施团队组织具有一次性的特征。

- (4) 项目管理具有可预见性与变化性的特征。
- (5) 项目管理具有项目周期性和内容独特性的特征。
- (6) 项目管理具有方法和技术随项目的不断实施而不断创新特征。

### 2.1.2 项目管理知识体系

PMBOK 将项目管理知识体系划分为 5 大过程组和 10 大知识领域。表 2.1 是 5 大过程组，表 2.2 是 10 大知识领域。

表 2.1 PMBOK 的 5 大过程组

过程组名称	过程组描述
启动过程组	一个新项目的开始，授权开始项目
规划过程组	制定各个知识领域的管理计划
执行过程组	按照管理计划执行工作
监控过程组	对项目管理过程进行控制
收尾过程组	一个项目的收尾，产品和文档的交付，结束采购

表 2.2 PMBOK 的 10 大知识领域

知识领域名称	描述
项目整合管理	根据项目情况确定各个管理阶段的管理工作和过程
项目范围管理	确定成功完成项目，需要交付的项目工作内容
项目进度管理	明确项目各阶段的时间安排和目标成果，并按时完成
项目成本管理	确保完成项目的实际成本不超过预算成本
项目质量管理	确保达到客户规定的产品质量要求
项目资源管理	确保最大的发挥和利用项目有限的资源
项目沟通管理	确保将合适的项目信息在合适的时间传达给合适的人
项目风险管理	应对项目实施过程中可能遇到的各种不确定的因素
项目采购管理	从项目组织外部获取资源或服务的一系列活动
项目相关方管理	为确保项目的顺利进行，对项目相关利益者进行管理

各个过程组之间是通过输出进行关联的，一个过程组的输出是下一个过程组的输入。表 2.3 为 PMBOK 的 5 大过程组、10 大知识领域、49 个模块之间的关系。

项目管理知识体系把一些零散的、独立的知识点加以整合，使之形成了具有关联的完整的知识体系，项目管理知识体系作为项目管理的重要内容，也是项目的核心<sup>[33]</sup>。

表 2.3 PMBOK 的 5 大过程组、10 大知识领域、49 个模块之间的关系

知识领域	项目管理过程组				
	启动过程组	规划过程组	执行过程组	监控过程组	收尾过程组
项目整合管理	制定项目章程	制定项目管理计划	指导与管理项目工作 管理项目知识	监控项目工作 实施整体变更控制	结束项目或阶段
项目范围管理		规划范围管理 收集需求 定义范围 创建 WBS		确认范围 控制范围	
项目进度管理		规划进度管理 定义活动 排列活动顺序 估算活动持续时间 制定进度计划		控制进度	
项目成本管理		规划成本管理 估算成本 制定预算		控制成本	
项目质量管理		规划质量管理	管理质量	控制质量	
项目资源管理		规划资源管理 估算活动资源	获取资源 建设团队 管理团队	控制资源	
项目沟通管理		规划沟通管理	管理沟通	监督沟通	
项目风险管理		规划风险管理 识别风险 实施定性风险分析 实施定量风险分析 规划风险应对	实施风险应对	监督风险	
项目采购管理		规划采购管理	实施采购	控制采购	
项目相关方管理	识别相关方	规划相关方参与	管理相关方参与	监督相关方参与	

## 2.2 软件项目管理概述

### 2.2.1 软件项目

软件项目是为了给用户开发一套计算机软件而进行的临时性工作。软件项目具有以下显著特点<sup>[34]</sup>:

- (1) 可见性差。软件产品与其他任何产业的产品不同,是一种无形的逻辑产品,非物理实体。
- (2) 定制化。没有明显的制造过程,也不存在重复生产,基本上都是定制的。
- (3) 高复杂性。软件本身是相当复杂的,涉及因素众多。
- (4) 成本高。软件项目投入大、成本高、脑力劳动强。
- (5) 需求不确定性大。用户在项目初期无法清楚描述需求,并且随着项目推进,不断变更需求。
- (6) 高风险。软件项目过程中不确定因素较多,使得其具有极高的风险。

### 2.2.2 软件项目管理

软件项目管理是美国在 20 世纪 70 年代中期提出来的,美国国防部分析了软件项目无法按时完成、费用超出预算和质量不达标的主要原因,研究发现 70% 的软件项目失败是由于管理因素导致的,而非技术原因造成的<sup>[35]</sup>。于是,越来越多的研究学者们开始重视软件项目中的各种管理问题。

软件项目管理是项目管理在软件项目中的应用,是为了使软件项目在规定的时问、成本等资源约束,提交高质量软件项目产品,而进行的管理活动<sup>[36]</sup>。软件项目管理可以保证软件项目在整个生命周期都能够在监控下开发,并且在保证产品质量、满足成本预算、达到预期目标以及在计划的进度前提下,确保交付的软件产品是满足客户要求的<sup>[37]</sup>。

软件项目管理相对其他项目管理具有以下特殊性<sup>[38]</sup>:

- (1) 项目目标不明确,开发进度难以控制,质量难以保证,成本难以控制,生产效率难以度量。
- (2) 项目执行中需求变化频繁。



- (3) 项目开发周期长，复杂度高，不可控的因素很多。
- (4) 项目对研发人员的技能水平要求高。
- (5) 项目要满足不同利益相关者的需求，协调难度大。

所以，进行软件项目管理是非常必要的。软件项目管理的根本目的是以最合理的、最有效的、最经济的方法来保证软件项目的成功完成。

软件项目管理包括人的管理、软件配置的管理和软件开发，其中的软件开发有很多的软件开发模型，软件开发模型的不同，软件项目管理也会出现较大的差异。软件开发模型也称软件过程模型或软件生命周期模型或软件生存期模型，它是软件开发过程中各项任务按照一定的流程来进行的，合适的软件开发模型可以提高研发效率，提升产品质量。常见的软件开发模型包括：

(1) 瀑布开发：瀑布开发是传统软件开发模型，又称线性模型，是早期广泛应用的开发模型，瀑布开发的各个阶段是自上而下顺序执行的，一个阶段的输入是上一个阶段的输出，如同瀑布流水一样，逐层下落<sup>[39]</sup>。如图 2.1 所示为瀑布开发流程图。

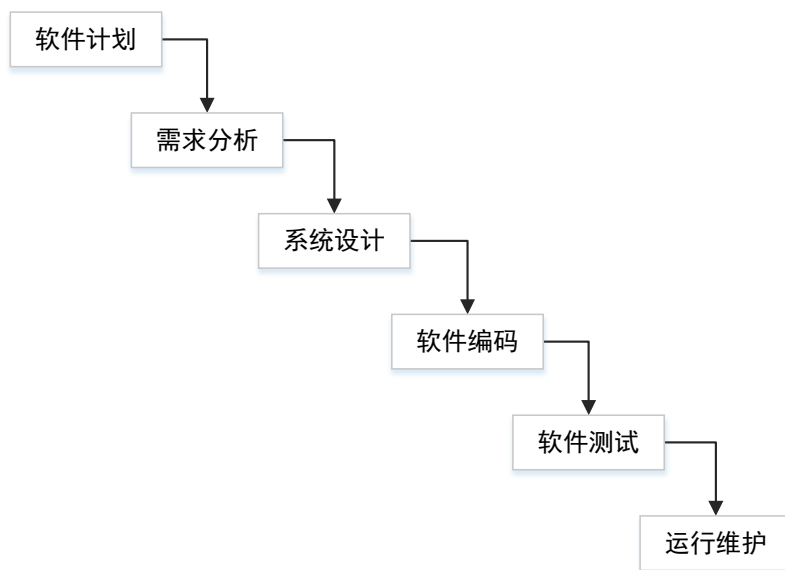


图 2.1 瀑布开发

(2) 增量开发：增量开发是将软件产品分为一系列增量产品，首先增量交付的是基本功能产品，经过用户反馈形成下一个增量开发计划，不断重复这个过程，最终完成产品<sup>[40]</sup>。如图 2.2 所示为增量开发流程图。

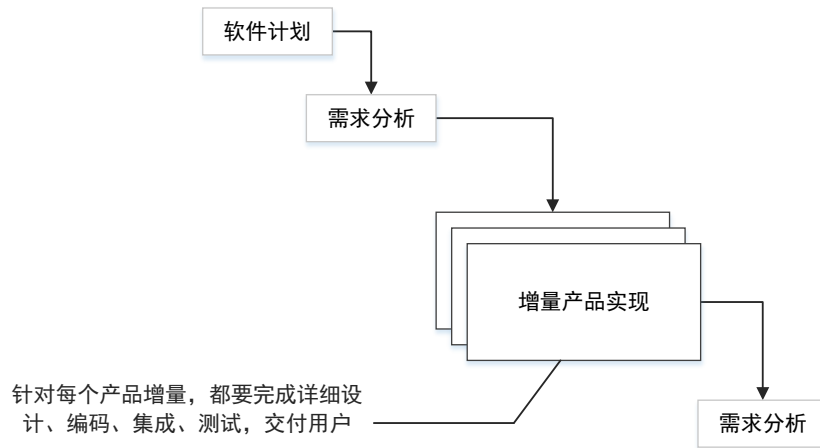


图 2.2 增量开发

(3) 快速原型开发：快速原型开发是首先快速建一个能够原型的原型系统，让用户试用并得到反馈意见，研发人员根据意见快速修改原型系统，并重复这一过程，直到用户确认原型系统满足要求为止，然后根据这个原型系统进行需求分析后再开展设计和编码工作<sup>[41]</sup>。如图 2.3 所示为快速原型开发流程图。

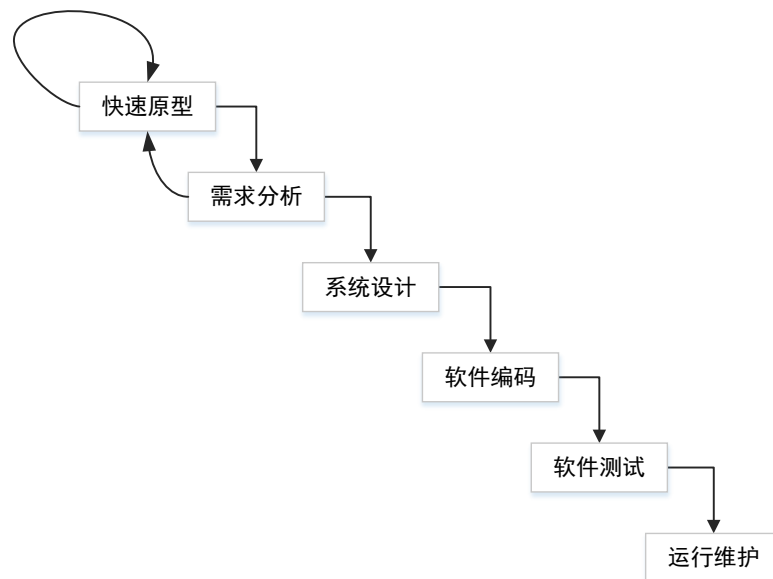


图 2.3 快速原型开发

(4) 螺旋开发：螺旋开发引入了风险分析，特别适合大型复杂系统，螺旋开发分为若干阶段，其每个螺旋都对应一个阶段，每个阶段都包括制定计划、风险分析、实施工程和客户评估这四个过程，不断重复这四个过程，直至开发出完整且用户满意的产品<sup>[42]</sup>。如图 2.4 所示为螺旋开发流程图。

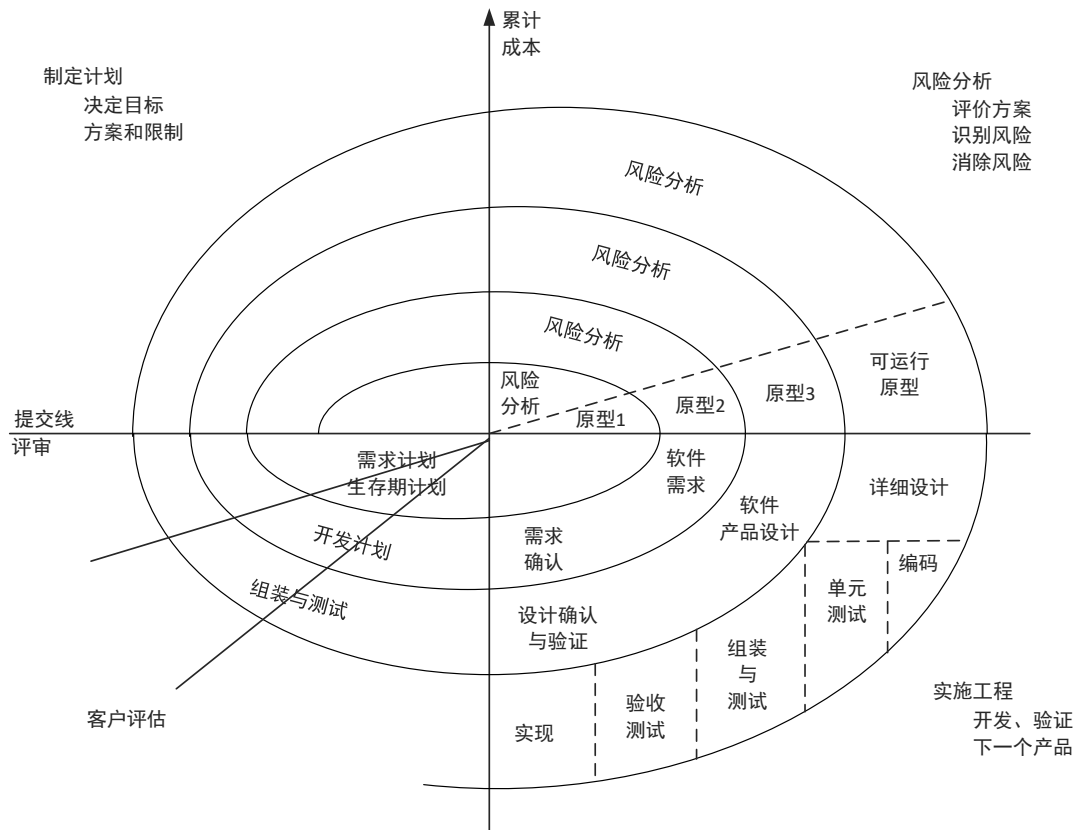


图 2.4 螺旋开发

（5）敏捷开发：敏捷开发是以人为核心、迭代循序渐进的方法，它的核心思想在于快速、增量的交付用户可工作的软件，在敏捷开发中，软件产品分为多个子产品，每个子产品的开发都包括计划、需求分析、设计、编码和测试，然后向用户交付可运行的产品<sup>[43]</sup>。图 2.5 为敏捷开发流程图。

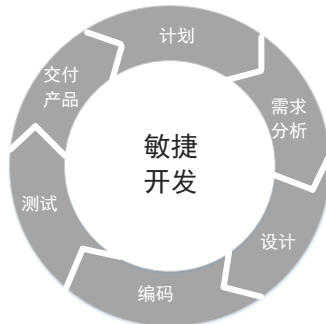


图 2.5 敏捷开发

在软件项目中选择哪种软件开发模型,需要根据企业的实际情况而定。表 2.4 对以上五种模型进行了简单对比,选择合适的软件开发模型对于软件项目的成败至关重要。

表 2.4 软件开发模型比较

软件开发模型	优点	缺点	适用场景
瀑布开发	阶段清晰、活动明确 过程控制性强 文档全	不灵活,需求变更困难 文档多,成本高 用户在项目后期才能看到系统	需求明确
增量开发	适应变化,降低开发风险 更快的开发出可用系统 减少用户需求变更	各产品增量集成困难 软件过程控制失去整体性	需求不明确
快速原型开发	原型反复修改,需求确认循序渐进 用户直接参与	需求管理要求高 不适合大型项目 快速开发可能导致产品质量低下	核心需求明确
螺旋开发	任何阶段都会考虑风险	依赖风险评估专家	适合大型、复杂、高风险系统
敏捷开发	产品快速并经常交付 得到用户及时反馈 适应变化需求	不适合小型开发项目 必要的设计和文档可能缺失	需求不明确 需求频繁变更

## 2.3 敏捷开发概述

敏捷开发是一种思想,它是一套软件开发的价值和原则,提倡迭代开发、尽早交付、持续改进并对变化做出快速的反应。敏捷开发是一种面对变化需求快速开发的能力,它的核心思想注重沟通、合作、可用产品和适应变化。敏捷开发是能够迅速适应变化而不是提前预测变化,在实际项目中也无法预测用户的需求变化,更不可能提前为需求变化做准备。为了实现敏捷开发这一思想,研究人员提出了很多敏捷开发方法,如 Scrum、极限编程等。

### 2.3.1 敏捷开发的起源

随着高新信息技术不断发展,产品生命周期日益缩短,用户的需求多样化和个性化,高度不确定的项目需求变化快、复杂性和风险都很高,软件项目面临着越来越多的问题,这些问题包括用户早期需求不明确、用户中后期频繁变更需求、需求优先级划分不合理、交付产品功能用户使用率低、用户参与度低、无法快速响应市场变化、进度滞后、成本超支、产品质量不符合用户要求、无效沟通或沟通效率低、团队成员积极性不够等<sup>[44]</sup>。

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》,敏捷宣言定义了敏捷开发的 4 条价值观和 12 条原则,指出敏捷开发就是用高效的方法交付有价值的产品,敏捷是人的本性,找到适合自己的、适合团队的、适应客户的就是最好的敏捷<sup>[45]</sup>。

敏捷的诞生是对实践经验的总结,是来源于经验主义的。敏捷宣言是将一些经验整理总结为共同的价值观和原则,给项目起到指导作用的。越来越多的企业利用敏捷开发更快地响应市场和需求的变化,更快地将产品推向市场,并对现有的项目管理流程进行改善,通过敏捷开发对企业组织进行创新和自我重塑<sup>[46]</sup>。

### 2.3.2 敏捷开发简介

敏捷开发是一种涵盖了各种框架和方法的总和,它是指符合敏捷宣言的价值观和原则的任何方法、技术、框架和实践。如表 2.5 给出了敏捷开发的 4 个价值观,我们可以看出,敏捷开发的第一个价值观是以人为本,第二个价值观是目标导向,第三个价值观是客户为先,第四个价值观是拥抱变化<sup>[9]</sup>。

表 2.5 敏捷开发 4 个价值观

敏捷开发的 4 个价值观
人员沟通和协作重于流程和工具
有用的产品重于全面的文档
与客户合作重于与客户谈判
响应变化重于按计划行事

如表 2.6 给出了敏捷开发的 12 条实践原则,敏捷开发的这些原则中提倡尽早持续交付有价值的软件,适应需求的变化,更注重人的作用,强调面对面沟通,

通过迭代交付可工作的软件，指引客户参与到项目中来，然后随着业务的不断深入逐步调整<sup>[19]</sup>。

表 2.6 敏捷开发 12 条原则

敏捷开发的 12 条原则
1.尽早地、持续地交付客户想要的产品，从而满足客户的要求。
2.在项目的任何阶段都要拥抱变化。
3.尽可能短的周期交付产品。
4.项目中业务人员和开发团队一起办公，提升沟通效率。
5.人是核心，激励、支持和相信他们，帮助他们达成目标。
6.面对面沟通是最有效的沟通方式。
7.及时交付可工作的软件。
8.项目相关干系人处于一个稳定的、可持续的节奏中。
9.追求卓越技术和不断完善设计会提升团队能力。
10.做到简洁，交付刚刚好的产品。
11.打造一个成熟的自组织团队，该团队可以通过高效协作，出色完成工作。
12.团队通过反馈不断地进行反思和回顾，并进行调整和改进。

敏捷开发是指以人为核心的，根据用户不断变更的需求采用迭代、循序渐进的方法，是一种轻量级的方法<sup>[47]</sup>。敏捷开发是价值驱动的，而传统的项目管理是计划和成本驱动的，敏捷开发实现快速多次迭代交付，可通过早期用户反馈，降低需求变更带来的风险，灵活性和适应性都非常强<sup>[48]</sup>。

### 2.3.3 敏捷开发优势

敏捷开发主要体现在及时响应客户不断变化的需求，快速交付产品并获得早期反馈，尽快交付满足客户要求的产品。敏捷开发具有很多的优势<sup>[49]</sup>：

（1）交付用户所需的产品。采用迭代方式一步一步在原有用户认可的功能基础上添加更多功能。

（2）降低项目不确定性。尽早的提供用户可用的产品，得到了用户的频繁反馈，减少项目风险。

（3）更快的将产品推向市场。敏捷开发周期短，能更快占领市场，抢占先机，增强企业竞争力。

（4）资源利用率最大化。敏捷团队一直都只关注当前最需求的、最具有价值的部分产品功能，人力和物力资源都用在刀刃上。

（5）高灵活性和强适应性。敏捷开发具有较强的灵活性和适应性，能快速

适应变化。

(6) 产品质量得到保证。敏捷开发对每一轮的迭代都有严格的质量要求,使得产品得到了可靠的质量保证。

(7) 高效的自组织团队。敏捷团队是自组织团队,可激发团队成员的主动性和积极性。

### 2.3.4 敏捷开发常用方法

#### (1) Scrum

Scrum 是一个轻量级的框架,它通过提供对复杂问题的灵活性和自适应解决措施来帮助人们、团队和组织创造实用价值。数字化转型公司(digital.ai)在发布的《2021 年敏捷年度状态报告》中指出,Scrum 仍然是敏捷开发中运用最广泛的方法,Scrum 及 Scrum 混合其他方法的占比超过了 75%。

Scrum 框架有 3 大支柱、3 个角色、3 个工件、5 个活动和 5 个价值,简称“33355”,3 大支柱包括透明、检查和调整;3 个角色包括产品负责人、敏捷教练和开发团队;3 个工件包括产品待办列表、冲刺列表、产品增量;5 个活动包括产品待办列表梳理会议、冲刺计划会议、每日站会、冲刺评审会议和冲刺回顾会议;5 个价值包括承诺、专注、开放、尊重和勇气<sup>[50]</sup>。表 2.7 展示了 Scrum 框架内容。

#### (2) 精益开发

精益思想源于日本丰田公司的精益生产,精益思想包括精益开发原则和一些实践工具,其核心思想是查明和消除浪费,在软件开发过程中出现的 bug、交付用户不使用的功能、团队成员之间互相等待都是浪费,浪费及其源头都是必须分析和查明的,再设法将它们消除<sup>[51]</sup>。表 2.8 为精益开发 7 个原则。

和其他敏捷方法相比,精益开发是基于最小化浪费和最大化价值的原则,不断优化开发流程,从而提高研发效率和降低成本的方法,因此,将精益开发与其他敏捷开发方法一起使用将会取得不错的效果<sup>[52]</sup>。

#### (3) 极限编程

极限编程是敏捷开发中应用最为广泛和最有效果的方法之一,更强调适应性,在项目开始后,需求变更会带来项目成本的急剧增加,而极限编程的主要目的就是降低因需求变更而带来的成本增加的方法,在项目的每个阶段适应变化,

更加有效的方法<sup>[53]</sup>。如表 2.9 展示了极限编程的 4 个价值、5 个原则和 12 个核心实践内容。

表 2.7 Scrum 框架的“33355”

框架	框架内容	框架内容说明
3 大支柱	透明	团队成员清楚知道要交付的产品功能和开发进展情况。
	检查	定期检查产品功能和开发过程是否发生偏差。
	调整	发现问题立马进行调整和改进。
3 个角色	产品负责人	负责规划要开发什么,以什么顺序开发,并确定每项需求的验收标准。
	敏捷教练	负责指导团队使用 Scrum 并提出改进,帮助团队正确使用 Scrum。
	开发团队	负责完成产品负责人规划的需求,交付每次冲刺结束时可发布的产品增量。
3 个工件	产品待办列表	是一个根据用户价值进行优先级排序的需求列表。
	冲刺列表	冲刺需要完成的需求列表。
	产品增量	可交付给用户的产品。
5 个活动	产品待办列表梳理会议	将收集的所有产品需求,按照用户价值进行优先级排序
	冲刺计划会议	在每个冲刺之初,从产品代办列表中挑选最高优先级的需求进行讨论、分析和估算得到本次冲刺的任务列表的会议。
	每日站会	每日 15 分钟站立会,每个团队成员回答昨日做了什么,今日准备做什么及遇到的问题。
5 个价值	冲刺评审会议	团队成员演示产品功能,产品负责人进行需求验收。
	冲刺回顾会议	在冲刺结束后确定这次冲刺有哪些地方可以改善。
	承诺	团队成员承诺实现团队的目标。
	专注	团队成员专注于团队目标。
	开放	团队成员都能看到项目中的工作内容和进展情况。
	尊重	团队成员相互尊重,理解他人的独特性。
	勇气	团队成员有勇气做出承诺并迎接挑战。

表 2.8 精益开发 7 个原则

原则	原则说明
消除浪费	任何对产品不增加价值的工作都是浪费。
增强学习	软件开发的过程是不断发现、不断学习、不断改进的过程。
尽可能晚的决定	在获取足够的信息之前,不草率地做决定。
尽快交付	缩短交付周期,快速得到用户反馈,更快投入市场。
给团队自主权	授权给团队,可以让团队产生更强的动力和创造力。



续表 2.8 精益开发 7 个原则

原则	原则说明
内置完整性	用户与团队成员之间沟通顺畅，保证用户对产品的完整性体验。
优化整体	整体优化优于局部优化。

表 2.9 极限编程 4 个价值、5 个原则、12 个核心实践

框架	框架内容	框架内容说明
4 个价值	沟通	通过沟通确认产品的功能需求。
	简单	只针对当前的需求，进行简单的设计。
	反馈	重视用户、团队及相关利益者的反馈，反馈越快越好。
	勇气	勇敢的使用极限编程的基本原则去做事情。
5 个原则	快速反馈	反馈要做到及时和迅速，这样就能发挥重要作用。
	假设简单	不考虑未来的变化和代码的可重用性，用简单的方法来解决任何问题。
	增量变化	一次一个小变化的新版本，通过这种增量叠加的方式，使系统对于用户来说是可控的。
	拥抱变化	不确定的因素总是存在的，我们不应对这种变化产生抗拒心理，而应拥抱它们。
12 个核心实践	高质量的工作	每个人都期望出色的高质量的完成工作。
	短交付周期	短期内增量交付，用户反馈及时处理。
	计划游戏	快速制定计划，随着需求的变化而改进。
	结对编程	结对编程是指代码由两个人共同完成，一个人在电脑前编写代码，另外一个人对代码进行评审。
	可持续的节奏	团队需要长期持久地努力工作，才有获胜的希望。
	代码集体所有	任何人都可以修改系统中的代码，每个人都对所有的代码负责。
	编码规范	团队所有成员遵守一个统一的编码规范，就像一个人写的代码。
	简单设计	用最简单的办法实现每个小需求，只处理当前的需求，保持设计的简单。
	测试驱动开发	先写测试代码，再写程序。
	重构	对每个需求进行简单设计，同时不断地对设计进行改进。
	系统隐喻	用形象生动的比喻描述系统是如何工作的。
	持续集成	频繁的集成可以帮助项目尽早的发现问题。
	现场客户	在整个项目开发周期中，需要现场客户确定需求，与开发团队有频繁的交流和讨论，并进行验收测试。

极限编程就是这样通过引入一些价值、原则、核心实践等来达到降低需求变更成本的目的，使用极限编程可以灵活的应对项目过程中的需求变更。

#### （4）DevOps

DevOps 是敏捷开发的延续，它将敏捷的精神延伸到运维阶段，DevOps 是开发运维一体化平台，用于软件开发、运维和测试三个部门之间沟通协作的过程、方法与系统的统称<sup>[54]</sup>。DevOps 以改善产品交付而闻名，通过三大支柱，具备一站式持续集成、持续交付、自动化测试、持续部署等功能简化了开发过程中集成测试、兼容性测试和部署过程，大大提高了研发效率。

DevOps 的三大支柱包括人、流程和平台，其中人和流程形成了文化，人和平台为团队赋能，平台和流程构建了工具。表 2.10 为 DevOps 的关键实践。

表 2.10 DevOps 关键实践

关键实践	关键实践说明
持续集成	持续集成经常地将代码集成到一起，每次集成后都进行自动化测试，从而尽快地发现集成问题。
持续交付	持续交付是指在持续集成的基础上，通过手动方式部署生产环境。
持续部署	持续部署是比持续交付更高的一个阶段，通过自动化测试的变动都可以自动部署到生产环境中。

DevOps 不仅解决了开发和运维的壁垒，而且是整个项目过程端到端的协作。DevOps 将项目交付中重复使用率高度的工作利用自动化工具，使用流水线形式，快速交付用户产品功能，用这种一小步一小步的方式不断改进软件质量，快速交付用户产品功能<sup>[55]</sup>。

#### （5）方法比较

上述四种方法，尽可能有效、快速地交付用户产品，并且是迭代增量的方式来完成用户的需求，从而避免用户需求频繁变更而带来的项目损失。表 2.11 对敏捷开发的四种方法进行了比较。

敏捷开发被认为是有效的解决软件项目遇到的挑战和方法，企业需要根据自身企业的实际情况和面临的问题来考虑是否引入敏捷开发。没有一个过程或方法是十全十美的，生搬硬套效果往往不好，更重要的是找到适合企业自己的方法。

敏捷开发就是通过一个一个迭代过程来应对需求和技术的快速变化。敏捷团队每一次迭代结束交付给用户的都是一个可部署、可用的系统，这样用户可以尽早的使用系统并反馈意见。因此，敏捷开发就是这样利用短的迭代周期，来处理频繁变更的需求和用户的持续反馈。

表 2.11 敏捷开发方法比较

方法	核心价值	适用场景
Scrum	Scrum 是用于开发、交付和持续支持复杂产品的一个框架，是一个增量的、迭代的开发过程	一个新的完整的开发项目 开发过程规范化 用户有时间及时反馈增量产品意见
精益开发	寻找项目管理流程中的浪费，设法消除这些浪费，从而降低成本	用户熟悉自己产品 用户不注重过程文档，只看结果
极限编程	通过各种核心实践来提高开发效率，目的在于降低因需求变更而带来的成本	用户熟悉自己的产品 用户全程参与项目
DevOps	DevOps 是敏捷开发的延伸，将开发与运维打通，实现持续部署	开发团队有足够的人力去弄自动化

## 第3章 基于敏捷开发的T公司软件项目管理现状与存在的问题

### 3.1 T公司简介

#### 3.1.1 T公司基本情况

T公司长期为电网及能源行业客户提供软件产品及服务，典型产品与解决方案已在全网各省公司及其下属单位得到了大量推广应用。

#### 3.1.2 T公司组织架构

T公司设有6个中心，其中研发中心负责产品规划、项目管理、人员培养，产品的设计、研发、测试、部署等工作。如图3.1为T公司的组织结构。

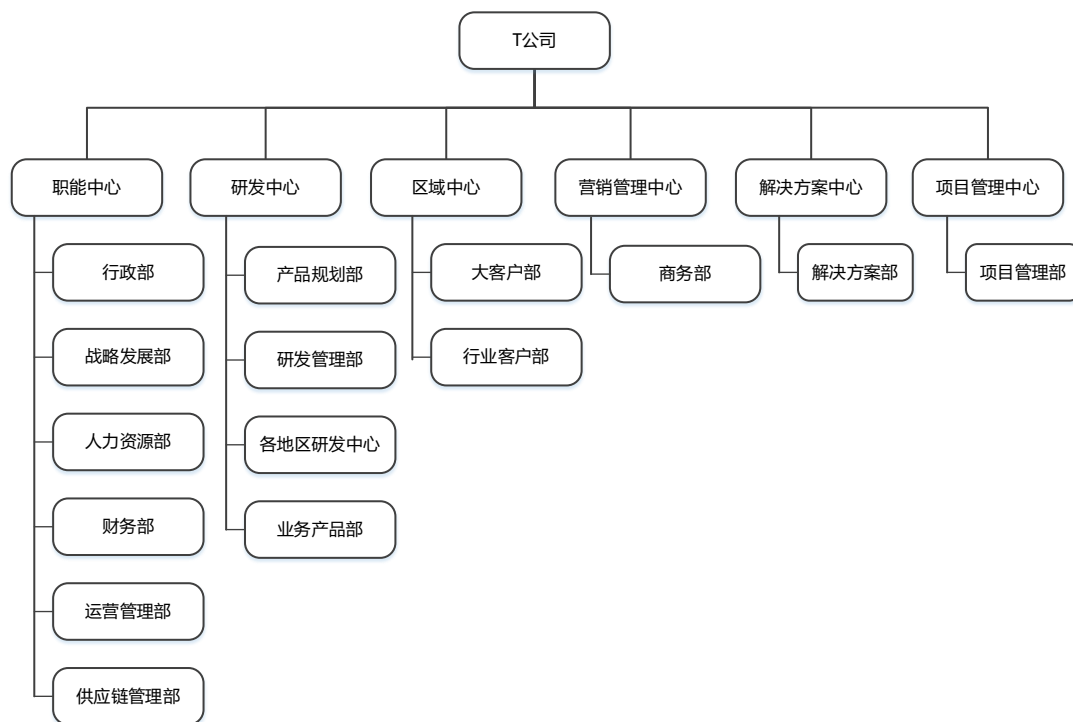


图 3.1 T 公司的组织结构

## 3.2 基于敏捷开发的T公司软件项目管理现状

### 3.2.1 基于敏捷开发的软件项目管理流程

T公司软件项目是基于敏捷开发的管理流程，由三个各自循环的阶段组成，这三个循环阶段为需求分析循环阶段、研发冲刺循环阶段和测试版本循环阶段，如图3.2所示。

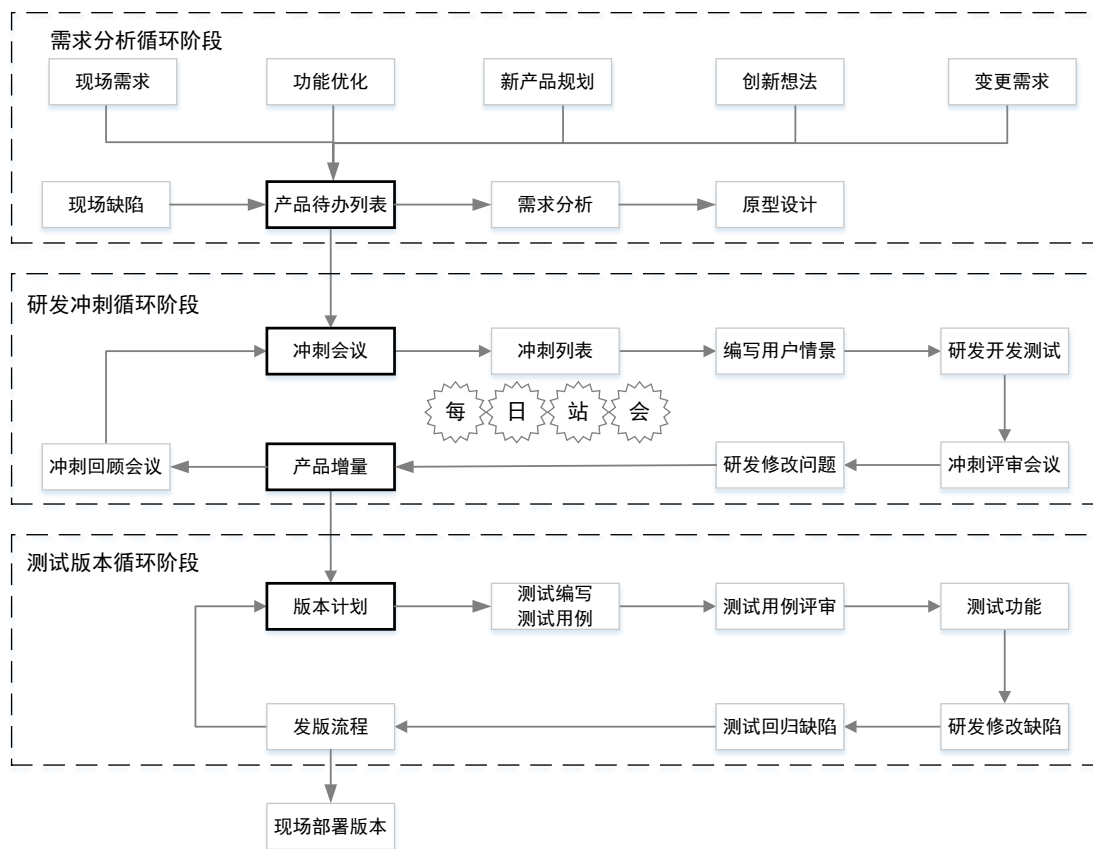


图 3.2 T 公司软件项目管理流程

#### (1) 需求分析循环阶段

需求分析循环阶段主要是由产品经理和需求人员完成。产品经理会在该阶段持续的将现场提的需求、现有产品优化需求、新产品规划需求、变更需求以及创新想法需求放到产品待办列表中进行统一记录，并排好优先级。需求人员对产品待办列表中优先级高的需求进行分析和设计，形成原型设计，然后与提需求的相关干系人进行评审，再根据评审意见修改，修改后的原型与文档继续评审，经

过这样反复评审和修改后，最终得到一个大家都认可的原型设计。需求人员在该阶段不断的将一个个的需求拆解成原型，拆解完一个，再拆解下一个。

#### （2）研发冲刺循环阶段

研发冲刺循环阶段主要由研发人员完成。产品经理在该阶段首先会召开一个冲刺会议，敏捷团队成员一起从产品待办列表中选择有价值需求，通过大家一起讨论，分析和估算得到该冲刺阶段需要完成的需求列表，即冲刺列表。一个冲刺时间为2周，就这样一个新的冲刺开始进行，需求人员宣讲冲刺列表中需求的原型和设计，需求人员编写用户情景，研发人员开始进行开发工作，在开发过程中遇到的需求问题找需求人员答复，遇到的开发难题找架构师解决。在冲刺期间，敏捷团队每日早上9点都要进行站立会，研发人员每天回答三个问题，昨天干了什么？今天准备干什么？遇到了什么问题？产品经理对整个冲刺进度进行监控并及时解决团队遇到的问题。研发人员在完成需求功能开发测试后，产品经理会对完成的功能进行验收，然后研发人员修复验收过程中发现的问题，最后得到了这次冲刺的产品增量。冲刺结束后，产品经理会组织一次冲刺回顾会议，参与冲刺的每个人都需要写三个这次冲刺中做得好的地方及三个需要改进的地方，然后全员进行投票，找出最需要改进的一条，放到产品待办列表中。这样一轮2周的冲刺就结束了，产品经理又会组织冲刺会议来进行新一轮的冲刺，研发冲刺阶段就是这样一个由研发将冲刺列表中的需求开发成产品增量的循环过程。

#### （3）测试版本循环阶段

测试版本循环阶段主要由测试人员完成。冲刺列表中的每个用户情景都配置了对应的版本号，现场需要产品版本时，产品经理会建一个版本计划，这个版本计划中包含需要发版的用户情景，测试人员会编写该版本内用户情景的测试用例，有疑问或不清楚的地方，会与需求人员进行确认。测试用例编写完成后，测试人员会组织需求对应的研发人员、需求人员、测试经理进行用例评审，测试人员根据评审意见，修改测试用例。版本计划里的所有用户情景都关闭后（即功能开发完成），测试人员开始对本次版本计划里的所有用户情景进行测试，并提问题单，研发人员对这些问题进行修复，测试人员回归问题，本次版本测试完成。最后，产品经理新建一个发版流程，经过相关责任人评审后，将版本发往现场部署使用。测试人员就是这样一直不停地根据一个又一个的版本计划进行测试版本的循环工作。

## 3.2.2 基于 Azure DevOps 工具管理软件项目

T 公司是基于 Azure DevOps 工具来管理软件项目的。Azure DevOps 是由微软开发的一个 WEB 版服务平台，它提供了很多种工具，可使用这些工具更好地进行团队协作。Azure DevOps 提供了 5 个主要模块，如表 3.1 所示。

表 3.1 Azure DevOps 的 5 个主要模块

模块名称	模块说明
Azure Boards (电子看板)	提供了一组敏捷开发的工具，来支持工作规划和跟踪、代码缺陷以及问题报告的工作。
Azure Repos (代码仓库)	提供 Git 代码存储库，用于代码的控制管理。
Azure Pipelines (流水线)	提供了构建和发布服务，以支持应用程序的持续集成和持续交付。
Azure Test Plans (测试计划)	提供一些工具来测试应用程序，包括手动测试和持续测试。
Azure Artifacts (包管理)	允许团队共享来自公有和私有源的包，并将包集成到 CI/CD 管道中。

如图 3.3 所示为 T 公司使用 Azure DevOps 工具进行敏捷开发管理流程，该工具主要用于研发冲刺循环阶段和测试版本循环阶段。

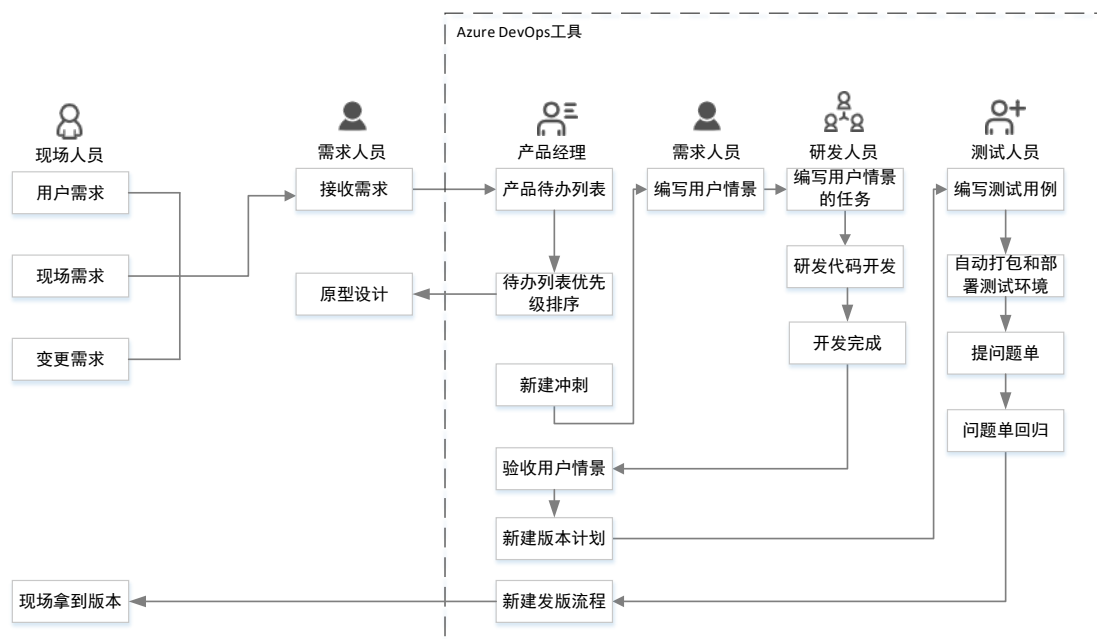


图 3.3 T 公司使用 Azure DevOps 工具进行敏捷开发管理流程

T公司使用 Azure DevOps 工具进行敏捷开发管理流程如下:

(1) 产品经理将需求和问题录入到产品待办列表

现场人员与客户对接需求后,会将用户需求、现场需求和变更需求通过微信、微信群和电话的方式与需求人员对接,现场发现的问题基本是发在微信群里。产品经理会将这些需求和问题录到 Azure DevOps 工具的产品待办列表中,并对其排优先级。

(2) 产品经理新建冲刺

在每次冲刺开始的时候,产品经理都会在 Azure DevOps 工具中新建一个周期为两周的冲刺,这样一个新的冲刺就开始了。

(3) 需求人员录入用户情景

需求人员在产品经理新建的冲刺内,将冲刺列表中的需求在 Azure DevOps 工具中录入成一个个的用户情景,用户情景是敏捷开发中的一个概念,它就是使用简明扼要的语言描述用户需求。同时,需求人员会把冲刺前设计好的原型截图到用户情景里,用于指导研发人员开发和测试人员测试。

(4) 研发人员录入用户情景的工作任务

研发人员会对新建冲刺内的所有用户情景拆解成多个工作任务,再将这些工作任务录入到 Azure DevOps 工具的任务面板中,任务面板是一个可视化的管理工具,任务面板第一列是用户情景,接下来几列依次是新建、暂停、活动、已解决、已关闭,在研发的过程中,研发人员将任务的完成情况通过鼠标在页面上拖动到相应的列上,团队成员就能一目了然的了解工作任务的进展情况。在一个用户情景下所有的工作任务是已关闭的状态,就表示该用户情景研发人员已经完成了开发测试工作,然后产品经理对该用户情景进行验收,验收完成后,由产品经理关闭该用户情景。如图 3.4 为任务面板示意图。

(5) 产品经理新建版本计划

产品经理会根据现场的要求在 Azure DevOps 工具中新建一个版本计划,这个版本计划中包含现场需要的用户情景、版本开始时间、版本计划提测时间、版本计划发布时间等,以便测试人员安排时间进行测试相关工作。

(6) 测试人员编写用户情景测试用例

测试人员拿到版本计划后,会根据版本计划上的用户情景在 Azure DevOps 工具中编写测试用例,完成后,需要安排相关人员对测试用例进行评审,测试人员根据评审意见修改测试用例。





图 3.4 任务面板示意图

#### （7）工具自动打包和部署

当一个版本计划中所有的用户情景都完成后，产品经理会根据版本计划的版本号在 Azure DevOps 工具里建一个 release 分支，然后该工具会获取该 release 分支上的代码并通过该工具自动打包成安装包，再将安装包自动部署到测试环境并启动起来，这样测试人员就可以开始测试了。

#### （8）测试人员测试，并录入问题

测试环境自动部署成功后，测试人员根据测试用例对用户情景进行测试，测试过程中发现的问题录入到 Azure DevOps 工具，研发人员及时修复测试人员提的问题，等研发人员修复全部问题后，测试人员会再通过 Azure DevOps 工具将修改后的代码自动打包和自动部署到测试环境中，然后对问题进行回归测试。

#### （9）产品经理新建发版流程

测试人员在经过几轮测试后，没有再发现新的问题后，测试人员会给出一个测试报告，以此来表示这个版本计划测试完成了。产品经理会在 Azure DevOps 工具中新建一个发版流程，发版流程经过各级负责人审核后，现场人员就可以获取版本相关部署包和升级文档在现场进行部署了。

### 3.3 基于敏捷开发的T公司软件项目管理存在的问题

#### 3.3.1 企业相关员工访谈

为了全面地、深入地了解T公司应用敏捷开发进行软件项目管理可能存在的问题,通过访谈的方式,选择了一些项目相关干系人作为访谈对象进行沟通。这些访谈对象包括项目经理、产品经理、需求人员、前端研发人员、全栈研发人员、测试人员和现场实施人员,通过项目管理者 and 项目执行者两种不同的角色了解项目存在的问题。表3.2所示为具体的访谈对象和数量。

表3.2 访谈对象清单

序号	访谈对象	访谈数量
1	项目经理	1
2	产品经理	2
3	需求人员	1
4	前端研发人员	2
5	全栈研发人员	8
6	测试人员	3
7	现场实施人员	2
合计		19

本次通过对19位企业相关员工的访谈记录,并对收集的问题进行汇总归类,获取T公司使用敏捷开发的软件项目管理的问题清单,如表3.3所示。

表3.3 访谈收集软件项目管理问题清单

问题类别	问题描述
需求管理问题	现场人员提需求比较随意,提的需求并不是自己想要的功能 现场人员刚入职不久,对业务了解不深,随着项目的进行,频繁的修改需求 需求人员业务知识不足,理解现场需求慢 各个地区需求过多,谁催的急就给谁做,导致其他地区投诉 有些地区与客户沟通没有规划,今天提需求明天要 地区定制的功能太多,增加了项目成本
质量管理问题	用户情景测试点描述不到位,导致测试发现不了问题 项目进度紧,研发人员自测不到位,测试人员发现问题较多 发版的部署包或多或少的都会在现场发现问题 代码实现没有文档和注释,过一段时间修复问题,要重新看代码

续表 3.3 访谈收集软件项目管理问题清单

问题类别	问题描述
沟通管理问题	由于项目迭代快、任务多，团队内成员沟通不畅 项目沟通规划不足 团队成员反馈意见渠道匮乏
进度管理问题	研发人员存在等待需求的情况，导致进度延期 研发人员被打扰过多，定位现场紧急问题或需求，影响开发进度 并行任务多，导致进度控制不到位
组织问题	跨部门间协调困难 版本计划提交过晚，测试部无法及时安排人员测试 研发交付的版本，现场人员不及时验收

我们已经通过访谈法找到了T公司实施敏捷开发过程中遇到的很多问题，但是我们并不知道这些问题对于项目影响的大小程度，因此我们需要对访谈问题清单中的这些问题做定量分析，从而识别出比较严重的问题，以便后续针对这些问题进行优化。

### 3.3.2 基于层次分析法定量分析

#### （1）层次分析方法介绍

美国著名运筹学家 T. L. Saaty 在 20 世纪 70 年代提出了层次分析法(AHP)，该方法是指将需要决策的复杂问题分解成多层次，通过两两之间的比较来确定层次中各因素的相对重要性，并在此基础上进行定性和定量分析的多层次决策方法<sup>[56]</sup>。T 公司使用敏捷开发的软件项目管理中问题严重程度属于多层次、多因素的复杂决策问题，因此可以用该方法来找到问题严重程度。为了使因素决策判断定量化，形成判断矩阵，引用 T. L. Saaty 1~9 标度法判断，如表 3.4 所示。

表 3.4 T. L. Saaty 1~9 标度法

标度值	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

#### （2）建立层次结构模型

为找出当前软件项目中需要重点解决的问题，将访谈得到的所有问题划分为不同的层次和因素，第一层为分析的目标，第二层是问题分类，第三层为第二层问题分类的具体因素，如图3.5所示。



图 3.5 T 公司软件项目管理问题层次结构模型

#### （4）构建判断矩阵

对访谈中所有的问题建立层次结构模型后,根据 T. L. Saaty 1~9 标度规则,通过团队成员投票的方式,对同层因素两两之间比较重要程度,得到判断矩阵,见表 3.5~表 3.10。

表 3.5 第二层矩阵

	A	B	C	D	E
A	1	2	2	3	5
B	1/2	1	2	2	4
C	1/2	1/2	1	2	3
D	1/3	1/2	1/2	1	2
E	1/5	1/4	1/3	1/2	1

表 3.6 第三层需求管理问题 A 判断矩阵

	A1	A2	A3	A4	A5	A6
A1	1	1/4	1/3	1/2	1/4	1/7
A2	4	1	3/2	2	1	2/5
A3	3	2/3	1	1	1/3	1/3
A4	2	1/2	1	1	1/3	1/4
A5	4	1	3	3	1	1/2
A6	7	2 1/2	3	4	2	1

表 3.7 第三层质量管理问题 B 判断矩阵

	B1	B2	B3	B4
B1	1	1/3	1/4	1/2
B2	3	1	2	2
B3	4	1/2	1	2
B4	2	1/2	1/2	1

表 3.8 第三层沟通管理问题 C 判断矩阵

	C1	C2	C3
C1	1	2	5
C2	1/2	1	3
C3	1/5	1/3	1

表 3.9 第三层进度管理问题 D 判断矩阵

	D1	D2	D3
D1	1	1/2	1
D2	2	1	3
D3	1	1/3	1

表 3.10 组织管理问题 E 判断矩阵

	D1	D2	D3
D1	1	2	3
D2	1/2	1	3
D3	1/3	1/3	1

## (4) 层次分析法计算过程

按行元素求积，再开  $n$  次方根

$$\overline{W}_i = \sqrt[n]{\prod_{j=1}^n a_{ij}} \quad i, j = 1, 2, \dots, n \quad (3.1)$$

通过归一化，得到指标权重和特性向量

$$W_i = \frac{\overline{W}_i}{\sum_{i=1}^n \overline{W}_i}, W = \{W_1 \ W_2 \ \dots \ W_n\}^T \quad (3.2)$$

计算最大特征根

$$\lambda_{max} = \frac{1}{n} \sum_{i=1}^n \frac{(AW)_i}{W_i} \quad (3.3)$$

一致性校验

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (3.4)$$

其中， $n$  为矩阵的维度。

一致性比率

$$CR = \frac{CI}{RI} \quad (3.5)$$

其中，随机一致性指标  $RI$  和判断矩阵的阶数有关， $RI$  与矩阵阶数对应关系如表 3.11。

表 3.11 随机一致性指标  $RI$  取值表

矩阵阶数	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

一般如  $CR < 0.1$  时，则认为该判断矩阵通过一致性检验。

## (5) 计算第二层和第三层权重和排序

本论文使用 Excel 来计算各层因素判断矩阵的，第二层判断矩阵 Excel 计算公式见附录 A，第三层各层因素判断矩阵和第二层计算公式一样，在附录中就不重复了。第二层和第三层计算权重结果见表 3.12~17，从计算结果表格中我们可以看出， $CR$  小于 0.1，表示这些判断矩阵都满足一致性校验。

表 3.12 第二层判断矩阵权重计算结果

	A	B	C	D	E
权重	0.381113958	0.25470801	0.182239643	0.117434135	0.064504254
$\lambda_{max}$	5.066918663				
CI	0.016729666				
RI	1.12				
CR	0.014937202				

表 3.13 第三层需求管理问题 A 判断矩阵权重计算结果

	A1	A2	A3	A4	A5	A6
权重	0.045468091	0.174773322	0.104728813	0.08893473	0.21784471	0.368250334
$\lambda_{max}$	6.079076725					
CI	0.015815345					
RI	1.24					
CR	0.012754311					

表 3.14 第三层质量管理问题 B 判断矩阵权重计算结果

	B1	B2	B3	B4
权重	0.098903035	0.407434444	0.30958323	0.18407929
$\lambda_{max}$	4.096278719			
CI	0.032092906			
RI	0.9			
CR	0.035658785			

表 3.15 第三层沟通管理问题 C 判断矩阵权重计算结果

	C1	C2	C3
权重	0.581552067	0.308995644	0.10945229
$\lambda_{max}$	3.003694598		
CI	0.001847299		
RI	0.58		
CR	0.003184998		

表 3.16 第三层进度管理问题 D 判断矩阵权重计算结果

	D1	D2	D3
权重	0.24021087	0.549945607	0.209843523
$\lambda_{max}$	3.018294707		
CI	0.009147354		
RI	0.58		

续表 3.16 第三层进度管理问题 D 判断矩阵权重计算结果

CR	0.015771299
----	-------------

表 3.17 第三层组织管理问题 E 判断矩阵权重计算结果

	E1	E2	E3
权重	0.527836133	0.332515928	0.139647939
$\lambda_{max}$	3.053621576		
CI	0.026810788		
RI	0.58		
CR	0.046225496		

### 3.3.3 各层因素权重及总排序

根据层次分析法对T公司问题进行定量分析后,我们得到了项目需求管理、项目质量管理、项目沟通管理、项目进度管理和组织管理对项目整体问题的综合影响程度,也可以得到第三层各因素对第二层问题分类的影响总排序,如表3.18所示。

表 3.18 权重总排序表

第一层		第二层		第三层				
因素	CR	因素	权重	CR	因素	权重	综合权重	总排序
问题	0.014937202 <0.1	A	0.381113958	0.012754311<0.1	A1	0.045468091	0.017328524	18
					A2	0.174773322	0.066608553	6
					A3	0.104728813	0.039913612	10
					A4	0.08893473	0.033894267	12
					A5	0.21784471	0.08302366	4
					A6	0.368250334	0.140345342	1
		B	0.25470801	0.035658785<0.1	B1	0.098903035	0.025191395	14
					B2	0.407434444	0.103776816	3
					B3	0.30958323	0.078853328	5
					B4	0.18407929	0.04688647	9
		C	0.182239643	0.003184998<0.1	C1	0.581552067	0.105981841	2
					C2	0.308995644	0.056311256	8
					C3	0.10945229	0.019946546	17
		D	0.117434135	0.015771299<0.1	D1	0.24021087	0.028208956	13
					D2	0.549945607	0.064582387	7
					D3	0.209843523	0.024642793	15



续表 3.18 权重总排序表

第一层		第二层		第三层			
因素	CR	因素	权重	CR	因素	权重	综合权重
		E	0.064504254	0.046225496<0.1	E1	0.527836133	0.034047676
					E2	0.332515928	0.021448692
					E3	0.139647939	0.009007886
							11
							16
							19

从表 3.18 的权重总排序中, 5 个问题分类的相对重要性排序: 项目需求管理问题>项目质量管理问题>项目沟通管理问题>项目进度管理问题>组织管理问题, 我们可以看出, 需求管理问题的权重达到了 0.38, 需求管理的确在软件项目管理中起着举足轻重的作用, 并且在管理过程中, 需求的问题也比较多, 提了 6 个问题。除了需求外, 产品的质量和团队成员间的沟通依旧是大家比较重视的, 毕竟无论过程怎么样, 交付给客户的产品才是最重要的。在第三层整体权重排序中, 定制化需求太多、团队成员之间沟通不畅、项目进度紧质量不好是排在问题前三的, 说明 T 公司在使用敏捷开发后, 已经不再惧怕需求的变更了, 组织问题对项目的影影响最小。

因此, 通过对访谈收集过来的所有问题进行层次分析后, 我们找出了后面要重点解决的问题, 由于大家对需求相关方面提出了很多问题, 我们把需求管理和质量管理的相关问题作为主要问题, 还有第三层排序前三中的团队成员间沟通不畅问题也作为主要问题。本论文会针对 T 公司软件项目管理中的需求管理问题、质量管理问题、沟通不畅三个问题进行原因分析, 并寻找合适的优化方案来解决这三个主要问题。

### 3.4 基于敏捷开发的 T 公司软件项目管理问题原因分析

#### 3.4.1 基于原因鱼骨图对主要问题的原因分析

鱼骨图又名因果图或石川图, 是一种找出问题根本原因的分析方法, 鱼骨图分为问题鱼骨图、原因鱼骨图和对策型鱼骨图三种<sup>[57]</sup>。本论文使用的是原因鱼骨图对需求管理问题、质量管理问题和成员之间沟通不畅这三个问题进行原因分析, 通过对问题原因的层层分解, 发现问题的根本原因。

##### (1) 需求管理问题原因分析

需求管理问题, 是访谈时大家反馈比较多的问题, 可见在需求这一块, 尽管

T公司是使用的敏捷开发进行软件项目管理的,但还是遇到了很多问题。如图3.6所示,针对需求管理问题的鱼骨图。

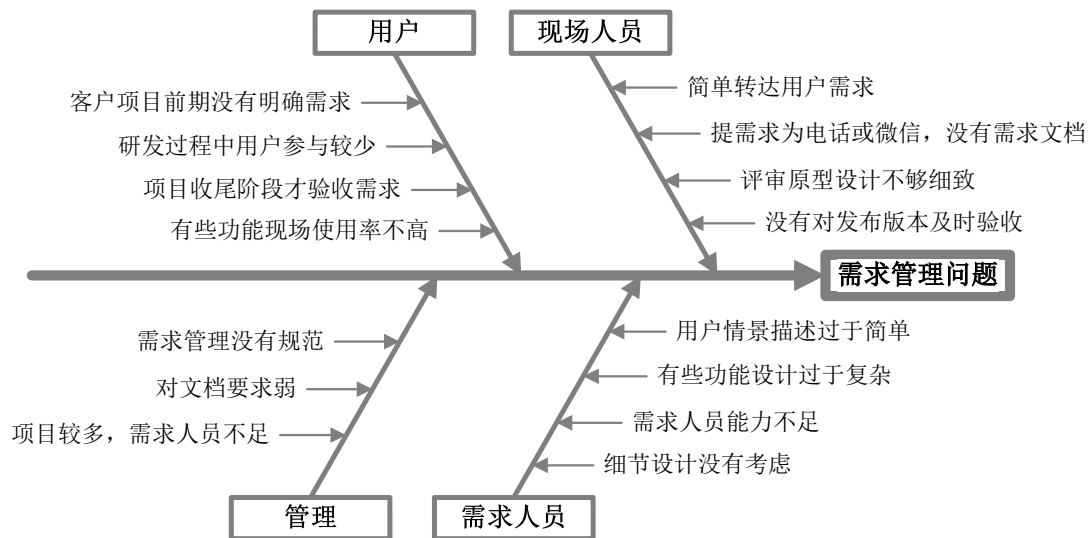


图 3.6 需求管理问题的鱼骨图

从用户、现场人员、需求人员和管理四个方面分析了需求管理问题的原因。

用户方面,用户在项目前期没有很明确的需求,只能提出一些大概的想法,而在项目研发的过程中,用户参与度又不高。尽管在研发过程中发布了很多个产品版本,但用户直到项目收尾阶段才来验收项目需求,又重新提了很多需求和问题,导致项目后期进度紧张,并且有些完成的功能用户使用率低。

现场人员方面,现场人员是近距离接触客户的,获取用户需求后,有时没有认真思考,也没有与用户确定,而是直接通过电话或微信转达给研发团队。在评审原型设计时,现场人员评审有时会出现不仔细、遗漏的情况,同时在版本到现场后,没有及时进行功能验收,导致研发做出的功能不是现场人员想要的。

需求人员方面,需求人员的输出只有原型设计,很多细节上的设计没有考虑全面,用户情景的描述也比较简单,需求人员通过原型与研发人员宣讲后,研发人员就开始开发工作。研发人员会根据自己的理解去补充细节设计,有时做出来的功能与需求存在偏差。还有些需求人员能力不足,做出来的设计非常复杂,研发人员实现起来费时费力,现场人员也表示不好用。

管理方面,现场人员如何提需求、需求人员如何分析需求、产品经理如何跟踪需求、是否需要写文档、文档写到何种程度等,没有一个统一的规范。另外,团队项目比较多,而需求人员较少,也是一个问题。

## (2) 质量管理问题分析

产品质量问题多主要表现在两方面，一是测试人员测试出的问题多，二是版本发给现场后，依旧发现一些问题。如图3.7所示，针对质量管理问题的鱼骨图。

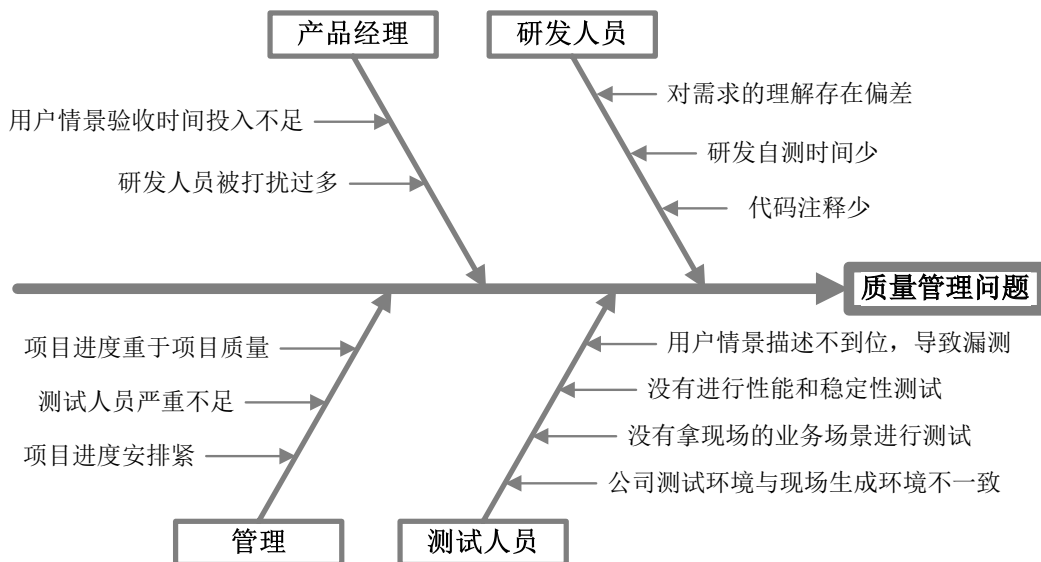


图 3.7 质量管理问题的鱼骨图

从产品经理、研发人员、测试人员和管理四方面分析了质量管理问题的原因。

产品经理方面，由于产品经理需要与很多人对接沟通，处理的事情也比较多，于是很多产品经理投入用户情景的验收工作时间严重不足，导致测试人员发现较多的问题。在研发过程中，研发人员被过多的打扰不仅会影响项目进度，也会影响产品质量。

研发人员方面，很多研发人员认为代码实现功能了，就是完成需求了，在自测上花的时间比较少，这可能是产品质量不好的主要原因，有时研发人员对需求的理解与测试人员不一致，实现与产品原型有差异，测试人员也会提问题单。还有研发人员写代码时的注释比较少，导致后期代码维护困难，修改的过程中还极易引入新的问题。

测试人员方面，用户情景的描述简单、不清晰，是测试人员比较苦恼的，容易引发测试遗漏，进而将问题带到现场。测试人员测试的时候，测试环境没有与现场保持一致、没有使用现场的业务场景进行测试、没有测试性能和稳定性测试等，是导致版本到现场有问题的原因。

管理方面，在实际的项目管理中，大家对项目进度的关注多于产品质量，项

目进度紧，是产品质量的天敌，如何平衡好进度与质量之间的关系是项目管理的一大挑战。另外，测试人员严重不足，对产品质量有很大的影响，而现场的频繁要求发版，会让测试人员更加力不从心。

### （3）沟通不畅原因分析

项目成员间工作协作好的前提是有效沟通，沟通不畅会引发很多项目上的问题。如图3.8所示，针对沟通不畅的鱼骨图。

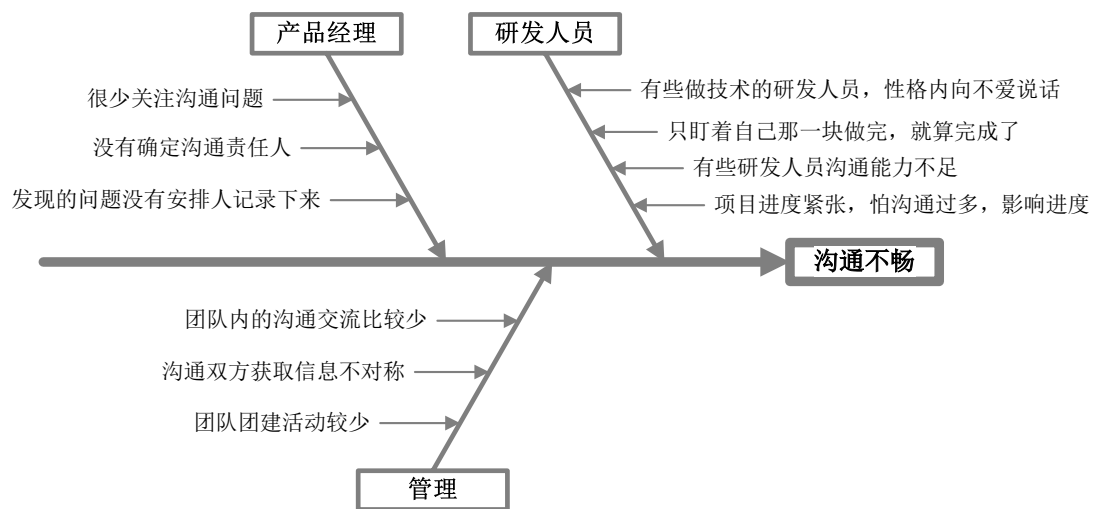


图 3.8 沟通不畅的鱼骨图

从产品经理、研发人员和管理三个方面分析了沟通不畅的原因。

产品经理方面，在沟通时，产品经理没有确定沟通责任人，并让其与研发人员对接，导致实现可能并不是按照沟通结果进行开发的。还有产品经理没有安排人记录发现的问题及问题沟通结论，可能会导致一个问题的反复沟通，浪费时间。

研发人员方面，有些研发人员，本身性格不爱说话，不喜欢与别人沟通，也有些研发人员是沟通能力不好，导致与他人无效沟通。因此，每次出现问题的时候，你就会听到研发人员解释，“我以为...”，这其实就是沟通上存在问题。还有些研发人员，只盯着自己那块功能，不考虑其他人，或者由于项目进度紧张，不与人沟通，节省时间，这样也是会有问题的。

管理方面，有些产品经理，本身也是技术人员出生，对沟通这一块就不重视，所以团队成员之间沟通较少也不觉得是个问题，也很少组织团建活动去增进团队成员之间的感情。另外，在进行沟通前，要确保沟通双方信息一致，否则这样的沟通是毫无意义的。

### 3.4.2 问题原因总结

本论文为了有针对性地去设计优化方案，将问题原因总结为以下几方面：

#### （1）需求传递不到位

项目前期用户并没有明确的需求，只会提出一些自己对产品的大概理解，然后现场人员把用户的需求再结合一些自己的想法，通过电话或微信直接提需求，需求人员再根据自己的经验做产品原型，如果用户还不参与产品原型的评审，这样最终实现的功能往往会与用户真正的需求偏差较大。需求人员通过产品原型对研发人员和测试人员进行功能宣讲，宣讲完成后，研发人员和测试人员根据这个产品原型然后再加上自己的理解来进行开发和测试的，这样做出来的功能能否满足用户需求是很难保证的。

如图3.9所示为T公司需求的传递关系。我们可以看出从用户提需求到研发实现要经过好几轮的传递，只要有一个传递环节出了问题或者偏差，研发出来的产品就可能与用户想要的功能相差很大。

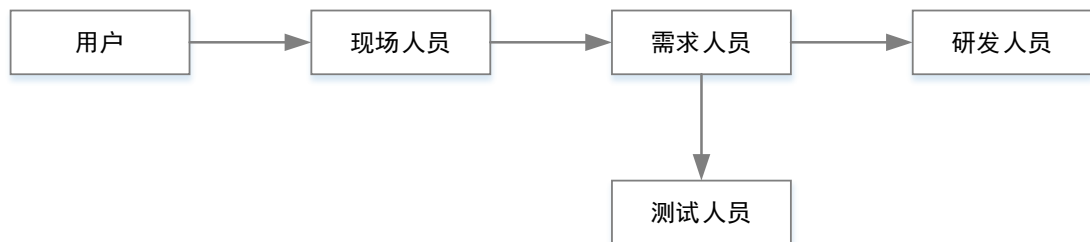


图 3.9 T 公司需求的传递关系

#### （2）交付的产品质量不高

项目进度紧张，研发人员的自测时间少了，然后产品经理又没有投入足够的时间进行用户情景验收测试，这个产品质量是得不到保证的。而测试人员的测试环境与现场环境不一致，没有对系统功能进行性能测试和稳定性测试，也没有拿现场的业务场景进行测试，这样发布的版本在现场怎么可能是没有问题的呢。产品质量不满足客户要求，客户就不会为这次产品买单，在失掉客户的同时，也失去了企业竞争力。

#### （3）团队成员之间沟通不畅

产品经理对沟通管理这一块不重视，很少组织团建活动，团队成员之间沟通也少，也没有建立意见反馈渠道。由于有些技术人员性格内向不爱说话，也有一

些是本身沟通能力不足，再加上沟通双方获取的信息不对称，团队成员之间的沟通不畅问题很快就显现出来了。还有就是有些工作任务进行讨论沟通得出结论后，并没有将结论反馈给做该任务的研发人员，导致研发人员做出的功能没有用，重新返工开发。团队工作中发现的问题也没有记录下来，下次又发现该问题，又重新定位讨论，讨论时突然有人发现之前讨论过这个问题，这就纯粹是浪费团队时间了。有效的沟通是解决一切问题的关键，没有沟通就没有团队成员之间的协作。

## 第4章 基于敏捷开发的 T 公司软件项目管理优化方案

### 4.1 项目需求管理优化方案

#### 4.1.1 需求管理流程优化

##### （1）需求录入原始需求

T 公司一直都要求现场人员在 Azure DevOps 工具上录入自己的需求或者用户的需求，我们把这种需求叫做原始需求，这种原始需求的录入有以下的好处：第一，便于原始需求的跟踪，防止研发的过程中出现需求遗漏的情况。第二，整理成文字描述能让现场人员花时间思考这个需求，而非只是起到一个中间转达的作用，直接将用户的需求转达给需求人员。第三，留证据，可以避免后期现场人员和需求人员起争执，如出现需求人员说这个需求是现场提的，而现场人员表示很无辜，说我怎么会提这种没有用的需求。第四，文字描述可以使用户、现场人员与需求人员之间更好的沟通，更好地去理解需求。

在 T 公司有些项目组已经做到了要求现场人员必须在工具上录入原始需求，但还是有些项目组并没有做到这一点，后续要求这些没有做到的项目组也要要求现场人员将用户需求作为原始需求录入工具。现场人员发现的问题也要录入 Azure DevOps 工具中，需求人员千万不要给现场人员省事，而给自己带来不必要的麻烦。因此，通过 Azure DevOps 工具与现场人员对接需求，即可以帮助需求人员和现场人员减少需求理解上的偏差，也方便后续双方对需求的跟踪。

##### （2）坚持需求分析基本原则

需求人员对于用户提出的需求，通常情况下，立马考虑如何实现，并没有考虑到用户提出的需求可能并不是真正的需求，而是经过自己思考理解后的问题解决方案，而该解决方案是否能解决实际问题就不得而知了。这样即使我们实现了用户需求，依然没有解决用户遇到的问题，很多研究也发现有很多开发的产品功能很少或者从来不被使用。

还有我们的用户往往是无法准确表达自己的需求的，第一，用户觉得习以为常的需求，他们在表述时可能就会忽略了，所以用户的需求表达有时候会出现失真；第二，用户表达需求时，由于自身认知和表达能力有限，表达不出或者表达

不准确自己内心的真正想要的需求，询问用户时，用户也不知道需求是什么；第三，用户在特定的场景下为了迎合需求人员的问题，说出违心的需求，可能会误导需求人员，这样需求人员获取的需求就会产生偏差，所以不能完全听从用户，需求人员应该通过多问问为什么做来发现用户真正的需求<sup>[58]</sup>。所以需求人员在对接需求的时候，要坚持一个需求基本原则即“不是看怎么做，而是看为什么做”<sup>[24]</sup>。

如图 4.1 所示为需求分析基本原则，用户提出需求以后，需求人员需要与用户沟通为什么会提这样的需求，是为了解决什么样的问题，坚持需求分析基本原则，从而发现用户的痛点，真正的解决用户的问题。



图 4.1 需求分析基本原则

### （3）优化需求优先级排序

每个项目的资源都是有限的，所以很少有软件项目能够将用户提的所有需求一次性的都交付给用户，因此每个项目都需要对用户需求进行优先级排序。需求优先级排序可以解决有限资源与大量需求之间的矛盾，对于 2 周一次迭代的敏捷开发项目，优先做最重要的或者最有价值的需求，这样有助于在各种资源约束情况下，项目组能尽快交付价值最高的产品。

把需求的重要性和紧急性统称为商业价值原则，基于这个商业价值，我们将优先级分为重要紧急、重要不紧急、紧急不重要、不紧急不重要四种情况。KANO 模型定义三个层次的用户需求：基本型需求、期望型需求和兴奋型需求，基本型需求是去掉这些需求后，这个产品就不能使用，期望型需求是使用户满意度增加，兴奋型需求是使用户产生惊喜<sup>[58]</sup>。我们一般认为基本型需求是重要紧急、期望型需求是重要不紧急、兴奋型需求是紧急不重要，按照先做重要紧急的需求，后做重要不紧急的需求，接着做紧急不重要的需求，最后做不紧急不重要的需求。再结合需求估算工作量（天）、前置条件这两个影响因素，来将需求安排到敏捷开发冲刺中。在 T 公司使用的 Azure DevOps 工具，会针对每一个需求都会生成



一个唯一的需求编号,需求工作量估算是敏捷团队全体成员按照斐波那契数列 1, 2, 3, 5, 8, 13, ...来估算的,前置条件指是需要完成某个其他需求,才能开始这个需求。如表 4.1 为敏捷开发冲刺待办列表中的排序及冲刺分配。

表 4.1 敏捷冲刺代办列表

需求简 要描述	需求类型	重要紧急 程度	估 算 工 作 量 (天)	前置 条件	需求 优先级	迭代 计划	完成 情况
需求 1	现场	重要紧急	3		1	迭代 1	进行中
需求 2	现场	紧急不重要	3	需求 5	9	迭代 3	未开始
需求 3	现场	重要不紧急	5		2	迭代 1	进行中
需求 4	现场	重要紧急	5		3	迭代 1	进行中
需求 5	现场	重要不紧急	8	需求 4	6	迭代 2	未开始
需求 6	优化	重要紧急	3		4	迭代 1	进行中
需求 7	现场	重要不紧急	5		7	迭代 2	未开始
需求 8	优化	不紧急不重要	1		10	迭代 3	未开始
需求 9	优化	重要紧急	8		5	迭代 1	进行中
需求 10	优化	重要不紧急	3		8	迭代 2	未开始

需求排优先级是一个动态的、持续进行的过程,这个优先级是根据项目过程中不断调整的,有些需求在不同的时期的优先级可能会产生变化,但是尽量不修改当前迭代的需求。需求排优先级时一定要避免“音量优先级”,哪个嗓门越大,获得的优先级就越高。

#### (4) 优化用户情景任务面板

每个用户情景任务拆解的时候,除了拆解研发任务,还要拆解研发自测、代码评审和验收用户情景三个任务,只有这些任务全部完成时,这个用户情景才能算完成。

#### (5) 需求人员跟踪用户验收

让用户参与进来,是敏捷开发的一个核心理念。在测试版本循环阶段发布版本给现场后,需求人员就要跟踪现场人员和用户使用该版本的情况,是否有功能需要改善或者提新的需求,只有得到这些反馈后,才算一次迭代完成,否则用户到项目收尾阶段才来验收需求,那和瀑布式开发方式就没有什么区别了,依旧是客户不满意功能,同时提了一堆需求变更和新的需求,这很可能导致项目返工,验收困难的,没有做到真正的敏捷开发了。因此,应用敏捷开发来进行软件项目管理的时候,用户及时验收是很重要的,这样可以减少用户需求频繁变更对软件

项目带来的冲击。

#### 4.1.2 提升需求人员能力

需求人员的职责是获取、分析、记录和验证用户需求，通俗一点理解就是将用户的需求传递给研发去实现功能，因此需求人员需要具备一定的分析能力、业务知识和沟通能力才能圆满地完成这个工作。高效率的需求人员是多面手，不仅要有超强的沟通、引导和人际交往的能力，还需具备工作所需的技术和业务知识，个人魅力，耐心并且真诚地希望与他人合作。所以需求人员应该具备如下三种能力：

（1）分析能力：具备从一个用户的需求中总结出一套可以满足多个用户的需求的能力，还要从错综复杂信息中找到解决问题的方法，更要有区分真正需求与解决方案的差异，找到用户真正想要的需求。

（2）业务知识能力：拥有较强业务知识的需求人员能够极大的减少与客户之间沟通不畅的情况。

（3）沟通能力。需求人员在与人沟通时，态度要随和，表达要清晰一致。

每一个需求人员都要根据自身的情况提升工作中所需的这三个能力，主动弥补自己的不足，这样才能高效的完成需求分析工作。

#### 4.1.3 建立需求跟踪流程

需要建立一个表格用来跟踪需求的进展和状态。需求跟踪管理表具体描述如表4.2所示。

表4.2 需求跟踪管理表的具体描述

需求跟踪管理表内容	需求跟踪管理表内容含义
版本	需求所属版本号。
需求描述	需求简要概括。
需求提出人	需求提出人。
需求类型	包括新增、优化、缺陷。
需求状态	待讨论、暂缓、拒绝、需求分析、UI设计、数据库设计、研发、测试、版本发布、用户验收、已关闭。
备注	其他信息，如暂缓或拒绝的理由

## 4.2 项目质量管理优化方案

### 4.2.1 质量管理流程优化

#### （1）测试环境保持与现场生产环境一致

测试人员需要保证 Azure DevOps 工具上的通过自动打包和自动部署的测试环境与现场生产环境一致，否则可能会出现现场环境有问题，而在公司测试却没有问题的情况。现场人员使用研发团队发布的版本出现各种问题时，他们的第一反应是这个版本是没有经过测试的或者测试人员能力不行，这严重影响了整个研发团队在现场人员中的形象，导致现场人员对于研发团队的极大不信任。总结现场出现过的问题如下：

①现场 window 环境是虚拟机没有显卡，而测试人员用的是有显卡的电脑测试，导致很多带有酷炫效果的 web 页面在用户现场需要加载很长时间。

②现场是集群环境，而测试环境是单节点环境，而某些问题只有在集群下才会出现。

③现场数据库是主备双机，而测试环境是单节点数据库，而某些现象只有在主备双机情况下才会出现。

因此，在测试人员测试前，需要将 Azure DevOps 工具连接的测试环境部署成和现场生产环境一模一样，这样可以给大家省去很多的麻烦，产品经理也能减少因为此类问题而被现场投诉的烦恼。

#### （2）研发人员投入充足时间自测

对每一个研发人员来说，质量问题都应该高度重视，这些问题不仅仅会影响到产品的使用，同时发现并修复软件中的问题通常要花费很多的时间和金钱，所以质量问题通常是导致项目成本超支和时间进度延期的主要原因。

在敏捷开发快速迭代下，我们明显可以感觉到研发的时间被压缩了，一个用户情景拆成了好几个任务，一个任务的完成时间不允许超过两天，有些任务是半天或者一天就要完成。而在实际的研发过程中，参加一个会议、讨论一个问题或者中途被打断，都会影响到这个任务的完成，而大部分研发人员这个时候的做法还比较一致，那就是压缩自测的时间。

有一些研发人员非常的乐观，容易忽略细节，特别是看到代码运行正常，就会认为功能完成了，直到测试人员提了很多问题单，才一脸吃惊的样子，这些问

题自己怎么没有测试出来呢。研发人员没有对功能进行充分的测试,这是导致产品质量不高的一个主要原因。因此,研发人员在功能开发完成后,还是要投入充足的时间进行测试,从而提升产品质量。

### (3) 产品经理投入充足验收测试时间

研发完成的产品功能不可避免的会与最初定义的产品出现偏差,可能是不同的人理解上的不一致,也可能是研发根据自己的理解修改了需求,因此产品经理这个验收测试就显得尤为重要了。

但是在实际工作中,由于产品经理的工作比较多,也比较杂,于是很多产品经理就会选择压缩或者省掉验收测试的时间,以节省时间去干他们认为更重要的工作。但是这样做可能会出现很多问题,如测试人员开始测试时测试不下去导致版本被打回,版本问题数量过高等,最终产品进度延期,导致用户不满意。产品经理验收测试的目的在于保证产品质量,达到设计预期。产品经理的验收测试跟测试人员测试不同,产品经理会把自己当成用户,测试产品功能是否与需求一致,流程是否能走通。

产品经理是整个项目组中关键人物,对整个项目的需求和开发进度是了如指掌的,因此,在研发人员完成产品功能开发后,产品经理在测试环境上对用户情景进行验收测试是非常重要的,而且还必须投入充足的时间进行验收测试,这个时间是不能想压缩就压缩的、想省掉就省掉的,否则很可能出现“出来混,总是要还的”,后面又花大量的时间来填坑。

### (4) 架构师代码评审

代码评审,是一种通过走读代码,验证代码是否满足需求,同时指出其中不满足规范的“低质量”代码,从而提升整个团队的代码质量的过程。

这条改进方案是在访谈的时候研发人员提到的,他们说写完代码后,并不知道自己的代码是写的好还是不好,希望可以由架构师组织代码评审会议,团队所有研发人员都参加,这样研发人员就知道哪些代码写的好,哪些代码是需要改进的。

因此,代码评审不仅帮助研发人员提升代码编写能力,而且可以提升代码的可维护性和可读性,从而提高代码质量。

### (5) 制定项目规范,研发过程进行规范化管理

项目规范管理可以让新员工快速融入敏捷团队,提升团队研发效率、团队协作效率和降低沟通成本,提高项目的可维护性,同时规范管理也是团队知识的积

累。因此,规范管理在项目管理中有着非常重要的作用,如表4.3为项目规范的作用。

表4.3 项目规范作用

项目规范的名称	项目规范的作用
设计规范	确保不同研发人员做出相同样式的页面,保持页面样式的统一。
开发规范	帮助研发人员养成良好的编程习惯,写出清晰易懂的代码,减少软件编码过程中不必要的错误,提高软件质量与研发效率。
实施部署规范	确保任何人都能顺利的在用户现场部署应用系统,降低部署成本,提高工作效率。
问题规范	用于研发人员快速的记录问题及相应的解决方案,同时这些问题的处理思路可以用于指导团队其他人员处理类似的问题。
运维规范	确保现场系统稳定安全运行,进一步提高系统的维护质量和效率。

项目规范在团队实施的过程中,还会根据具体的实施情况进行优化,如表4.4为项目规范的内容。

表4.4 项目规范内容

规范名称	规范内容
设计规范	色彩规范、字体规范、边框规范、按钮规范、单选框规范、多选框规范、输入框规范、下拉选择框规范、级联选择框规范、开关规范、日期选择规范、表格规范、tab 标签规范、提示框规范、弹出框规范、翻页规范、步骤条规范、穿梭框规范、树形控件规范
开发规范	代码目录规范、代码命名规范、代码注释规范、配置规范、异常处理规范、日志规范、打包规范
实施部署规范	数据库部署规范、系统部署规范、中间件部署规范,系统升级部署规范
问题规范	每个问题都包括问题描述、问题分析和解决思路
运维规范	操作规范、终端使用规范、系统账号使用规范、服务器管理规范、数据定期备份和检查规范、现场应急处理规范

#### 4.2.2 增加测试内容

##### (1) 测试现场实际应用场景

现场实际业务比我们想的要复杂得多,而我们在公司测试的时候,只是拿一些简单的业务来进行测试,这样只能验证功能是没有问题,不能满足现场实际的业务需求。如某个项目研发了一堆数据组件,方便用户处理数据,这些数据组件

单独使用都没有问题，但是结合起来使用时，就发现错误或者数据组件功能失效的情况，而将这些组件组合起来使用才是现场人员真正需要的功能。因此，测试人员在测试的时候一定要拿现场的业务来进行测试，这样才能发现更多的问题，保证我们的产品在现场使用没有问题。

#### （2）增加产品性能测试和稳定性测试

在敏捷开发的软件项目管理流程下，提到更多的是及时响应客户的需求，快速交付用户所需的产品，但是这种快速并不是让我们以牺牲产品质量为前提的。

性能测试是软件的一种非功能性测试，是通过逐步增加系统的负载，使系统某些资源达到饱和或者失效，从而得到系统能够提供的最大服务的测试。如并发量、任务执行量等。稳定性测试是异常性测试，即发生异常情况时，系统如何反应的测试。包括断网、断电、服务器异常等情况。性能测试和稳定性测试在软件项目中是非常重要的，这些测试如果出了问题，用户会觉得软件产品是存在很大问题的。

因此，测试人员在做完每次版本的功能测试后，需要安排时间来对产品做性能测试和稳定性测试，以保证在突发的异常情况下系统能正常运行。

#### 4.2.3 增加需求验收文档

很多人说敏捷就是强调面对面沟通，都沟通好了，就不需要文档了。如果敏捷团队是小团队的话，并且产品也不复杂，几句话就能搞定的事情，讨论一下就有了结论，这个时候是不需要写文档的，写了文档也没有人去看。但是对于大一点的敏捷团队，并且团队角色比较多，写一个文档，会极大提高信息传递效率，而且当产品架构非常复杂时，如果没有文档，估计过了一段时间谁也记不清、理不清里面的业务逻辑了。总之，敏捷团队要根据自己团队的实际情况，选择最高效的开发方式。

在 Azure DevOps 工具的用户情景中除了用一句话描述需求功能和原型图外，还有一个验收条件，要求需求人员用简单明了，不会产生歧义的语言来描述用户情景验收条件，包括用户情景的正确执行以及系统的展示。当我们对用户情景测试时，系统表现与验收条件中的描述一致，就表示该用户情景实现满足要求了。

T 公司敏捷团队目前都有产品原型设计，需要增加需求验收文档的编写，后续编写用户情景时可以直接将需求验收文档中的验收条件拷贝到用户情景的验

收条件中。验收条件在用户情景中本来就是要写的，我们只是把验收条件提前写了。

将验收条件提前写有如下几个好处：

（1）编写验收条件有助于需求人员对需求细节的把握，若等研发人员研发时才发现有些细节没有考虑到，这个时候需求的改动有可能影响整个功能实现。

（2）验收条件有助于后续研发人员和测试人员进行相应的工作，没有细节，他们很可能就会按照自己的对需求的理解把功能做偏了。

（3）由于需求人员的需求分析是早于冲刺的，等冲刺时再编写用户情景的验收条件的话，需求人员很有可能忘记当初的设计思路，又要重新整理需求，这样可能会出现功能遗漏情况。

（4）用户情景的验收条件，需要考虑很多细节，编写需要一些时间，冲刺的时候才编写验收条件，可能会影响研发进度。

（5）需求分析是分析的一个完整的需求，而用户情景是拆成了一个个细粒度的功能，有一个需求验收文档，会让我们对需求功能有一个整体的把握。

因此，在产品原型设计完成后，就编写需求的验收条件是很有必要的，在需求发生变化时，需求人员需要同步修改需求验收文档，这样才能保证需求验收文档一直是最新的。

### 4.3 成员之间沟通优化方案

有效沟通的实质就是在合适的时间和合适的地点，将合适的信息通过合适的方式传递给合适的人，并产生了预期的效果<sup>[58]</sup>。T公司敏捷团队成员之间经常由于沟通不顺畅，从而出现工作上的偏差，进而影响工作进度，因此有效沟通在项目管理中起到非常重要的作用。本论文从优化沟通流程、培养沟通意识和提升沟通能力三个方面来改善T公司敏捷团队成员之间沟通不畅问题，实现团队成员之间的有效沟通，从而提升敏捷团队成员的凝聚力和团队的研发效率。

#### 4.3.1 沟通管理流程优化

##### （1）面对面沟通减少沟通偏差

敏捷开发一直都在强调面对面沟通，其实最快最有效的沟通方式就是面对

面沟通,这样能减少不必要的浪费和误会,从而加快沟通效率。

面对面沟通主要表现在三个方面:

①与他人沟通,尽量采用预定会议室来进行面对面沟通需求或问题,如果不在同一个地方,则可以采用腾讯会议或其他网络软件进行线上沟通,这样的沟通可以快速达成一致。

②进行线下或线上会议时,需要召集所有相关干系人,防止出现会后转达,因为转达可能出现偏差。

③采用直接沟通的方式,不要通过他人去转达,信息传递的层次太多,往往会造成信息失真。

#### (2) 减少沟通双方信息不对称

信息不对称是指有的成员知道一些信息,而有的成员不知道这些信息,这在日常生活中是非常常见的,信息不对称导致的后果是不知道这些信息的成员做的一些工作可能是没有用的。因此,在进行沟通之前,需要让沟通相关干系人在信息上达成一致,再进行沟通,这样即提升沟通效率,也减少了浪费。

#### (3) 做好沟通记录,确定责任人

敏捷开发弱化了文档,强调面对面的沟通,以达到沟通参与者各方达成一致意见。但是,达成一致的意见只存在沟通参与者们的短期记忆中,随着项目时间的推进,后续又有多人参与工作,沟通结果的会议纪要文档就显得非常重要了。

相信大家都有过这样的工作经历,经过激烈的沟通讨论后,大家的意见终于达成了一致,但是最后实现出来的功能和大家讨论的东西不是一个功能,追溯原因才发现,原来是没有一个人将讨论的结果告知实现该功能的研发人员,研发人员是按照自己的理解和想法实现的功能,导致的后果是工作返工,造成资源上的浪费。

因此,每次沟通会议之前,都要确定一个沟通责任人,由该责任人对沟通会议负责,并进行后续工作跟踪。我们可以用一个在线编辑文档,将需要跟踪的问题和临时发现的问题记录到在线文档中,写上相应责任人,以便后续继续跟踪。

#### (4) 建立沟通会议检查清单

团队成员在进行会议沟通前后,都要对检查清单确定一下,是否这些事项都做到了,为沟通会议的成功提供了保障。沟通会议检查清单如下:

①确保会议相关干系人是齐全的,没有遗漏的。

②确定会议相关内容提前通知了相关干系人,让他们在沟通前,信息达成一



致。

- ③确保会议的沟通方式是最佳的，优先面对面沟通，再是网络会议沟通。
- ④指定会议责任人，优先选择职位高的人，方便进行协调。
- ⑤指定会议纪要人，并在会议结束后，输出会议纪要。

#### 4.3.2 培养沟通意识

敏捷团队成员发现问题不主动汇报也不找其他成员沟通，放任其自然发展，到最后很可能出现重大失误，这是我们每一个人都不愿意看到的，即浪费时间，又浪费精力，大家做的工作也化为虚有。

因此，敏捷团队每个人都要认识到沟通的重要性，产品经理和团队成员也都要培养沟通的意识，有了沟通的意识，成员之间的顺畅沟通又向前迈了一大步。首先，产品经理作为敏捷团队的管理者，要强化自我的沟通意识，有效地促进团队成员之间的沟通，发现团队沟通中的问题并及时处理；其次，产品经理还要重视敏捷团队成员的想法和意见，以人为本，充分地发挥每个成员的潜力和价值，让他们能够在工作中获得满足感和成就感，能够不断地完善和发展自我；再次，产品经理还要提高敏捷团队成员的沟通意识，从而提高成员沟通的积极性和主动性。最后，敏捷团队成员也要增强自我沟通意识，敢于表达自己的看法，团队成员之间相互信任，协同合作，一个好的工作氛围是靠团队每一个成员努力的结果。

什么情况下，我们会完全没有沟通的想法呢？比如这个工作任务与我没有关系、不需要我做决定或不是我来做等，在这些情况下，我们是不会想着要去沟通的。

因此，产品经理可以从以下几个方面提升敏捷团队成员的沟通意识：

- (1) 需要开会讨论时，全员参与，并多问问大家的意见，尽量减少置身事外的人。
- (2) 针对比较腼腆的或者不爱沟通的成员，让其给组里其他成员培训自己做得好的或者擅长的东西，增加他们表达和说话的次数，说得多了，沟通起来就越越来越顺了。
- (3) 定期组织团建活动，加强团队成员之间的了解与沟通。
- (4) 定期组织会议，给员工反馈自己意见或者想法的渠道，从而了解员工

的想法，发现员工的问题并及时给予帮助。

### 4.3.3 提升沟通能力

沟通是否能达到有效沟通，与敏捷团队成员的沟通能力是息息相关的，敏捷团队成员要做到积极倾听、适时反馈和情绪控制，对于对方提出的意见和看法，要客观分析，科学评价，不要轻易打断，不要过多评判和指责，让对方感到被尊重和重视，并且在适当时候提出自己的看法，这样才能让沟通有效<sup>[59]</sup>。

我们与其他人沟通的时候应该牢记鼓励、询问、反应、复述和同理心这五个沟通基本原则，并将其慢慢地应用到我们的工作沟通中，我们的沟通能力也会得到提升。这五个沟通基本原则包括：

- （1）鼓励：在与人沟通时，多鼓励别人，促进对方表达的意愿。
- （2）询问：发现疑问时，可以通过询问的方式获得更多的信息。
- （3）反应：在其他成员表达想法时，要针对对方的话语给出相应的反应，告诉对方你是在听的。
- （4）复述：当不确定自己的理解是否正确时，可以通过复述的方式来确定双方理解是否一致。比如：你的意思是...？你说的是...吗？及时复述是缓解沟通失真的最有效的方法。
- （5）同理心：在沟通时，我们不能以我们的立场或者观点去思考，而是要能够站在对方立场设身处地思考和处理问题。如果大家都以达成自己的目的来进行沟通，很可能出现对方利益受损而不配合，最终导致沟通失败的情况。我们应心怀同理心，站在对方的角度来思考，这样双方才能快速达成共识。

沟通是人与人之间联系的纽带，在日常工作沟通中，我们不仅要听到别人说什么，而且还要听懂别人说什么，最终理解别人真正想要表达的是什么，这样才能达到有效沟通，从而满足沟通相关干系人的利益，快速推进项目进度，保障项目圆满成功。

## 第5章 优化方案的实施与预期效果

### 5.1 T 公司 D 项目介绍

T 公司 D 项目是开发大数据平台的, 该项目帮助电网及能源企业抽取、清洗和整合源端系统数据, 支持不同数据库类型和不同数据格式之间的转化, 对源端数据进行统一管理, 通过数据血缘关系实现数据链路的跟踪, 支持对数据进行质量校验, 也可通过可视化配置服务接口, 将数据共享给其他应用使用。大数据的概念已经在中国风风火火了很多年了, 但是如何将大数据技术应用到我们实际的企业中, 还需要不断地探索和研究。因此, 将 T 公司 D 项目作为优化方案的试点项目是非常合适的。

本论文跟踪 T 公司 D 项目实施优化方案的过程, 并对如何实施进行了论述, 由于 D 项目正在实施本论文的优化方案, 还没有看到真正的实施效果是什么样的, 所以本论文的实施效果是预期的实施效果。

### 5.2 优化方案启动

T 公司 D 项目组在进行优化之前, 进行了一系列启动工作, 为后续方案的顺利实施奠定了基础, 首先团队成员一起确定了共同目标, 这样团队每个成员有了一个共同奋斗的目标, 然后开了一个沟通专项会议, 大家一起讨论了沟通的相关问题, 再就是一起学习了项目规范, 团队后续的工作按照规范来执行, 同时确定了定期会议和定期团建的时间, 最后, 产品经理新建了几个在线编辑文档, 用于团队跟踪需求、会议、问题和培训相关工作事宜。

#### 5.2.1 确定团队共同目标

团队的目标是结合当前和未来市场需求, 经过团队成员共同讨论, 并最终达成一致的共识。团队有一个共同的目标是很重要的, 没有这个目标, 团队就会失去动力, 失去前进的方向, 最终团队的战斗力、竞争力和凝聚力也就消失不见了。

T 公司 D 项目产品经理组织团队会议，确定了如下团队目标：

产品愿景：提供电网和能源行业一站式大数据平台解决方案。

客户价值：数据资产化，充分发挥数据的价值，提升业务效率。

公司价值：储备大数据平台相关技术，提升公司产品竞争力，为公司其他产品提供大数据平台技术支撑。

团队近期目标：加强和完善现有大数据平台功能。

团队中期目标：通过多维度分析，将统计数据以报表和图形形式展示给用户。

团队长期目标：通过数据挖掘，预测未来数据，辅助用户决策。

### 5.2.2 沟通专项会议

T 公司 D 项目产品经理安排了一个团队沟通专项会议，该会议提出沟通在团队中的重要性，为有效缓解团队成员之间的沟通问题起了一个好的开端。会议具体安排如下：

（1）产品经理讲了沟通的 5 个基本原则，包括鼓励、询问、反应、复述和同理心，并以自己的实际经历来讲述了如何应用这些基本原则。

（2）团队成员每个人说说自己的性格、自己的工作、自己的工作与其他人的工作有哪些关联、在这些关联中遇到了哪些问题等。

（3）全员讨论：当我们需要别人配合怎么做？遇到意见冲突时怎么办？如何控制自己的情绪？

（4）会议结束，产品经理对会议进行总结，希望后续团队每个成员能多反馈自己的想法和意见，一个良好的工作沟通氛围，能让工作事半功倍。

### 5.2.3 管理制度的贯彻

（1）学习项目管理规范

D 项目产品经理安排敏捷团队成员一起学习项目规范，包括设计规范、开发规范、实施部署规范、问题规范和运维规范，这些项目规范在团队实施的过程中，还会根据具体的实施情况进行优化。

（2）定期组织团队沟通会议，强化团队目标共识

D 项目建立敏捷团队定期的沟通会议，确保团队工作朝着团队共同目标方向前行。定期会议如表 5.1 所示。

表 5.1 公司敏捷团队定期沟通会议

会议名称	会议频率	会议主题
每周周会	每周四下午 5:30	敏捷团队成员本周的主要工作和遇到的问题，下周的重点工作
每月月会	每月最后一周周五下午 4:00	敏捷团队成员吐槽时间，用来反馈意见和提自己的想法建议
每月培训	每月第一周周一晚上 6:00	培训工作需要用到的业务、技术、规范，提升大家技术和业务能力

（3）定期组织团队活动，进一步增强成员之间沟通

组织团队活动增强团队成员之间的沟通，包括每个月安排一天周末团建活动，如爬山、近距离的旅游等；每周三晚上安排 1~2 个小时的活动，如打羽毛球、乒乓球等。

（4）在线编辑文档

产品经理新建了四个在线编辑文档，需求跟踪矩阵用来跟踪需求的进展情况，问题跟踪记录用来跟踪日常工作中发现的问题及处理情况，会议沟通跟踪记录用来对会议过程的跟踪，培训跟踪记录用来跟踪培训需求及培训的情况。表 5.2 为在线需求跟踪矩阵，表 5.3 为在线问题跟踪记录，表 5.4 为在线会议沟通跟踪记录，表 5.5 为培训跟踪记录。

表 5.2 在线需求跟踪矩阵

版本	需求描述	需求类型	需求提出人	需求状态	备注
1.4.1	左边菜单支持鼠标拖拽	新增	XX（四川）	测试	
1.4.1	角色批量添加用户	新增	XX（四川）	测试	
1.4.1	数据目录	新增	XX（四川）	测试	
1.4.1	数据搜索	新增	XX（四川）	测试	
1.4.1	菜单目录添加搜索功能	新增	XX（四川）	测试	
1.5.0	业务授权	新增	XX（四川）	需求分析	
1.5.0	用户定制化首页	优化	XX（四川）	UI 设计	

表 5.3 在线问题跟踪记录

时间	问 题 类型	简要描述	问题发现 人	问题状态	问题处理 人	备注
2023-2-6	现场	偶尔会出现登入 报错	XXX	挂起		未重现
2023-2-8	内部	打包后启动失败	XXX	已解决	XXX	引入 jar 包导致

表 5.4 在线会议沟通跟踪记录

会议时间	会议地点	会议主题	是否相关人	是否提前通知并发资料	会议沟通方式	会议责任人	会议纪要人
2023-3-1	三楼会议室	多个平台统一登入	是	是	线上	XXX	XXX
2023-3-9	三楼会议室	元数据管理优化	是	是	线下	XXX	XXX

表 5.5 在线培训跟踪记录

培训时间	培训地点	培训内容	培训人	是否按期培训	下次培训时间	备注
2023-3-6 18:00:00	四楼大会 议室	liquibase 自动 升级数据库	XXX	是		
2023-4-3 18:00:00	四楼大会 议室	Mongodb 数据 库使用	XXX			
2023-5-8 18:00:00	四楼大会 议室	ETL 如何增加 组件开发	XXX			

### 5.3 优化方案实施

在 T 公司原有软件项目管理流程的基础上增加优化方案，包括现场人员录入原始需求、需求人员坚持需求基本原则、需求人员需要编写验收文档、优化产品经理待办列表优先级排序、增加用户情景的拆解任务（研发充分自测、架构师代码评审、产品经理验收测试）、现场人员收到版本后及时进行验收和反馈等，产品经理对流程的实施过程进行全程监督，只有完成了相应的流程节点，才能进行到下一个流程节点。T 公司优化后软件项目管理流程见如图 5.1 所示。

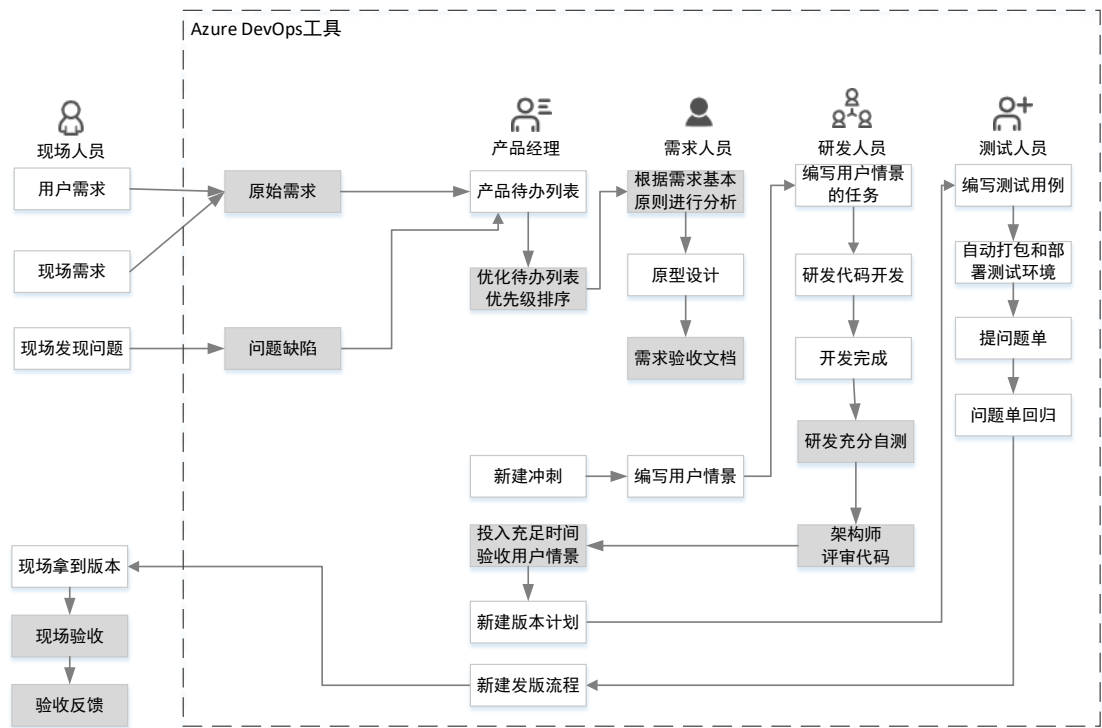


图 5.1 T 公司优化后软件项目管理流程

5.3.1 现场人员工作优化

(1) 现场人员录入原始需求

敏捷团队不再接受现场人员通过微信、微信群、电话等方式提用户需求和问  
题，现场人员必须按照 T 公司统一规定，将用户需求和问  
题录入到 Azure DevOps 工具中，在录入需求的过程中，有助于现场人员对用户需求的深入思考，同时便  
于后续对用户需求的跟踪。如图 5.2 所示为 T 公司录入的原始需求。

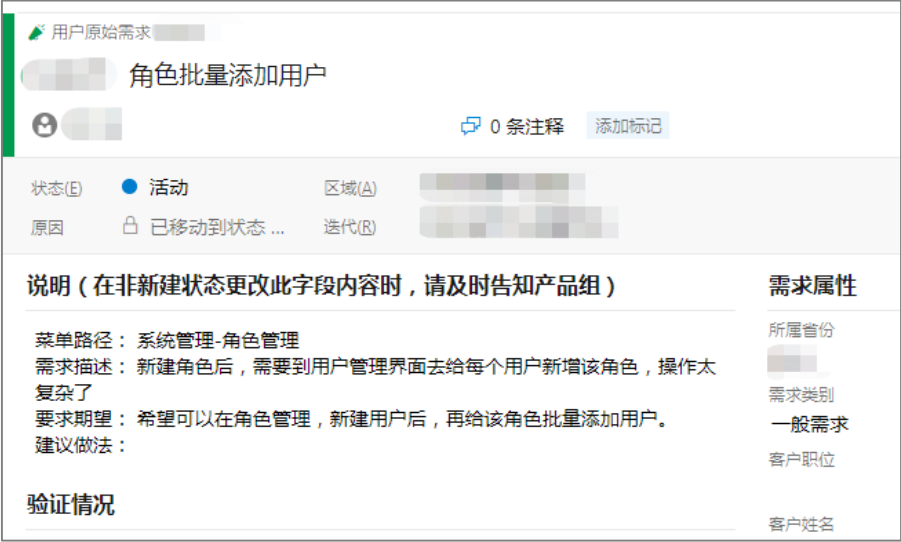


图 5.2 Azure DevOps 工具的原始需求

(2) 及时验收发布版本

现场人员拿到发布版本后，对着 Azure DevOps 工具上自己的录入原始需求和问题单，看版本是否满足了要求，并反馈相关的意见给研发团队。

5.3.2 产品经理工作优化

(1) 跟踪现场验收和反馈

在版本发布到现场后，产品经理需要及时询问现场的部署情况，让现场及时对该版本的用户情景进行验收，有新的需求或者该版本中发现的问题让现场人员继续录入到 Azure DevOps 工具的原始需求和问题单，从而达成需求和问题及时闭环。

(2) 优化产品待办列表优先级排序

需求优先级排序优先考虑重要紧急程度，重要紧急程度排序为重要紧急>重要不紧急>紧急不重要>不紧急不重要，再根据估算工作量（天）和需求之间的前置条件来确定需求优先级顺序，最后根据需求优先级顺序将需求放到对应的敏捷开发冲刺中。但是有一种特殊情况除外，如果这个需求的优先级顺序比较前，但是该冲刺剩余的工作时间完不成该需求，那么这个需求则要放入下一个冲刺中。如表 5.6 是对 T 公司 D 项目需求优先级排序情况。



表 5.6 T 公司 D 项目需求优先级排序

需求简要描述	需求类型	重要紧急程度	估算工作量(天)	前置条件	需求优先级	迭代计划	完成情况
左边菜单支持鼠标拖拽	用户	重要紧急	1		1	迭代 1	进行中
角色批量添加用户	用户	重要紧急	3		2	迭代 1	进行中
业务授权	现场	重要紧急	3		3	迭代 1	进行中
用户定制化首页	现场	重要紧急	8	业务授权	4	迭代 1	进行中
数据搜索	现场	重要紧急	8		5	迭代 2	未开始
数据目录	现场	重要紧急	3		6	迭代 1	进行中
系统报表自动生成	现场	重要紧急	2		7	迭代 2	未开始
基础框架搭建	优化	重要紧急	8		8	迭代 2	未开始
框架深色系优化	优化	重要不紧急	5		9	迭代 2	未开始
框架浅色系优化	优化	重要不紧急	5		10	迭代 2	未开始

### 5.3.3 需求人员工作优化

#### (1) 需求人员坚持需求分析基本原则

在做需求分析时,需求人员一定要坚持需求的基本原则,即用户为什么要做这个需求,如果只是被“用户要求”牵着走,即使功能完成了,可能用户遇到的问题还是没有解决。很多人其实都是这样的,只知道有问题,却抓不住问题的核心,从产品的角度来说,深入了解用户和获取用户的真正需求才是做产品的王道。需求是由一个问题引出的,并且存在一定隐蔽性,需要通过深层次的分析之后,才能得出用户真正需求。

需求人员对于现场人员在 Azure DevOps 工具提的原始需求,首先根据原始需求中的需求描述(即问题描述)和要求期望(即用户期望需求)进行初步分析,然后组织一个需求讨论会议,双方沟通一下需求,主要是要问为什么会提这个需求,会议结束后输出达成一致意见的会议纪要。需求人员会根据需求会议的结论形成原型,然后组织原型讨论会议,与现场人员或用户确定该原型是否能满足他们的需求,会后,根据意见修改原型,再进行原型讨论会议,重复这个过程,直到得到最终认可的需求原型。在这个过程中,需求人员不要怕麻烦,同时也要防

止用户在反复确认的过程中出现疲软状态，用户的全程参与，不仅可以减少需求变更，而且能交付用户的真正需求，很大程度上减少了研发成本。如图 5.3 所示为需求分析流程。

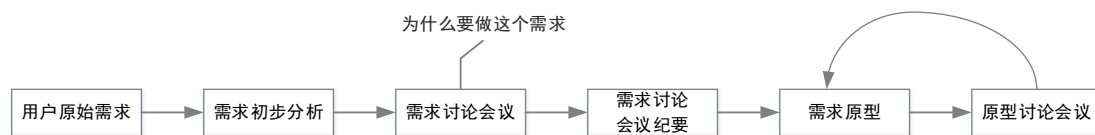


图 5.3 需求分析流程

除了从现场人员录入 Azure DevOps 工具的原始需求获取需求外，需求人员每个月还会组织与用户访谈的会议，定期从用户那里获取需要改进的功能和新需求。访谈会议的内容包括用户使用系统时哪些功能用起来满意、哪些不满意、哪些使用起来有点别扭等，通过访谈的方式从中挖掘出用户的需求，也需要录入到 Azure DevOps 工具的原始需求进行统一跟踪。每月定期访谈会议会和用户确定好每月最后一周进行，在最后一周的周一会和用户确定好访谈会议时间、地点、参与人员、访谈内容等，让用户提前准备好，开一个高效的需求会议。

#### （2）需求人员编写需求验收文档

需求人员除了画产品原型，还需要编写验收文档，其中产品原型用于展示需求功能的外形，验收文档一方面用于指导研发人员进行开发，另一方面用于开发人员和测试人员进行功能测试。产品原型和验收文档在冲刺的用户情景中都会体现出来，将验收文档中的验收条件拷贝到 Azure DevOps 工具的用户情景的验收条件中，如图 5.4 所示为角色批量添加用户需求的产品原型，如图 5.5 所示为角色批量添加用户需求的用户情景。

分配用户

角色名称：管理员

输入用户名

Q

☐ ☒ 集控中心

☒ 张三

☐ 李四

☐ 王五

☒ 自动化部门

☒ 小明

☐ 小红

☐ 研发部门

☐ 丁一

☐ 甲二

☒ 小李

☐ 小马

»

«

<input type="checkbox"/>	序号	用户名	部门
<input type="checkbox"/>	1	张三	集控中心
<input type="checkbox"/>	2	小明	自动化部门
<input type="checkbox"/>	3	小李	-

取消

确定

图 5.4 产品原型



图 5.5 Azure DevOps 工具的用户情景

5.3.4 研发人员工作优化

在 Azure DevOps 工具的任务面板中一个用户情景会建多个子任务，完成这些子任务，这个用户情景也就完成了。一个用户情景的子任务除了原有的研发任务外，还需要增加研发自测、架构师评审代码、产品经理验收用户情景三个子任务，只有这些子任务全部完成的情况下，才算这个用户情景完成，从流程上约束敏捷团队成员。如图 5.6 所示为用户情景拆分的子任务。



图 5.6 用户情景拆分子任务

5.3.5 测试人员工作优化

(1) 测试环境保持与现场环境一致

将 D 项目 Azure DevOps 工具自动部署的测试环境部署成与现场环境一致，首先将单节点服务器改成集群服务器，然后将单节点数据库部署改成主备数据库数据，最后在 linux 服务器上部署 windows 虚拟机，研发人员和测试人员测试系统时连接该 windows 环境进行测试。因为用户现场出于安全的考虑，所有访问系统的人员都只能通过这个 windows 虚拟机，这种 windows 虚拟机最大的特点是没有显卡，系统不能有太多的页面渲染功能，否则会出现页面刷新太慢的情况。图 5.7 为 D 项目测试环境部署变化。

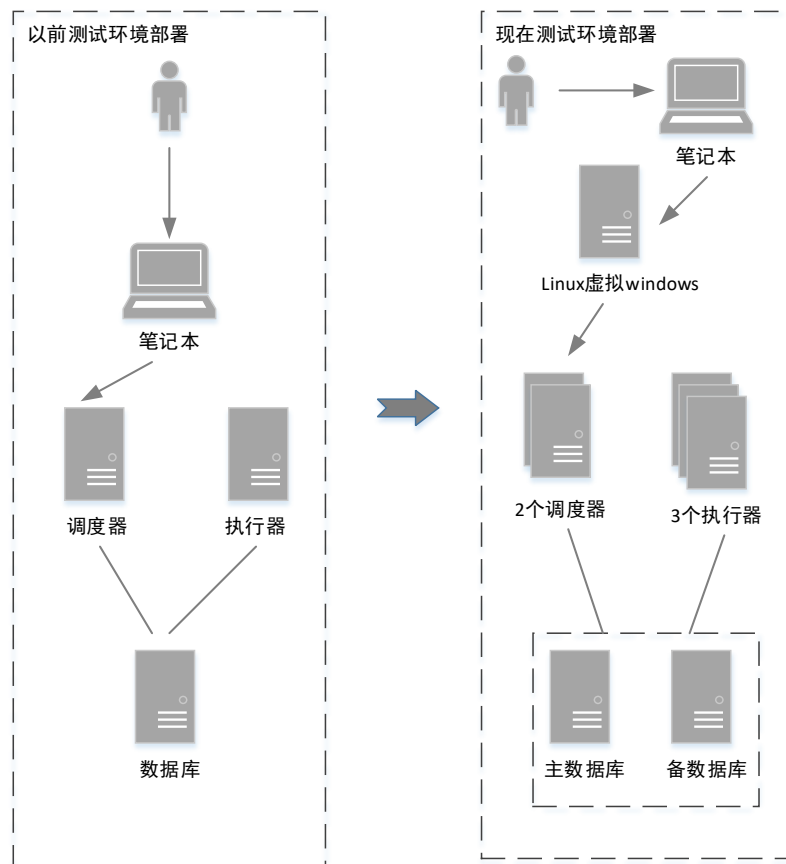


图 5.7 测试环境部署变化

### （2）进行稳定性测试

稳定性测试是在保证系统功能正常的前提下，系统在内外部条件发生异常时，验证系统在突发异常时能否快速做出响应，以减少系统崩溃的发生。D 项目根据自身系统的特点，进行了一些稳定性测试包括 CPU 负载高、内存异常、磁盘空间不足、网络断开、网络波动、数据库重启、数据库断开、数据库连接数过多等，D 项目稳定性测试专门安排测试人员在版本发布后进行测试并记录测试结果。

### （3）进行性能测试

性能测试是针对系统的性能指标而进行的模拟测试，最终得出性能结果来判断系统的性能指标是否满足要求，因此性能测试也叫压力测试或极限测试。D 项目根据用户要求，进行了系统登入响应时间、简单业务响应时间、复杂业务响应时间、接口调用响应时间等性能测试。D 项目系统性能测试专门安排测试人员在版本发布后进行测试并记录测试结果，还需要将测试结果与各现场用户性能

要求进行对比，判断性能指标是否满足客户要求。

5.4 预期实施效果

5.4.1 产品质量提升

T 公司 D 项目实施改进方案后，在研发过程中增加研发自测时间、代码评审及用户情景验收等任务后，产品质量得到了明显改善。测试人员在测试阶段提的问题单数量大幅降低，很多问题在研发自测和用户情景验收的时候就暴露了出来，大大提升了产品交付质量。如图 5.8 所示对项目实施方案前后的问题单数量的统计分析。

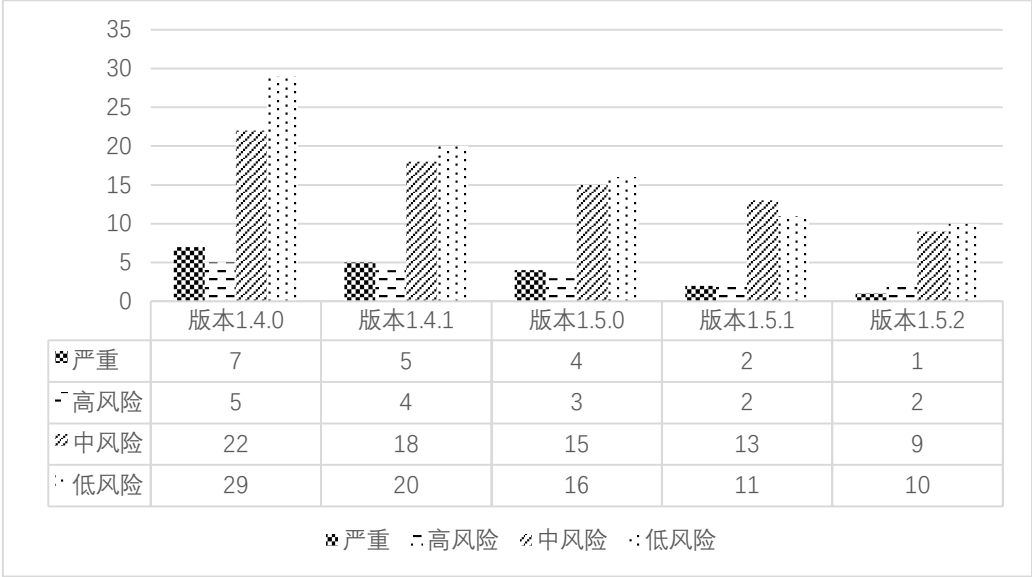


图 5.8 产品版本问题单统计

5.4.2 团队沟通良好

产品经理通过定期会议沟通和团队建设活动，敏捷团队成员之间相互了解的更多了，关系也密切了很多。在进行会议时，敏捷团队成员比以前更多地表达了自己的意见和想法，团队之间沟通效率和效果明显提高，并且由于成员之间沟通无障碍，让工作更快更顺利的得到了完成。

### 5.4.3 员工抱怨减少

T公司D项目实施改进方案后,团队成员间协作更加紧密了,在工作中,团队成员之间的抱怨明显减少了,幸福感得到了很大的提高。员工在以下几个方面的抱怨得到了减少,如表5.7所示。

表 5.7 敏捷团队成员抱怨减少的内容项

敏捷团队成员	优化前抱怨内容
现场人员	项目团队实现的功能并不是我想要的。 给现场的版本一堆问题。 实现出来的功能用起来不好用。
产品经理	又被现场人员投诉了,就不能好好和我沟通。 测试怎么又提这么多问题单。 怎么成员之间一点点工作都需要我来协调,难道我很闲? 怎么现场人员每次项目结束才做验收,现在提这么多问题和新需求,哪里有人做哦。
需求人员	现场人员就不能好好理解客户需求后再来提需求吗。 这不就是现场人员自己提的需求吗?怎么现在不认了。 研发人员的实现和我的设计不一致呢。
研发人员	测试人员是不是看我不顺眼?怎么提这么多问题单。 其他人怎么这么写代码,这让我怎么改。
测试人员	就一个原型设计我怎么测试哦。 现场的测试环境怎么和我的测试环境不一样。 怎么这么多问题,我的测试进度要延期了。

### 5.4.4 团队工作效率提高

利用规范来进行项目管理后,为团队带来了多方面的变化,尤其是团队工作效率方面,得到了很大的提高。在项目开发过程中,敏捷团队成员都遵守统一的规范来进行开发工作,成员之间的工作协作更加高效了,每个成员都能充分发挥自身的优势,工作也变得更加积极主动了,整个敏捷团队都保持着较高的工作热情。

### 5.4.5 用户满意度提升

在迭代开发过程中,由于与用户保持紧密的沟通,这样就能及时地获取用户



的新需求和新思路。随着项目的推进，D 项目的产品功能越来越符合用户的要求。D 项目正式上线后，通过问卷调查的方式对用户满意度进行调查，并对优化前后的结果进行了对比，优化后用户满意度明显得到了提升。如表 5.8 为优化前后用户满意度得分，每项满分为 10 分。

表 5.8 用户满意度得分

评价项	优化后评分	优化前评分
产品界面设计	9	7
产品功能	8	7
产品质量	9	7
团队协作程度	8	6
处理问题的速度	8	7

## 第6章 结论与展望

### 6.1 结论

随着敏捷开发大规模应用到我国软件企业软件项目管理中,为了更好地服务于客户,同时也为了提升企业自身的竞争力,T公司也在多年前引进了敏捷开发来进行项目管理。敏捷开发也帮助T公司在快速响应客户需求、研发效率等方面取了不错的效果,但是T公司在使用敏捷开发的过程中还是遇到了各种问题,这些问题包括用户需求对接问题、质量不好等。

本论文基于项目管理、软件项目管理和敏捷开发理论的研究,对国内外应用敏捷开发软件项目管理的文献进行了梳理和汇总,对T公司应用敏捷开发管理项目存在的问题进行了访谈、主要原因分析和深挖问题根本原因,以前期研究的理论为基础,针对T公司的实际情况,提出了相应的改进方案,并将改进方案实施到T公司的D项目中,并取得了不错的效果,提升了客户满意度和产品质量。本论文主要得到以下结论:

(1) 基于敏捷开发进行软件项目管理模式的深入研究,并以使用管理学研究方法访谈法对T公司存在的问题进行分析,得到了T公司存在的项目管理问题清单。通过层次分析法对问题清单进行量化分析,得到了问题清单的严重程度排序。再利用原因鱼骨图分析法对问题清单中的主要问题进行了根本原因分析,得到了主要问题发生的根本原因。这样不仅给T公司解决项目管理问题确定了优化方向,也为同行业解决同类问题提供了参考和借鉴。

(2) 本论文经过分析识别发现了T公司项目管理存在的问题,结合项目管理、软件项目管理和敏捷开发基本理论与思想,对T公司所存在的问题提出了改进方案。敏捷团队中的每一个成员都有需要改进的地方,需求人员提升能力、坚持需求原则“不看怎么做,而看为什么做”和在Azure DevOps工具上详细描述用户情景验收条件;现场人员在Azure DevOps工具上提原始需求和及时验收需求功能;产品经理建立需求跟踪机制、投入足够的时间进行用户情景验收、每次沟通确定沟通责任人;研发人员加强沟通和功能测试;测试人员保证测试环境与现场一致,并进行性能和稳定性测试;架构师确保投入足够的时候进行代码评

审。

(3) 本论文将前面研究的改进方案运用于 T 公司 D 项目中, 并对 D 项目的实施过程进行了跟踪, 跟踪发现改进方案对于 D 项目软件项目管理的改进具有显著提升效果。这样改进方案不仅帮助 D 项目在管理方面得到了改进, 也为 T 公司其他项目的推广提供了实践经验。

## 6.2 展望

敏捷开发不是万能的, 不是一套行之有效的方法, 生搬硬套很可能适得其反, 徒劳无功, 只有找到适合自己企业的项目管理方法, 才能提升研发效率, 提高用户满意度和企业竞争力。

虽然本论文作者在本次研究过程中付出了很多努力, T 公司 D 项目在实施改进方案后也取得了显著效果, 但是对本论文的成果进行进一步的研究分析具有十分重要的实用价值, 主要体现在以下三个方面:

(1) 由于时间原因, T 公司的 D 项目敏捷团队还在实施优化方案的过程中, 没有办法得出 T 公司 D 项目优化前后效果的对比结果, 在后续的工作中, 才能得出对比结果。

(2) 本论文对需求管理、质量管理和沟通管理等方面进行了详细的研究, 但是并未涉及项目管理知识体系的其他方面, 所以在成本管理、进度管理等还是存在很大的研究空间的。

(3) 敏捷开发本身就是一个不断实践、不断改进的软件项目管理过程, 这对 T 公司软件项目管理优化具有很强的实际意义。

## 参考文献

- [1] Gren L, Goldman A, Jacobsson C. Agile ways of working: A team maturity perspective[J]. Journal of Software: Evolution and Process, 2020, 32(6).
- [2] Amorim L F, Marinho M, Sampaio S. How (UN) Happiness Impacts on Software Engineers in Agile Teams?[J]. International Journal of Software Engineering & Applications, 2020, 11(3):39-57.
- [3] Tam G, Moura E J D A, Oliveira T, Varajão J. The factors influencing the success of on-going agile software development projects [J]. International Journal of Project Management, 2020, 38(3): 165-176.
- [4] Cram. Agile Development in Practice: Lessons from the Trenches[J]. Information Systems Management, 2019, 36(1):2-14.
- [5] Kuengjai Y, Ramingwo L. A Pilot Study of Requirement Prioritization Techniques in Agile Software Development[C]. 2021 4th International Conference on Computer Science and Software Engineering(CSSE 2021), Singapore, 2021.
- [6] Shim W, Lee S W. An agile approach for managing requirements change to improve learning and adaptability[J]. Journal of Industrial Information Integration, 2019, 14:16-23.
- [7] Arcos-Medina G, Mauricio D. Aspects of software quality applied to the process of agile software development: a systematic literature review[J]. International Journal of System Assurance Engineering and Management, 2019, 10(5):867-897.
- [8] Mathews L, Kai Holzweißig K. Agile Dokumentation: Anspruch und Wirklichkeit – eine fallstudienbezogene Analyse mehrerer Softwareentwicklungsprojekte Agile documentation: theory and reality – a case study based analysis of various software engineering projects[J]. HMD Praxis der Wirtschaftsinformatik, 2020, 57(1):117-132.
- [9] 丁慧,余亚萍,陈杰. 敏捷思想在软件研发中的研究与实践[J]. 航天工业管理,2021,(08):18-22.
- [10] 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.
- [11] 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.
- [12] 杨帆.敏捷开发流程管理优化探讨[J]. 电子技术与软件工程,2015,(20):52-53.
- [13] 王丹. DD 公司基于敏捷方法的研发项目流程优化研究[J]. 中国管理信息化,2018,21(02):55-56.
- [14] 黄欣. 对敏捷软件开发过程改进的行动框架分析[J]. 电子测试,2018,(01):75-76.
- [15] 张东红,刘丹,王振. 基于敏捷开发的 Scrum 模型的改进[J]. 信息技术与信息化,2021,(11):84-86.
- [16] 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.
- [17] 黄晨晨.基于精益思想的 SMIS 软件项目管理优化研究[D]. 中国矿业大学,2019.

- [18] 闫鹏斌,吴扬,丘仕能. Agile 软件开发研究综述[J]. 河南教育学院学报(自然科学版),2020,29(01):27-31.
- [19] 黄敏珍.CMMI、敏捷开发和 DevOps 在项目管理实践中的应用[J]. 项目管理技术,2020,18(09):91-95.
- [20] 龚桂芬,龚兰兰. Devops 的软件项目开发实践[J]. 工业控制计算机,2022,35(04):87-89.
- [21] 王海军. 软件项目的需求管理问题及对策[J]. 中国管理信息化,2015,18(02):84.
- [22] 杨东. 层次分析法在软件项目质量影响因素分析中的应用[J]. 项目管理技术,2021,19(06):155-158.
- [23] 王蕾. 项目管理中沟通管理在实践项目中的应用[J]. 河北企业,2021,(02):93-94.
- [24] 杨俊. 敏捷需求管理原则与实践[J]. 项目管理评论,2021,(04):79-82.
- [25] 彭昊.需求管理在敏捷开发中应用方式的思考[J]. 中国金融电脑,2021,(01):76-79.
- [26] 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.
- [27] 孙子谦,王雅琴,黄明明. 戴明循环在敏捷软件质量管理中的应用方法研究[J]. 计算机应用与软件,2016,33(11):8-10.
- [28] 束勉,杜宇. 谈自组织敏捷研团队下的质量管理作用[J]. 数字通信世界,2018,(05):216.
- [29] 黄凌凡. 沟通管理——敏捷项目中尤为重要的一环[J]. 信息系统工程,2008,(12):48-49.
- [30] (美)项目管理协会. 项目管理知识体系指南(第六版)[M]. 北京: 电子工业出版社,2018:4-10.
- [31] 马明军,肖萍. 项目的历史由来及其发展趋势[J]. 中国房地产业,2011,(03):89-90.
- [32] 李波. 项目管理理论综述与前沿问题研究[J]. 经济研究导刊,2012,(31):212-214.
- [33] 崔星花,汪洪亮,全晶,等. 关于项目管理知识体系内容与层次结构的思考[J]. 中国商论,2020,(24):119-120.
- [34] 姜茸. 软件项目管理最新研究综述[J]. 项目管理技术,2011,9(10):35-39.
- [35] 魏世雄. 软件项目管理中面临的七个问题[J]. 硅谷,2010,(11):133.
- [36] 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.
- [37] 张家欣,许树华,秦爽. 试论软件项目管理问题与改善对策[J]. 信息系统工程,2016,(12):46.
- [38] 韩万江,姜立新. 软件项目管理案例教程(第4版)[M]. 北京: 机械工业出版社,2019:5.
- [39] 罗孟华,李军,黄益辉. 瀑布模型在软件开发中的应用及其局限性[J]. 才智,2009,(09):251.
- [40] 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.
- [41] 肖建芳. 快速原型模型在学生选课系统中的应用[J]. 现代计算机,2020,(25):97-100.
- [42] 孙丽. 经典软件开发模型综述[J]. 产业与科技论坛,2014,13(15):94-95.
- [43] 蒋丹,刘永吉. 基于模型的敏捷软件架构设计方法[J]. 电子技术与软件工程,2019,(04):31-33.
- [44] 杨喆. 软件项目管理中的关键问题分析及对策[J]. 计算机产品与流通,2017,(09):38.
- [45] 袁辉. 规范敏捷应用实践[J]. 项目管理评论,2022,(01):80-82.
- [46] 陈新征,蔡琳琳,VitSopal. 敏捷,让大企业不再迷茫[J]. 通信企业管理,2021,(10):10-15.

- [47] 赵俊,石春. 敏捷思想在软件开发中的应用与实践研究[J]. 电脑知识与技术,2020,16(09):98-99.
- [48] 王娟. 敏捷项目管理提高管理灵活性和适应性的研究[J]. 中国集体经济,2021,(13):45-46.
- [49] 商志伟. 敏捷开发研究[J]. 电子世界,2018,(07):51-52.
- [50] 梁旭. 浅谈 Scrum 方法在软件项目技术中的应用[J]. 中国战略新兴产业,2018,(36):142.
- [51] 郭素芹,林小军,刘伟. 精益思想在软件质量保证中的应用[J]. 现代信息技术,2017,1(03):74-76.
- [52] 孙圣领. 精益思想在软件项目管理中的应用探索[J]. 中国新技术新产品,2014,(07):147.
- [53] 蔡宗慧. 极限编程在软件项目开发中的应用分析[J]. 电脑与电信,2016,(07):84-86.
- [54] 邹筱菁,柯林. 基于 DevOps 的软件开发管理模式[J]. 数字技术与应用,2016,(11):184-187+217.
- [55] 王红蕾. 基于 DevOps 的轻量级持续交付方案[J]. 计算机系统应用,2020,29(09):87-94.
- [56] 高路. 基于层次分析法的煤矿井底配仓方案评价[J]. 陕西煤炭,2022,41(05):185-188.
- [57] 皮圣雷. 综合鱼骨图及其在项目管理中的应用研究[J]. 中国软科学,2009,(04):92-97.
- [58] 闫荣. 神一样的产品经理[M]. 北京: 电子工业出版社,2012:136-180.
- [59] 沈凤蕾. 企业管理沟通问题与对策研究[J]. 商场现代化,2022,(09):91-93.

## 附录 A Excel 第二层次分析法计算公式

	A	B	C	D	E	F	G	H	I	J
1	矩阵									
2	阶数									
3	5									
4		A	B	C	D	E		$\bar{W}_i$	$W_i$	$AW_i/W_i$
5	A	1	2	2	3	5		=POWER((PRODUCT(B4:F4)),1/\$A\$2)	=H4/\$H\$10	=(B4*I\$4+C4*I\$5+D4*I\$6+E4*I\$7+F4*I\$8)/I4
6	B	=1/C4	1	2	2	4		=POWER((PRODUCT(B5:F5)),1/\$A\$2)	=H5/\$H\$10	=(B5*I\$4+C5*I\$5+D5*I\$6+E5*I\$7+F5*I\$8)/I5
7	C	=1/D4	=1/D5	1	2	3		=POWER((PRODUCT(B6:F6)),1/\$A\$2)	=H6/\$H\$10	=(B6*I\$4+C6*I\$5+D6*I\$6+E6*I\$7+F6*I\$8)/I6
8	D	=1/E4	=1/E5	=1/E6	1	2		=POWER((PRODUCT(B7:F7)),1/\$A\$2)	=H7/\$H\$10	=(B7*I\$4+C7*I\$5+D7*I\$6+E7*I\$7+F7*I\$8)/I7
9	E	=1/F4	=1/F5	=1/F6	=1/F7	1		=POWER((PRODUCT(B8:F8)),1/\$A\$2)	=H8/\$H\$10	=(B8*I\$4+C8*I\$5+D8*I\$6+E8*I\$7+F8*I\$8)/I8
10								$\bar{W}_i$ 的和	=SUM(H4:H8)	$\lambda_{max}$
11										=AVERAGE(J4:J8)
12										CI=(J10-A2)/(A2-1)
13										RI 1.12
										CR=J11/J12