

硕士学位论文

基于敏捷开发的 J 公司软件项目管理优化方案研究

Research on Software Project Management
Optimization Plan of J Company Based on Agile
Development

学科专业

工商管理（MBA）

专业领域

项目管理

作者姓名

指导教师

2021 年 10 月

中图分类号_____

学校代码_10533_____

UDC _____

学位类别_专业学位_____

硕士学位论文

基于敏捷开发的J公司软件项目管理优化方案研究

Research on Software Project Management Optimization Plan of J Company Based on Agile Development

作者姓名

学科专业

工商管理（MBA）

专业领域

项目管理

研究方向

软件项目管理

二级培养单位

商学院

指导教师

副指导教师

论文答辩日期_____

答辩委员会主席_____

中南大学

2021年10月

基于敏捷开发的 J 公司软件项目管理优化方案研究

摘要:作为金融软件行业排名靠前的金融科技研发公司，经过二十余载的快速发展，J 公司的研发团队一直使用专业化的项目管理方法来支持集团公司业务的发展及壮大，但是为了传统金融行业与互联网金融能够相互融合，寻求满足客户需求的快速发展，在系统更替及维护的过程中，经常会出现大量的三司三所新规导致需求变更频繁，响应不够及时，成员沟通不畅，文档不规范，周期长等各种常见问题，为了解决这些问题，本文对现有项目管理方法进行优化，减少客户的流失，增加公司的盈利。

本文通过对 J 公司现有的敏捷开发软件项目管理方式进行研究，使用沟通管理的 5C 原则、进度管理的紧前关系绘制图方法、成本管理的挣值管理方法、质量管理的持续增量改进方法作为指导，对 J 公司的敏捷软件开发模式 Scrum Board 和 Kanban 进行了分析，并针对现有项目具体实施设计了相应的问卷调查，最后根据问卷调查结果归纳总结，发现存在有以下几大影响敏捷项目管理方法的问题：（1）组织架构：缺少稳定的实施人员；（2）沟通：对接业务系统较多，缺少具体对接人负责解答问题；（3）进度：管理工具不完善，缺少整体把控工具；（4）成本：人员结构安排不合理导致成本增加，需求返工导致项目周期长，评审环节出现低级错误浪费评审时间；（5）质量：质量把控缺失重要的评审环节，进而通过这些问题找到合适的解决方案并制定相应的实施过程。为了保障这些敏捷优化方案的正常实施，提出了组织架构调整、公司规范与及绩效相结合、公司高层的大力支持等保障措施，优化后的敏捷项目管理能够正常保障质量提交，需求返工少，可以正常按照研发进度部署上线，最终赢得了客户的高度认可，因此也为公司的盈利贡献了一份力量。

此文主要是通过实际案例的研究来证明新的优化方案的适用性，能够更好的完善项目中敏捷软件开发项目管理的问题，找到适合的方法，并且希望能够对 J 公司其他的集团子公司的软件项目管理提供参考，供其他项目组借鉴并使用。

图 22 幅，表 14 个，参考文献 50 篇

关键词：敏捷软件开发；迭代；JIRA；软件项目管理

Research on Software Project Management Optimization Plan of J Company Based on Agile Development

Abstract : As a top-ranked financial technology R&D company in the financial software industry, after more than 20 years of rapid development, the R&D team of J company has been using professional project management methods to support the development and growth of the group company's business, but for the traditional financial industry It can be integrated with Internet finance to seek rapid development that meets customer needs. In the process of system replacement and maintenance, there are often a large number of new regulations of the three departments and three institutes, resulting in frequent demand changes, insufficient response, poor communication between members, and documentation Various common problems such as irregularities and long cycles. In order to solve these problems, this article optimizes the existing project management methods to reduce the loss of customers and increase the company's profitability.

This paper studies the existing agile development software project management methods of J company, using the 5C principles of communication management, the immediate relationship drawing method of schedule management, the earned value management method of cost management, and the continuous incremental improvement method of quality management. As a guide, the company J's agile software development model Scrum Board and Kanban were analyzed, and corresponding questionnaires were designed for the specific implementation of existing projects. Finally, based on the results of the questionnaire survey, it was concluded that there were the following major impacts on agile projects Problems with management methods: (1) Organizational structure: lack of stable implementation personnel; (2) Communication: There are many docking business systems, and there is a lack of specific docking personnel responsible for answering questions; (3) Progress: management tools are not perfect, and overall control is lacking Tools; (4) Cost: unreasonable personnel structure arrangement leads to increased costs, needs rework leads to long project cycles, low-level errors in the review process wastes review time; (5) Quality:

quality control lacks important review links, and then passes these problems Find a suitable solution and develop a corresponding implementation process. In order to ensure the normal implementation of these agile optimization solutions, safeguard measures such as organizational structure adjustment, the combination of company specifications and performance, and the strong support of the company's senior management have been proposed. The optimized agile project management can normally guarantee quality submissions and require less rework. Normally deployed and launched according to the research and development schedule, and finally won the high recognition of customers, so it also contributed to the company's profitability.

This article mainly proves the applicability of the new optimization scheme through actual case studies, can better improve the problem of agile software development project management in the project, find a suitable method, and hope that it can be used for other group subsidiaries of J company Provide reference for software project management for other project teams to learn from and use.

Keywords : Agile software development; Iteration; JIRA; Software project managemen

目 录

摘要.....	I
Abstract.....	II
目录.....	IV
第 1 章 绪论.....	1
1.1 研究背景和研究意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	3
1.2 国内外研究现状.....	3
1.2.1 国外研究综述.....	3
1.2.2 国内研究综述.....	5
1.2.3 文献评述.....	6
1.3 研究思路和研究方法.....	6
1.3.1 研究思路.....	6
1.3.2 研究方法.....	7
1.4 研究内容和研究框架.....	7
1.4.1 研究内容.....	7
1.4.2 研究框架.....	8
第 2 章 相关理论基础.....	9
2.1 项目管理.....	9
2.1.1 项目管理概念.....	9
2.1.2 项目管理过程.....	9
2.2 敏捷软件开发.....	11
2.2.1 敏捷软件开发方法.....	11
2.2.2 敏捷软件开发介绍.....	12
2.2.3 敏捷软件开发项目管理流程.....	13
2.2.4 敏捷软件开发优点和缺点.....	15
2.3 软件项目管理模型.....	16
2.3.1 瀑布模型.....	16
2.3.2 增量开发模型.....	16
2.3.3 各模型之间分析比较.....	17
第 3 章 J 公司项目管理现状与问题分析.....	19

3.1 J 公司概况以及软件项目介绍.....	19
3.1.1 公司概况.....	19
3.1.2 组织架构.....	19
3.1.3 软件项目管理流程.....	21
3.2 J 公司现有敏捷项目管理分析.....	22
3.2.1 项目介绍.....	22
3.2.2 J 公司现有敏捷软件项目管理.....	23
3.2.3 设计问卷调查分析影响 J 公司软件项目管理的因素.....	24
3.3 归纳总结项目管理存在的主要问题.....	25
3.3.1 缺少固定工程人员的组织架构问题.....	25
3.3.2 基于紧前关系绘制图的进度管理问题.....	25
3.3.3 基于挣值管理的项目成本管理问题.....	27
3.3.4 基于持续增量改进项目质量管理问题.....	28
3.3.5 基于 5C 原则的项目沟通问题.....	30
第 4 章 基于敏捷开发 J 公司项目管理优化方案设计.....	32
4.1 基于敏捷开发优化方案设计原则与思路.....	32
4.1.1 敏捷开发优化方案设计的基本原则.....	32
4.1.2 敏捷开发优化方案设计的总体思路.....	32
4.2 基于敏捷开发软件项目管理优化的方案设计.....	33
4.2.1 增加固定工程人员的组织架构方案设计.....	33
4.2.2 敏捷开发基于进度网络分析的进度管理方案设计.....	34
4.2.3 敏捷开发基于净值管理的成本管理方案设计.....	36
4.2.4 敏捷开发基于持续增量改进的质量管理方案设计.....	43
4.2.5 敏捷开发基于 5C 原则的项目沟通方案设计.....	49
第 5 章 基于敏捷开发 J 公司项目管理优化方案实施与保障措施.....	51
5.1 敏捷开发优化方案实施.....	51
5.1.1 应用敏捷开发方法的项目特点.....	51
5.1.2 应用敏捷开发实施方案.....	51
5.1.3 应用敏捷开发实施过程.....	51
5.1.4 应用敏捷开发优化预期效果.....	59
5.2 敏捷开发优化方案保障措施.....	60
5.2.1 组织架构调整.....	60
5.2.2 制度规范与绩效相结合.....	61
5.2.3 公司高层的支持.....	63

第 6 章 结论与展望.....	64
6.1 研究结论.....	64
6.2 研究展望.....	65
参考文献.....	66
附录.....	70

第1章 绪论

1.1 研究背景和研究意义

1.1.1 研究背景

新世纪新时代的到来，在互联网+金融模式背景的巨大影响下，随着信息技术的不断发展和扩大，信息化产业已经成为了我国一项重要的国民经济的产业支柱，信息技术快速崛起开拓了人们的思维和思想，因此，证券公司的需求越来越种类繁杂，并且也越来越多变，金融业务正在通过“线上+线下”的模式开展开来。近年来，互联网+金融模式的引入及客户互联网思维的逐渐形成，客户为了便捷都不太想通过临柜的模式办理业务，希望通过手机、pad等网上自助的模式解决，因此券商对于客户的服务提供也逐渐的往线上化和自助化发展，所以很多券商都在积极的打造面向于互联网客户的自助服务平台，比如集网上营业厅、网上商城于一体的客户终端，通过统一的客户终端，可以实现券商直面客户提供一站式的金融行业服务目标。互联网+金融的客户体验也在不断的提出更高的要求，他们越来越期望证券公司在经纪业务这块能够提供操作傻瓜式、服务便捷式，客户可以通过亲自到柜台的渠道7*24小时在任何地方的营业部办理跨机构业务，也可以通过移动端（手机、VTM、PAD）办理这些业务，所以如果证券公司的软件服务欠佳，就会导致客户的流失。

在线下，券商营业部也在同步转型，营业部在形态上逐渐往轻型化和服务多元化方向发展，传统营业部缺点：人员及硬件配备齐全，占地面积广，成本高；主要收入来源单一，依靠经纪业务；服务模式被动——是一种等客上门模式。新型营业部优点：人员及硬件设备简化，以门店形式铺开，单个营业网点成本低；除传统的经纪业务外，服务网点可增加设立以金融产品服务、投顾服务、财富管理等更为专业化和定制化，能够跨区域给有差异的客户分类提供多样化的服务需求，这样才能体现系统的灵活性；服务模式——主动出击，全员营销。这是在营业部层面的一个转变。

近年来，随着证券公司非现场开户的开放，证券公司客户的地域分布远远超出营业部网点的覆盖范围，因此，证券工作服务的压力也随之而来；而互联网证券公司的兴起，使得证券公司在经纪业务领域进入薄利时代，逼迫证券公司不断的缩减运营；并且随着证券公司新设营业部网点的门槛降低，证券公司营业部网点的数量呈现出爆发式的增长，如何更好的控制风险是券商在当下环境中瓶颈问题，亟待突破解决。在这些客户服务模式和营业部的轻型化转变的情况下，面向于客户的服务提供层面不管是线上还是线下都变轻了，这就意味着有一部分服务内容会被往后移，所以说券商想要通过建立集中作业中心的模式去管理也成为大

势所趋，包括设备和人力等服务资源的集中化，比如通过构建集中作业中心来实现服务资源的整合，为线上和线下服务提供后援的支撑，实现公司的资源共享以及对于营业部、员工和客户统一的公司级的运营管理，达到降低运营成本的目标。在这么一种新的模式下，显然现有的 IT 系统操作终端显然是无法满足的，因此也就产生了对于员工服务平台重新塑造的需求，以及对于客户自助服务终端业务服务的支持等需求，基于以上背景，证券公司提出将证券经纪业务运营模式有分散运营转化为集中运营，以“服务向前、管理集中”为核心，通过“营业部分散受理，总部集中审核”的方式，行业的业务制度重新塑造，公司的内部人力资源重新调整，券商办理业务的流程简化，公司的经营成本逐渐减低，营业部柜员的系统操作简单，提升运营销售渠道和合规风控能力，并通过能够线上化的线下业务尽量线上化来提升客户便捷式服务的体验。JZ 集团作为国内证券公司大型软件供应商之一，为了解决证券公司这一需求也提出了新平台的构建来满足券商以上的这些诉求。于是在 2015 年 J 公司专门成立了证券软件 YYCP 事业部负责该产品的研发工作，J 公司 YYCP 事业部创立以来，研发了一套系统，能够解决券商当下存在的这些问题，提供 7*24 小时支持客户线上、线下办理业务，大大的提升了客户的体验，但是由于金融行业内深交所、上交所、北交所/股转公司、中登公司、中国基金公司、中证公司基于监管的要求，经常会变更一些规则，在产品迭代的过程中经常会出现需求的反复变更，需求不清晰，系统研发过程中沟通受阻以及完善各种文档等多种问题，导致产品在整个周期过程中进度缓慢，产品质量较差，在同行业的竞争中失去了客户的信任和产品快速交付的时间优势，最终导致证券公司客户的流失，降低了公司的盈利。

软件产品需要更快的响应证券公司客户的需求变更以及日益扩大的规模，所以现有的软件项目管理应该更加具备一定的敏捷性。现有的敏捷软件开发项目管理也有一套流程，基本流程包括启动-计划-执行-监控-上线等一系列过程、而项目管理一般也有自己的一套管理目标，使用 project 的甘特图和 JIRA 等管理工具来进行项目的管理。通过使用 project 的甘特图制定项目进度计划，以及人力资源分配进行进度的把控和结果评价，通过使用 JIRA 来管理需求、管控版本进度。为了快速响应客户的需求新增和变更在软件开发项目管理中，大部分公司的系统建设仍旧使用的是通过需求、设计、开发、测试和实施运维组成的瀑布模型，他们从上而下衔接非常紧密，相辅相成。现阶段，J 公司在软件研发过程中使用的软件项目管理虽然表面上叫敏捷开发，实际上还是沿用了瀑布模型，跟真正的敏捷开发存在一定的差距。随着目前项目需求的增多，使用该模型随之也带来了诸多问题，比如，某个重要的环节一旦出现问题必定会影响跟它相关的其他环节，延迟项目进度，所以，针对这些需求变更频繁、新需求越来越多、工期紧张的项

目，其软件项目管理已经不适合使用现有的模型，亟需一套适合该项目组的方案来解决软件项目管理问题。为了解决这一问题，敏捷软件开发在这方面表现出来其强大的优势。

1.1.2 研究意义

为了应对软件项目开发过程中经常出现需求变更的现象，需要一套适合项目本身特性的项目管理方法来提升产品质量。敏捷开发的软件项目管理主要强调的是项目团队成员之间的沟通，以及使用高质量的代替各种复杂的文档，并且能够针对需求频繁变更和需求新增进行调整，制定新的项目计划。敏捷开发涵盖了传统的迭代开发、增量开发，是一种以客户价值作为发展方向，积极高效地应对客户需求变化的业务环境。因此，需要深层次的研究应用敏捷开发的软件项目管理，具体应从以下几点考虑：

(1) 可以使公司建立一套更完整的敏捷开发项目管理模型，供 JZ 公司其他项目组借鉴；

(2) 可以使该公司的项目管理流程规范达到敏捷项目管理要求、通过每项细化的流程来提升；

(3) 在确保软件产品按时保质保量发放的同时持续提升生产效率，提高了交付产品的价值，同时也让公司不会因为延期交付产品而产生违约金。

1.2 国内外研究现状

1.2.1 国外研究综述

国外对项目管理研究时间比较长，最早起源于美国，软件工程是一门研究软件项目管理的计算机科学分支，其在19世纪六十年代软件危机的背景下被提出^[1]，全世界最早盛行的是成功应用于美国研制原子弹的曼哈顿和阿波罗登月计划。德国和日本将这种思维方式应用到武器开发项目中，这也就是项目管理的雏形^[2]。到上个世纪80年代得到广泛应用，随着科技发展，产生新技术，从最初的瀑布模式、增量模型开发项目管理到现在被广泛应用的敏捷开发项目管理。

上世纪60年代，1968年,NATO(North Atlantic Treaty Organization)举办了第一届软件工程学术会议，指出“软件开发应该是类似工程的活动”。因此，软件工程领域提出系列理论、方法、技术和工具，在实践中得到成功应用^[3]。上世纪70年代提出软件项目管理，DOD(U. S. Department of Defense)对当前软件研发产品不能按照预期时间交付，成本超过预算以及产品质量差，不能满足客户需求等问题展开探索。研究表明，70%的软件开发项目未成功是因为项目管理不善引起的，技术原因带来的影响很小。以至于软件开发者渐渐对各项软件项目管理工作引起

重视。研究发现,影响软件项目风险的最主要原因是开发过程是否完整,同时使用软件工程和项目管理方法可以克服软件危机。20世纪90年代中,仍未解决软件研发项目管理不善问题。美国软件工程实施现状发现,无法预估软件项目管理将要发生的事情,大约只有10%的项目能够在预定的费用和进度下交付。随着我国软件行业快速发展,软件开发得到重视并成为热门^[4]。

20世纪70年代初期,温斯顿·罗伊斯提出了经典“瀑布模型”;相关学者对“瀑布模型”进行很多深入研究与应用^[5]。经过长期的反复试验中,又陆续出现了一些其他的经典模型,比如增量模型等等。

瀑布模型的需求需要先梳理清晰,必须要用户参与其中,评审工作也是各环节必不可少的,且各个阶段存在因果关系,适用于需求不易变更和完整的软件系统;快速原型模型则是建议客户参与,需求可以不预先确定好,可以在后续一步一步的增量补充,根据客户需求的变更做出相应的调整,适用于需求比较复杂又难以确定需要逐步变更完善的系统^[6];增量模型允许开发人员并行一个模块一个模块的开发,适用于用户需求稳定不会变更但是技术风险较大的软件系统;部分迭代模型也是分多个阶段增量开发系统,逐渐完善用户需求开发软件产品,适用于需求难以确定且变更频繁的系统。相对于瀑布模型而言,部分迭代开发模型能够让开发人员逐步把软件模块提交给客户,有效的降低了瀑布模型的因返工而延期的项目风险。除此之外,软件项目管理还用到了其他工程技术领域的关键链、PERT(网络分析法)、WBS(工作分解结构)等技术。

20世纪后,移动互联网技术的快速发展,客户的需求越来越多,也越来越频繁变更,软件系统更加需要适应客户的个性化的变化,传统的项目管理已经跟不上需求频繁变更的步伐。因此,2001年《敏捷软件开发宣言》的产生使敏捷开发在全球成为当下最主流的软件项目管理方法^[7],PMI(Project Management Institute)研究结果表明,高绩效组织经常采用敏捷项目管理策略,以帮助组织在复杂而充满变化的商业环境中取得成功^[8],是一套不同于复杂软件开发流程的轻量级体系(如经典瀑布模型)^[9],因此XP Srcum、迭代式增量软件开发过程等模型的产生得到大量软件项目管理的高度赞同。敏捷方法有4个观点:个体和互动高于流程和工具;工作的软件高于详尽的文档;客户合作高于合同谈判;响应变化高于遵循计划^[10]。经过长期实践,敏捷开发已从最开始的比较泛的模型渐渐得到了细化,开发方法不断发展,广泛应用于实践工作中。无数的案例研究表明,很多国内外专家学者为了降低延迟交付风险,研究出敏捷开发优化的调度算法。国外专家学者在最初软件产品线中引入以客户为中心的敏捷开发,将敏捷开发、设计思维和自组织团队成功结合,在构建客户价值的响应性和准确性方面提供了

显著的改进^[11]，敏捷软件开发项目管理模式因此得到了部分软件行业巨头的重视，其中最具代表的有 IBM、Google 等。

截止目前为止，ISO9000特意为软件行业定制的ISO9004标准，制定衡量软件质量的准则^[12]。一些研究人员比较了具有安全工程流程的敏捷流程，在过去的几年里，大部分软件公司已经从僵化的瀑布模型转向更灵活的敏捷方法^[13]。

1.2.2 国内研究综述

华罗庚说过：“我们的企业两条腿走路，一个是科学技术，一个是项目管理”^[14]。相对于其他国家而言，软件项目管理在我国开始的比较迟，这几年国内趋势很强势，实际实施操作过程中会遇到各种各样的问题^[15]。从上世纪诞生计算机以来，许多软件厂商就开始研发相关软件项目，但是许多软件管理缺仍然处于实践状态下，众多软件被返工重新制作，甚至还被当成废品^[16]。1992 年到2005 年，国内关于软件项目管理没有具体的研究实施，与其他国家相比已大幅度处于落后状态，2005 年，国内才开始申请相关软件项目的专利，经过短期的学习和积累，发展趋势一直很强势，因此，国内的软件项目一直都很兴盛，整个国家针对软件项目的创新技术也都在重点关注中^[17]。

2009年，陈婷婷在一篇论文中指出，项目经理对项目的成功有着至关重要的作用，不仅需要掌握本专业的各项技能，也需要懂得项目管理相关知识，尤为关键的是要管理好团队中人员的分配使用，挖掘团队成员的潜能，使其资源利用最大化。

2013 年，闫帅、许鹏翔分析了瀑布模型与敏捷开发模型的特点，证明在实际案例中将两种模型的优势相结合是一种很好的软件项目管理方法^[18]。

2015 年，在赵丽等人的研究成果中强调，企业积极主动的探索新的项目管理方法，按照企业实际情况优化现有流程，成功的提高了软件公司在市场中的占有份额。

2018 年，田莉指出，敏捷开发需要结合实际问题，具体项目具体分析，不能盲目的生搬硬套，才能达到软件项目管理效果^[19]。

20世纪初期，专家们看到开发团队陷入沉重软件过程中，总结出了敏捷软件开发方法，它能够让项目人员可以提高工作效率、积极适应需求变更。后续，又在其基础上衍生出来了一系列的项目管理体系和方法^[20]。

虽然实施敏捷开发的企业很多，但其中有一大部分企业并没有真正理解其内涵与核心价值^[21]。敏捷开发的管理方式是轻量级的，可以接受需求频繁变更和周期短的软件项目。敏捷开发团队中的每个成员都很重要，它是通过多次迭代形式的软件开发方法。它先将复杂的软件系统分解为多个子系统，在每个子系统都完

成开发和测试后,再对整个软件系统进行整合和集成,运用这种方法进行软件开发不仅降低了开发工作的复杂度,而且由于采用模块化开发,提升了模块重用度,缩短了研发周期达到成本的节约。目前,只有一些大型企业才应用了敏捷软件项目管理,例如华为、腾讯等,他们通过实际应用发现问题并持续完善。项目管理中重要的风险管理同这种方法一起结合使用,能够使得软件项目管理中的风险减少^[22]。

1.2.3 文献评述

软件产品的成功其关键因素就是要有完善的软件项目管理流程,国内在这方面虽然起步较晚,但是通过不断的追赶发展,已经同其他发达国家软件项目管理的水平差距逐渐减少,相信不久的将来凭借国内软件企业在实践中学习归纳总结经验应该很快就能超越发展。敏捷软件开发项目管理作为一种新型的项目管理模式,简化了以往其他项目管理模式的繁琐流程和文档,更注重项目内部成员间的沟通以及交流。常用的软件开发过程有增量迭代式,它是基于 Scrum 框架研发的,即使需求经常发生变更,仍然坚持简单、持续集成、不断交付、价值优先等原则。敏捷项目管理都是以工作量来划分,在进行每项任务时,都要经过详细分析后再评估时间,可以详细的拆分成按天、小时、分计算,当工作量大于 8 小时可以再拆分成多个子任务进行。敏捷开发是以多个迭代增量开发的,除了预估每个任务的时间外,也会预留相应的时间来能够解决未知的风险。项目过程中团队成员必须要通过经常沟通来完成任务,沟通方法和技巧就显得尤为重要,项目结项后项目团队成员也会组织经验分享会议,这样更加有利于团队的成长,提高团队的整体业务和技术水平。

敏捷软件开发项目管理需要有敏捷的思维,尽可能的满足客户提出的业务需求,能够适应并较快的解决需求实现过程中客户提出的业务需求变更请求,在确保软件产品按时保质保量发放的同时持续提升生产力与生产效益,提升交付的价值。

1.3 研究思路和研究方法

1.3.1 研究思路

此论文属于实践型案例的研究,主要以J 公司敏捷开发的软件项目管理为例通过对J公司的现有的项目案例进行分析,找到影响项目进度、成本、质量的原因,并结合现有的符合公司情况的项目管理方法提出适合的优化方案,建立一套完整的敏捷开发软件项目管理流程,并针对这个流程能够顺利执行提出对应的保障措施,使得新的优化流程可以在项目中应用并达到预期的应用效果,实际解决

公司当前敏捷开发软件项目管理中的问题,也可以提供给公司或集团其他项目组学习和借鉴这套项目管理方法。

1.3.2 研究方法

本论文主要想从文献资料、调查问卷、案例分析这三种研究方法着手研究。

文献资料:通过了解敏捷软件项目管理的发展历程,通过的阅读相关书籍和资料来积累敏捷软件项目管理的理论基础,通过大量阅读国内外敏捷软件项目管理和项目管理的相关文献资料来积累文献并将其进行分类管理。在已有最新研究成果的基础上,获取敏捷项目管理的最新知识,对其进行分析,寻找适用于J公司的敏捷项目管理方案;

调查问卷:根据J公司的目前实际的项目管理现状,设计一套相关的问卷,发放给集团公司使用敏捷软件项目管理组的成员根据自身了解到的情况填写,最后,收集归纳分析问卷结果并得出结论;

案例分析:运用案例研究分析法,以J公司现有的敏捷软件开发项目管理作为研究对象,进行实际调研分析,收集资料和相关数据,详细记录案例的进展,并得出相关的总结报告,更能有效的为企业提供有效的项目管理方法。

1.4 研究内容和研究框架

1.4.1 研究内容

本论文共分为6个章节,主要研究内容和逻辑关系如下:

第一章:绪论。此章节主要阐述研究背景和研究意义,国内外对敏捷软件以及软件项目管理的现状、思路和方法进行分析,因此结合实际针对第三章的J公司项目管理现状与问题分析。

第二章:相关理论基础介绍。此章节主要介绍项目管理概念、软件项目管理过程,敏捷软件开发项目管理模式的介绍、方法、流程、优缺点等,以及其他项目管理模式与敏捷开发模式进行对比。

第三章:J公司项目管理现状与问题分析。此章节先介绍J公司的概况以及软件项目管理模式,然后对软件项目管理现状进行分析,为后续问题的研究提供理论支撑。

第四章:基于敏捷开发J公司项目管理优化方案设计。此章节主要针对J公司的软件项目管理中存在的问题设计优化方案,分别从组织结构、沟通、成本、质量、进度方面提出适合J公司的优化对策。

第五章:J公司敏捷开发项目管理优化方案实施过程和保障措施。此章节主要通过优化方案在J公司软件项目的应用过程,为了能够让优化方案顺利执行针

对性的提出了适合的保障措施。

第六章：总结与展望。此章节对该论文进行归纳总结，以及分析不足的地方。通过对未来的展望，针对文中提出的问题进行成功的解决。

1.4.2 研究框架

根据本论文对J公司敏捷软件项目管理现状以及问题的研究，分析当前项目管理模式的缺陷，并针对这些实际缺陷详细分析，最终找到困扰项目管理缺陷的合适方案，制定相应的保障措施让其能在实际项目中实施，最终得出结论与展望。因此，论文框架如下图1-1所示：

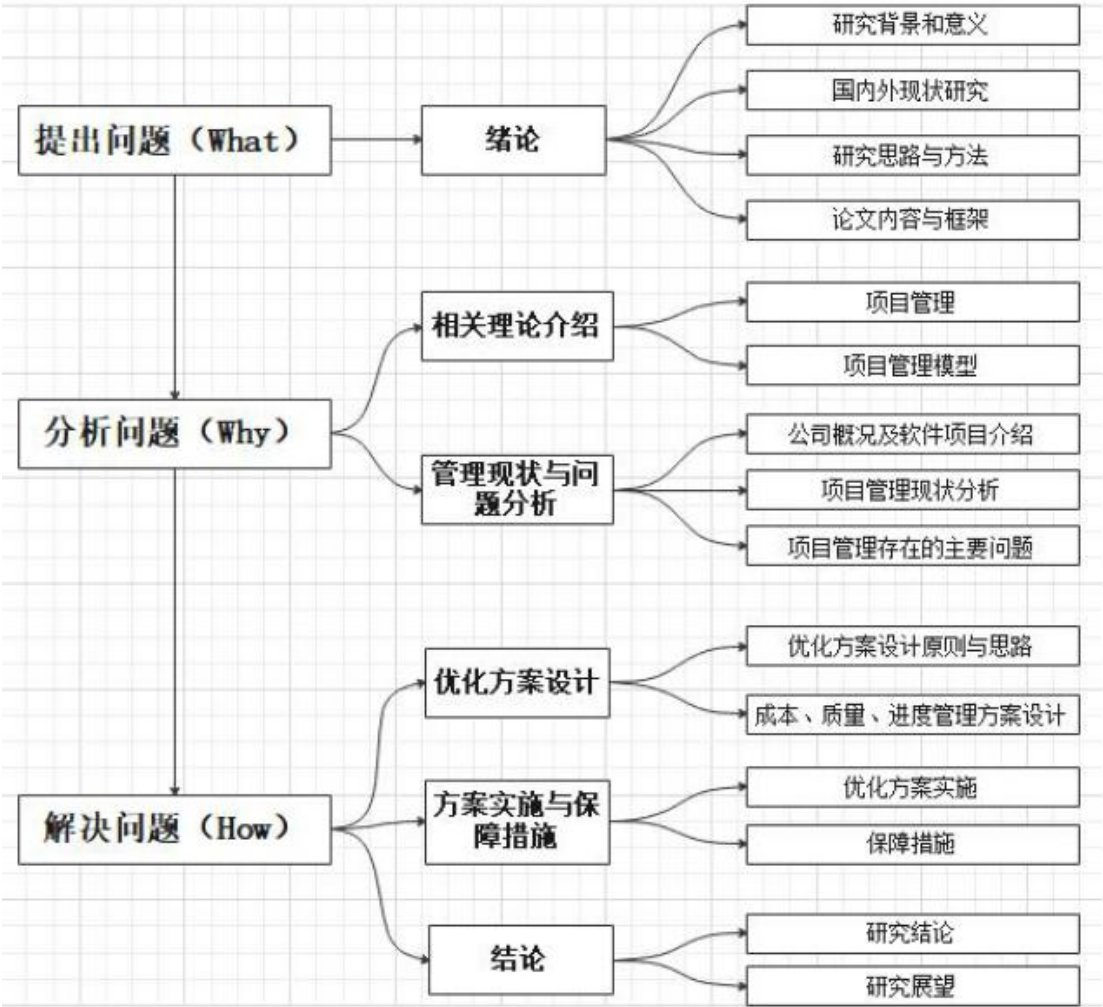


图 1-1 论文研究框架

第 2 章 相关理论基础

2.1 项目管理

2.1.1 项目管理概念

项目是一个有时间和资源要求,并且循序渐进的独一无二的任务。项目的目的就是为了解决可交付的工作成果。为了解决项目复杂多变的风险产生了项目管理学科,软件项目管理就是在其过程中为了解决软件缺陷通过将学习研究成果转变成使用工具的能力^[23]。管理学包括很多分支,项目管理就是其中一个,它是指在项目中运用专门的知识、技能、工具和方法,使项目在有限资源条件下,实现或超过设定的需求和期望的过程^[24]。它包括整合、范围、进度、成本、质量、资源、沟通、风险、采购、相关方十大管理知识领域^[25]。是通过监控和管理任务是否能够完成的过程,该过程的活动包括策划、计划和维护。它有一套特殊的不同于其他行业管理的体系和方式,它的实现就是关于“在复杂多变的风险中,处理好一个事”的理念和技术。

项目管理通过持续增量的方式来完善,必须有相关专业人员来进行管理和实施项目,解决实际中遇到的各种问题和阻碍^[26]。项目管理如果要使其在软性项目中产生巨大的作用,那么就需要应用科学的方法去管理,并与实践结合,持续不断的试验、优化和完善,在实际的实施过程中找到适合公司管理项目研发团队的方法。

2.1.2 项目管理过程

项目管理的过程大致可以分为开启、规划、实施、验收、维护这 5 个阶段,它们之间相互影响,每个阶段都缺一不可,都有规定自己的时间范围,一个阶段必须审核完成下一阶段开启,各阶段完成后也会产出对应的文档,为下一阶段提供帮助。

(1) 项目的开启

项目的开启是一个项目管理过程的首要阶段,意味着一个新项目的开始。项目启动前需要销售跟客户签订相关的合同,预付定金,合同签订后开启项目启动会,启动会需要需求人员、项目经理、产品经理了解这个项目现有的业务需求和未来的方向,并针对其做需求分析,总结出客户的真实想法和该想法是否真实可行,如果项目可行,就会产出软件需求文档,签定合同,随后会组织项目启动会,项目经理、PM(Product Manager)、需求、开发、测试、实施等都必须参与项目启动会,启动会结束后就会进行项目的规划。

（2）项目的规划

项目的规划是项目管理中最为繁琐的阶段，项目经理需要掌握多个领域的技术知识，有一定的实施经验。做好规划主要是为其他的阶段做一份更明确的计划，以便于后续阶段工作能够正常有序的开展。项目成员如果按照制定的项目规划执行可以减少成本的支出，获取一定的收益。项目规划文档包含目的和范围、背景、术语和定义、参考资料、技术方案。其规划一般可分为需求设计、开发、测试以及工程实施类的规划，每项规划都会将项目需求分解，制定阶段性任务以及每项任务的人力、成本、进度规划，并规划不同阶段产出不同的文档。

（3）项目的实施

项目的实施是针对项目规划阶段制定的项目需求、开发、测试、工程实施任务的实际操作，如果要完成这些工作就需要投入大量的人和时间成本。从需求到开发到测试才会进入到具体的工程实施阶段，为了能够让项目具体实施，还需要进行以下几项步骤：

首先，项目的实施需要确定需求、开发、测试、工程负责人熟知各团队成员的技能，根据每个人的能力分配具体的任务以及具体的完成时间，一般会使用 project 进行，最后产出更加详细的任务分解计划表。

其次，客户将需求提交给需求人员后，需要重新梳理，并同客户再次确认细节后产出需求分析文档，需求人员需要给开发人员、测试人员详细讲解需求，此过程中开发、测试可以针对需求不合理的提出自己的建议并要求需求人员更改；需求分析完成后开发人员针对需求产出自己的设计文档、测试人员针对需求产出自己的测试用例，开发设计文档提供给测试人员和需求人员评审，此过程可以让需求人员了解开发人员的设计是否符合要求，让测试人员了解开发的逻辑思路便于后续用例编写以及验证自己用例设计思路是否跟开发一致；测试用例产出后需要需求人员、开发人员一起评审，评审过程中可发现需求是否考虑周全、开发人员的开发逻辑是否与测试一致；需求开发、测试完成后需要工程实施人员进行生产部署，并产出相应的部署文档，供客户参考，也为以后其他项目组部署该项目提供方便。

最后，各阶段的执行过程中需要控制项目人力资源、沟通、进度、成本、质量的风险，把控需求、开发、测试进度，快速响应需求变更、严格按照项目规范控制开发、测试质量，确保能够按照计划执行，顺利交付产品。

（4）项目的验收

项目的验收意味着开发、测试工作的完成，工程实施人员协助客户按照他们提供的架构部署文档部署生产环境，部署成功后安排时间通知营业部在该环境进行测试，确认产品的功能是否与原始需求一致，测试通过后方可验收，验收通过

后需要签定对应的验收合同，销售根据验收合同督促客户结算尾款。

项目验收后开发、测试人员需参与项目回归会议，总结项目中的问题以及优点，针对问题提出解决方案并严格执行，针对优点需要继续保持。

（5）项目的维护

项目的维护是指项目验收后，产品已经投入生产使用后出现系统功能报错、环境部署不合理等问题，为了确保产品投产后不出问题，一定时间要对其进行升级或者清理一些系统日志或缓存等。此过程可针对解决的问题产出问题维护清单，记录问题并详细记录解决方法，方便后续出现同类问题可以快速解决。维护期一般没有固定的时间，维护期的长短依赖于该产品运行生命周期，若被其他产品替代后，那么也就意味着维护期的结束。

2.2 敏捷软件开发

2.2.1 敏捷软件开发方法

敏捷含义是指动作或者思维反应迅速快捷，能够非常灵活。敏捷开发它由传统瀑布、增量迭代等衍生出的一种新模式，强调项目的质量以及需求频繁变更的应对能力。敏捷开发项目管理主要是用于规范产品开发管理的一个过程。敏捷项目管理方法的重点还在于项目管理，需要清楚项目的具体操作以及过程，然后针对项目特殊性进行考虑，其重点要素在于强调“敏捷”或者“必须敏捷”，是比较复杂的^[27]。敏捷项目管理在迭代中与客户保持正常的沟通联系，获取他们的产品功能详细的实现意愿，这样才能达到要求提升质量。敏捷理念基本原则减少了不能向客户提供价值的活动^[28]。敏捷开发注重的是产品研发过程中频繁变更需求的产品质量以及客户对交付产品的满意度。敏捷开发各成员间沟通合作并能够适应并保质保量的快速处理频繁变更的需求是其主要宗旨，它是以需求为导向，将项目分解成多个小项目，使每个小项目的需求能够循序渐进执行，在实际操作中锲而不舍的持续增量改进项目管理方法。敏捷软件开发方法是当前使用最流行的软件项目管理方法，目前很多大型企业都在使用敏捷项目管理方法，它具有简洁性、灵活性，强调该做什么不该做什么，人尽其用，可以减少团队的支出，增强成员间的表达、倾听能力，提高了软件的质量。许多使用敏捷软件开发项目管理的都会使用 Scrum Board 和 Kanban，他们都是以成功为目标，学习和总结敏捷思维，统一达成敏捷价值观。

上世纪 80 年代后期，Scrum Board 开始流行，它将团队的工作量分解成更小的事故（story），事故又可以分级成更小的子任务，然后将这些事故或子任务分成多个迭代并确定优先级，优先处理优先级高的迭代，每个迭代预计 2 周时间，优先级高的迭代确认后让项目团队成员评估自己的工作量，然后一个迭代一个迭

代的完成任务，每个迭代需要严格填写实际的工作量，当一个迭代完成后，可以根据评估工作量和实际工作量进行对比，为下一迭代评估工作量做好样本，缩小后续迭代实际工作量与评估工作量之间的差距。

自 2004 年起，丰田工程师创建了 Kanban 方法，Kanban 在开发团队流行起来，与 Scrum Board 相比，Kanban 就没有显得那么规范，允许项目有变化并且能够快速响应需求的变化，为了不浪费资源，提供了一套透明的持续的工作流程，将工作细分成任务，可以根据细分的任务通过状态区分，便于看出已登记、已受理、已分配、开发中、待测试、测试中、已解决、已关闭状态下工作任务的数量和运行状态，确保每个人的任务都在合理范围内。每一个任务只能处于一个状态中，减少了团队之间转开发、转测试、转发布之间的沟通问题，降低了团队的沟通成本。此方法也便于负责人针对每个人的工作量进行预估，制定相应的计划，合理的分配每个迭代的团队人员的工作任务。

通过对敏捷 Scrum 看板、Kanban 的特点对比，如下表 2-1 所示：

表 2-1 Scrum 看板与 Kanban 的比较		
	Scrum Board	Kanban
任务大小	任务细分	任务细分
图表	使用燃尽图	无图标，使用发布状态区分
工作周期	以迭代为周期，每个迭代 2 周时间	以任务为周期，计算每个任务完成的时间
需求变更	不允许需求新增、变更	在时间、资源允许的条件下允许需求变更、新增
优先级	每个迭代需优先处理级别高的需求	未设定优先级
适用团队	适合成熟一点的产品和团队	成员需要能力强，敏锐性强，适合新开的产品

通过上述方法对比，可以看出 Scrum Board 与 Kanban 方法各有优缺点，J 公司作为金融软件供应商，为了适应需求快速变更，J 公司的敏捷软件开发项目管理优化需要同时结合他们的优点使用。

2.2.2 敏捷软件开发介绍

敏捷方法是一种以人为导向而非过程导向^[29]，以人为核心的，持续增量改进模式，各成员间相互分工配合能够快速发布产品，抢占市场。它能够适应需求紧急变更，强调的是人与人之间的交流，灵活的调动了团建成员之间的相互合作、相互沟通，更能注重团队成员间的作用。敏捷开发就是通过实践来发现问题，并解决问题。以此形成一套强有力成功援助提升软件项目管理的系统^[30]。敏捷开发主要包括 XP、Scrum 主流框架。其中，Scrum 的简便性、高效性、可复制性被人们广泛应用的敏捷项目管理方法^[31]。

敏捷项目管理与敏捷开发联系紧密，敏捷开发发展至今能够更加具体和准确，因此成为项目管理的标准化叫法^[32]。敏捷软件开发最重要的是能够快速的满足客户的需求，同时也要快速灵活的接受客户的需求变化，它是一种可持续交付开发的方法，持续时间最少不能少于 2 周，最长不能超过 4 周，持续周期尽量不要太长，项目的需求、开发、测试人员应该都在整个团队中工作，为了更好、更方便沟通，团队中的成员应该坐在一起，或者相隔距离较近，尽可能的减少不必要的沟通，简单明了的描述问题，每个迭代结束后，项目组的成员应该组织一次回顾会议，总结团队的问题，针对问题提出相应的解决方案，针对好的规范需要继续保持，用于提升团队的工作效率。

敏捷开发项目管理有合理的项目治理流程，以及全面的计划，与客户保持顺畅联系努力满足客户产品需求实现的意愿，调动人员积极性，高质高量完成任务^[33]。基于敏捷开发方法的项目管理更适合应用于项目周期短、沟通无障碍、需求不明确的项目，但是也存在文档不全面、团队人员能力要求较高的缺陷^[34]。

敏捷开发应该遵从四条基本原则^[35]：持续增进、减少非必要任务、相互配合、实事求是。

2.2.3 敏捷软件开发项目管理流程

2.2.3.1 敏捷通用过程模型

敏捷开发是将很多需求按照紧急程度优先级分成多个迭代，每个迭代有包含计划、开发、验收、回顾 4 大过程，其中迭代计划可通过 PB（产品待办，一个最终会交付给客户的，根据商业价值来排列优先级的产品待办列表，项目的列表为 JIRA 中的需求）查看，迭代开发则可通过 SB（spring backlog，迭代任务列表，从 PB 中挑选高优先级需求，SB 中的需求可被细化为更小的子任务，子任务工作量建议小于 16 小时）查看，详细明细见下图 2-1：

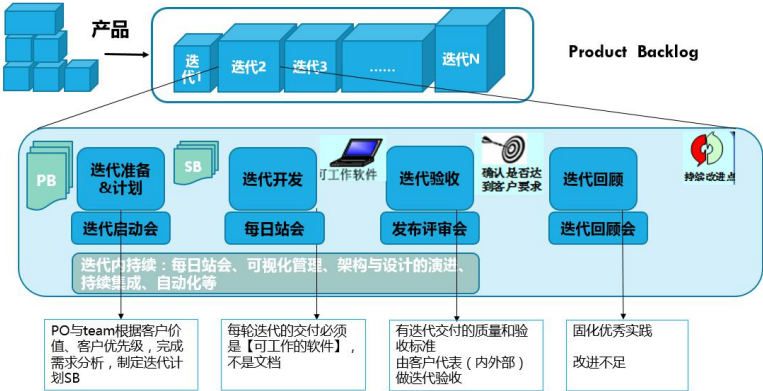


图 2-1 敏捷通用过程模型

2.2.3.2 敏捷过程管理模型

敏捷过程管理分为每个迭代的项目启动会前、项目启动、项目执行、项目回归及回顾，团队成员参加所有迭代，且优先级高的迭代结束紧接着就是次一迭代的开始，其中PO（Product Owner，产品负责人）、master（敏捷教练）、团队成员在各个阶段都会参与其中，详细明细见下图2-2：

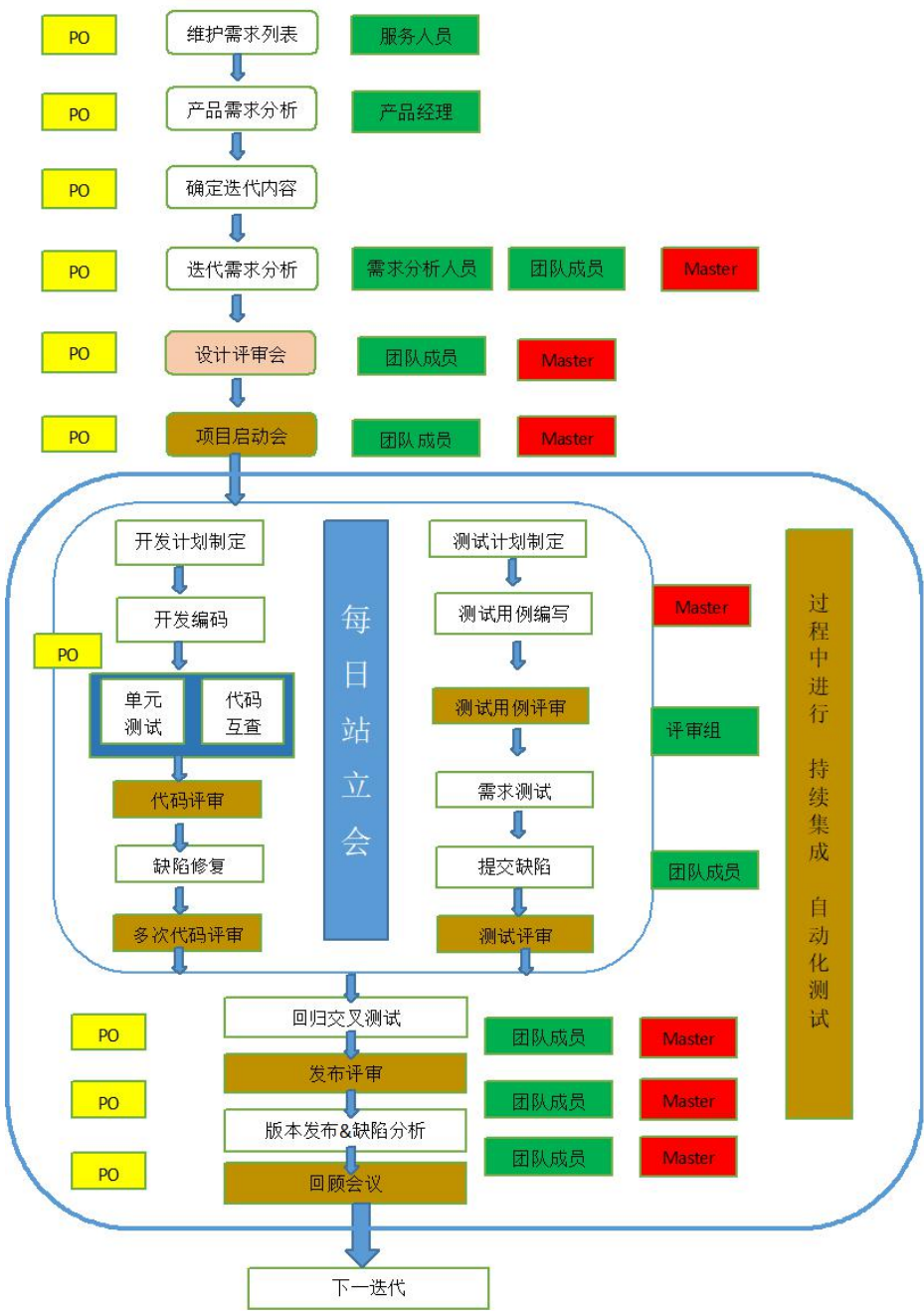


图2-2 敏捷过程管理模型

2.2.4 敏捷软件开发优点和缺点

敏捷软件开发的优点在每个迭代的交付过程中相对于其他的项目管理模型来说有一定的优势：

(1) 可以通过迭代交付：敏捷的目的主要是提高客户的满意度，敏捷项目管理是通过持续迭代完成的，迭代完成后会完成相对应的需求功能提交给客户部署，客户部署测试的过程中，可以针对这些需求的不完善提出更好的建议，并快速处理和解决；

(2) 能够应对频繁的需求变更：客户比较关注最终的产品是否能够满足其要求，在大量的真实软件项目管理过程中，需求变更在实际项目中是经常出现的，特别是需求不明确的，后续需求变更就尤为突出，因为敏捷软件项目管理是通过持续迭代完成的，迭代范围应该固定，变更某个需求只是针对当前迭代产生影响，其他未开始的迭代并不会受影响，所以就能够随时应对这些变化；

(3) 沟通能力的提升：敏捷开发强调的是当面沟通，敏捷团队的成员都会在一个办公区，当面沟通就会避免因为距离远浪费时间，也可以避免因为通过聊天工具讨论造成的一些沟通不畅；每个产品在实践中团队人员都是需要相互磨合适应，接触多了就会慢慢熟悉，也提高了个人的沟通能力；

(4) 易于管理：敏捷团队人数最合理的安排是不能超过 10 个人，人数太多一般就不便于管理，沟通也会产生分歧，因此团队人数较少能够更加便于管理，也能更加灵活。

敏捷软件开发的缺点相对于瀑布模型、增量开发模型等等就会突出的更加明显：

(1) 文档相对简单：敏捷开发重点培养团队间近距离沟通、口头讨论，并未形成完整的文档，因此就会造成后续维护困难，找不到原始的需求记录，不清楚当初需求设计的目的和背景，也就不知道该功能的作用，需要重新花时间去熟悉，如果文档较为详细，那么就可以通过文档很快知道该功能的作用，也可以将此文档提供给其他项目组需要熟悉该功能的人员使用，以及提供给新人学习新业务知识；

(2) 人员要求较高：敏捷开发的项目管理人员需要有较强的项目经验，针对大型的项目，经验丰富的人能够坚定的做出正确的选择，少走弯路，朝目标前进；与此同时团队成员也要有项目经验，如果都是新人，那么没有老员工带领，新人缺乏项目经验以及业务知识，沟通困难，导致很难把控项目质量，不利于迭代的进行。

2.3 软件项目管理模型

2.3.1 瀑布模型

1970 年，对软件项目管理的推动有积极作用的瀑布开发模式被提出^[36]。它属于最早的软件开发模型，为后续的其他模型提供了最基础的架构，它将产品研发过程分为 4 个阶段，分别是需求分析和定义、软件开发、软件测试和软件维护，它们像瀑布一样从上到下连接^[37]。瀑布模型将产品简化成多个分工阶段，现阶段使用此模型进行开发的公司占比较多，其流程图如下图 2-3 所示：

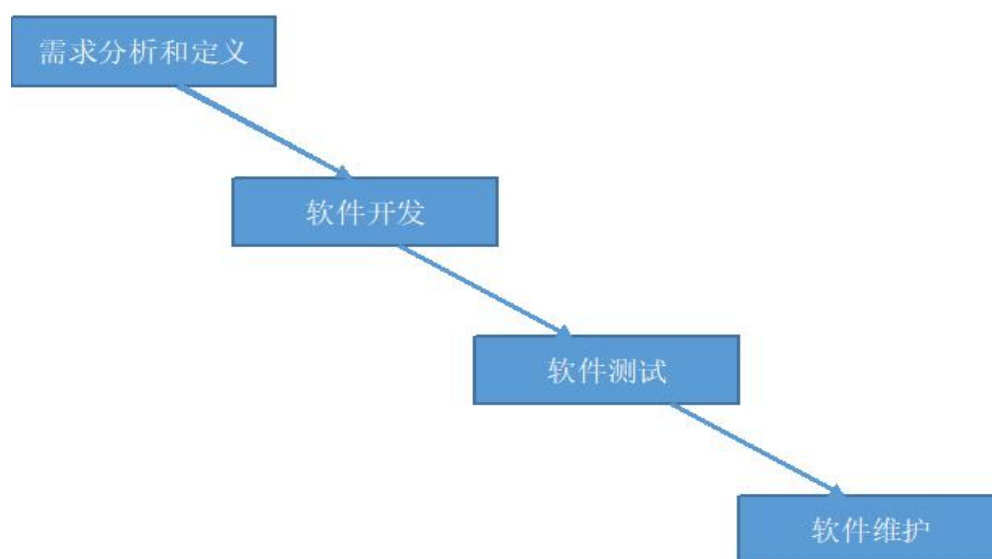


图 2-3 瀑布模型图

瀑布模型的核心是文档，它是各个阶段完成的一个纽带。瀑布模型按阶段划分，只有当前一个阶段完成后，下一个阶段才会启动，每个阶段之间反馈较少，如果某个阶段因为需求变更或需求分析不完善，那么必将导致该阶段延期，后面阶段的完成时间也会相应推迟，因此导致了整体的计划都会推迟，项目的结果只能在最后项目结项才能知道，因此，此模型只适用于需求文档更完善并且不会变更的软件系统。长期以来，大多数的软件开发项目都是采用这种模式。

2.3.2 增量开发模型

增量开发模型是基于瀑布模型基础上提升，该模型随着时间的发展，就会产生越来越多的增量功能，增量模型的第一个模型都是优先级别较高，属于核心的架构，后面的模型都会依照第一个模型的特性研发，一直到最后完成整个产品并发布使用。增量模型中某个阶段出现问题也只会影响该阶段的进度，基本上不会影响到其他的进度。其模型图大致如下图 2-4 所示：

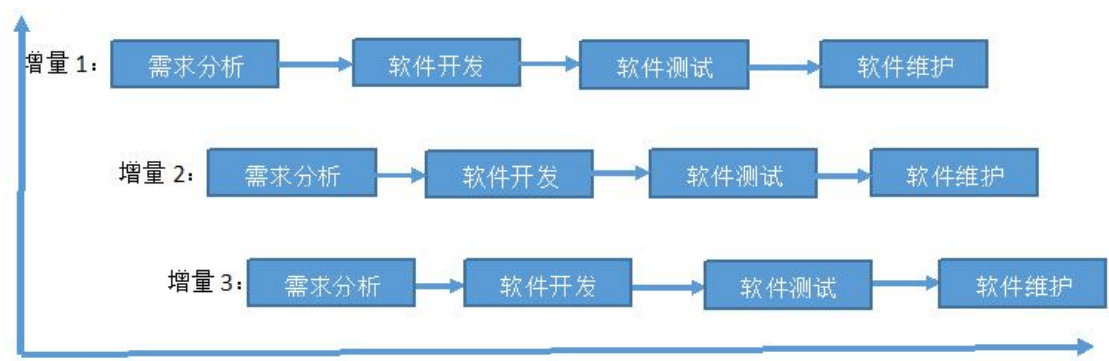


图 2-4 增量开发模型

增量开发模型可以随意分配不同增量的成员，项目初期可以不用全部投入到项目中，可以后续陆续增加项目成员，其功能也是在每个增量包中逐步完善。但是由于在软件开发过程中，经常会出现需求的变更，增量开发模型不能快速的应对其变化，因此，该模型只能适用于技术风险大，无完善需求文档的项目组。

2.3.3 各模型之间分析比较

综合上面几种模型的分析，现对这几种模型的技术特性、优缺点以及使用范围进行了综合的比较，详细明细如下表 2-2 所示：

表 2-2 瀑布、增量开发、敏捷模型之间分析比较

模型名称	技术特点	优点	缺点	使用范围
瀑布模型	1、各阶段很简单，但是又存在因果关系； 2、每个阶段后都有评审和反馈，需求需先确定，用户无需加入	明确划分不同阶段，便于产品更好的针对不同阶段进行不同的管理；	1、上一阶段必须完成后才能进行下一节点； 2、要求用户需求完善，不允许变更；	需求文档更完善并且不会变更的软件系统
增量开发模型	1、软件产品是被增量式的一个模块一个模块开发； 2、允许并行开发；	1、前期投入人力可以不多，且不同增量间人员可以灵活调动； 2、可以优先发布核心框架的东西	1、如果存在交互的增量包有问题，那么所有的增量包都将会前功尽弃； 2、不允许需求变更	技术风险大，需求文档不完善的项目组
敏捷开发方法	1、将系统分成不同的迭代进行，逐步完善产品	1、流程和项目计划合理化，提高了项目组成员的效率性和积极性； 2、客户间经常保持联系，获取客户需求的最真实意愿	1、文档不够全面； 2、团队成员要求高	需求频繁变更，项目风险较高

第 3 章 J 公司项目管理现状与问题分析

3.1 J 公司概况以及软件项目介绍

3.1.1 公司概况

J 公司是 JZ 集团公司的总部，JZ 集团前身就是 J 公司，是国内领军的金融科技全领域服务商，创立于二十世纪末期，截止目前为止有近 20 家分子公司，于 2003 年在上交所上市。它通过了 CMMI 的最顶级认证，即优化级，因此，具有超过其他企业的质量保障。

JZ 用十余年打造了中国金融证券软件民族品牌，2021 年软件百强榜单排名较前，比竞争对手高出 20 名的差距，业务覆盖了证券、基金等各区域市场以及各类要素交易所等金融机构，其经营模式属于服务型的软件行业，为中国证券软件贡献了一份自己的力量，并集团成功上市十余载，建成 JZ 人自己的大厦。

同时，公司制定了新的战略思想，通过软件行业的技术、金融行业的业务创新以及公司不同平台架构的建设与筹划，建立了一套相对稳定成熟的营销与管理模式，完成了集团的再次快速发展。在基于现有模式下继续前进和创新，使其在金融和计算机软件运营等行业占有市场较大的份额，并得到业内的认可。与此同时，第二代经营模式实现共享业务流量收费也在积极搭建云服务和互联网金融业务的综合运营平台；第三代经营模式通过互联网金融服务收费，参与第三方移动支付、投顾理财等来成为互联网金融的领头羊。公司一贯延续“营销平台公司化，技术产品集团化”的理念，推动平台架构积极发展，促进战略实施严格执行，再造多个 JZ 的不同平台公司。

JZ 将持续推进经营模式的转型升级，推出其专属移动终端，提供智慧城市、证券、基金、理财、众筹、网贷等综合互联网金融行业的服务，成为一流的互联网金融行业服务商。将不同的平台架构公司作为延续发展的根基，力求在全国各地建成多个分/子公司，努力将其打造成业内认可的国内品牌；实现技术架构平台的专业化，底层基础架构平台软件在金融行业大范围上线，得到国家和行业更高的认可。

3.1.2 组织架构

JZ 集团有多个分子公司组成，其中 J 公司属于其总部，J 公司按照其产品不同分成了 JZJY、ZHXT、CXYW、JGYW、YYCP 等部门，是以 YYCP 事业部的实际案例为对象进行探究，其需求方是属于证券公司负责经纪业务的各工作人员，包括其下属的各分支公司营业部。他们结合目前现有经纪业务结构以及目前

的经济形式做了分析，总结后亟需一套新系统解决目前现有的问题，于是将其需求同 J 公司销售以及产品经理商讨，最终达成协议研发一套新系统，J 公司因此成立了 YYCP 事业部，将由该部门对其进行研发。

J 公司经过多年的发展，其组织架构也在不断的调整，它在整个 JZ 集团的架构图如下图 3-1 所示：

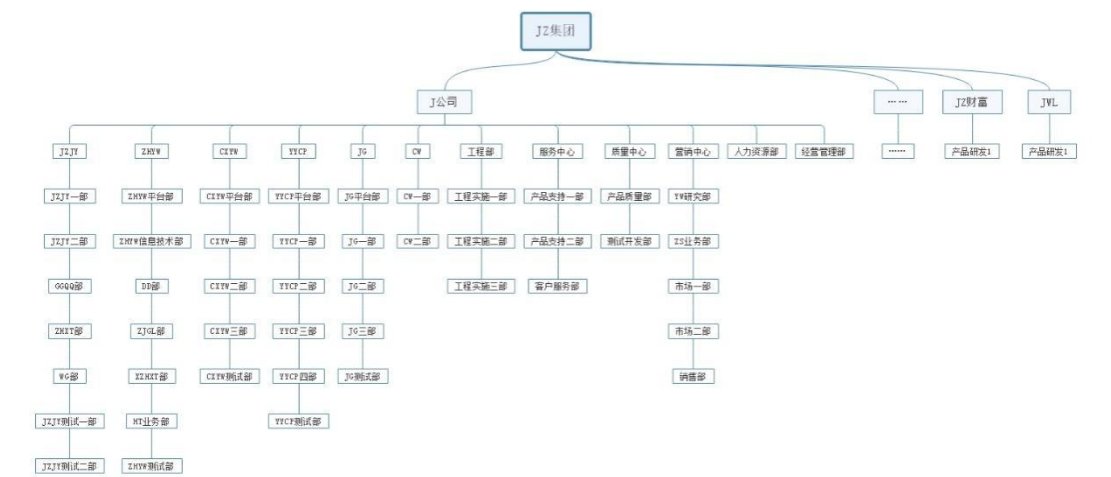


图 3-1 公司组织机构

J 公司成立的 YYCP（即:运营产品事业部）分为 YYCP 平台部、YYCP 一部、YYCP 二部、YYCP 三部、YYCP 四部、YYCP 测试部这 6 个部分，该部门人数已经超过一百多人，每个项目中都包含了项目经理、测试负责人、需求分析人员、项目助理等，都有资深、高级、主管、普通、助理级别的工程师，这些人员合理的分布在各项目中，很多人都是在公司待了很多年，部分堪称元老级人员，熟悉公司的平台架构以及运营，他们参与了很多项目，对公司的业务相对也很熟悉，因此，YYCP 事业部人员能力各方面相对于其他事业部都是比较优秀。

YYCP 平台部负责运营产品事业部的所有产品的架构搭建，维护产品系统架构的正常使用，此外，该部门还包含需求分析人员，需求分析人员负责各个产品的需求分析工作，了解客户的最原始需求，并将其记录成需求文档供产品成员使用；YYCP 一部、YYCP 二部、YYCP 三部、YYCP 四部分布研发和维护各自的产品项目，每个项目中都会配备对应的项目经理负责各项目的计划，同时也有对应的项目助理人员协助项目经理跟踪项目进度、组织项目会议、记录项目问题，各项目之间比较孤立，但是在某些功能模块上又是相辅相成的；YYCP 测试部将部门人员分散在各产品项目中，负责系统的测试工作，并运用功能测试、自动化测试、性能测试等测试方法；另外还会同公司工程部申请工程实施人员，工程实施人员负责产品的上线工作，以及上线后的维护工作，保证产品上线后的正常使用；此外，公司有单独的 QA(质量管理)部门，监督和把控所有产品质量，规范

流程等。

3.1.3 软件项目管理流程

软件项目目的是将需求演变成详细设计，然后将设计软件通过开发代码实现，测试进行测试，最后部署完成上线。软件项目管理就是对这一过程的把控和监督。通过对现有资源的合理分配，控制项目成本，制定相关的需求、开发、测试、实施计划，然后对所有的计划进行监控，保质保量的完成客户的需求，以达到上线目标，上线后软件能正常使用。

J 公司目前 YYCP 项目组敏捷项目管理流程如下图 3-2 所示：

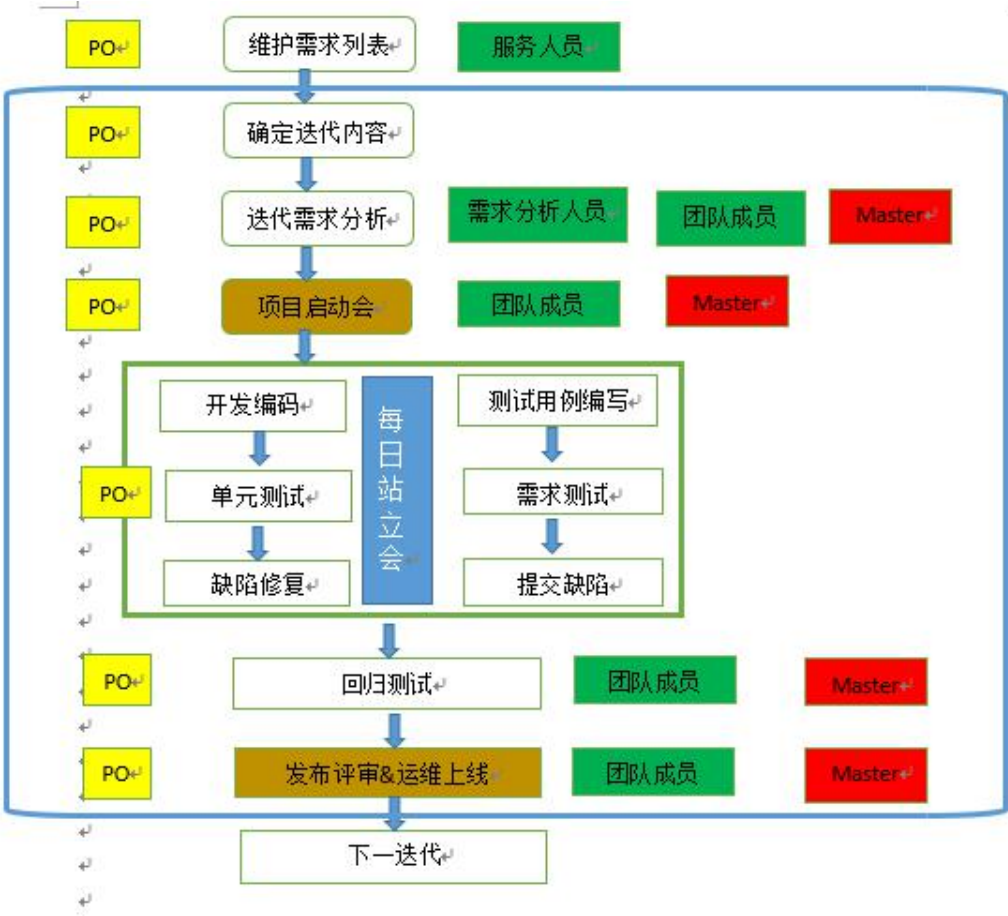


图 3-2 YYCP 项目管理流程

通过上图可以看出，J 公司现有的敏捷模式与上文提到的敏捷模式对比后缺少需求设计评审、用例评审、代码评审、版本总结等部分环节的存在，并针对各个环节的文档也没有做到及时更新完善。

运营产品事业部研发的产品为了解决证券公司经纪业务在互联网化冲击下带来的薄利、缩减运营成本的影响，证券公司提出将证券经纪业务运营模式由分散运营转为集中运营，以“服务向前、管理集中”为核心，通过“营业部分散受理，

总部集中审核”的方式，重塑业务制度，重整内部资源，券商办理业务的流程简化，公司的经营成本逐渐减低，营业部柜员的系统操作简单，提升运营销售渠道和合规风控能力，并通过能够线上化的线下业务尽量线上化来提升客户便捷式服务的体验。同时，由于三所三司需求也不断变更，需求也需要不断的变更，因此，推出新的产品，并且支持需求的多变无疑成了眼下迫在眉睫的事情。在软件的研发过程中，客户同需求、开发、测试人员频繁沟通，针对软件提出自己的意见，完善需求。此外，由于该产品的特殊性，需要对接的关联系统较多，以及需要了解关联系统业务的人员较多，比如集中交易系统、融资融券系统等，对接系统多导致各项目组的人员沟通也会比较麻烦，并且也需要项目组成员需要熟悉对接系统的业务。

3.2 J公司现有敏捷项目管理分析

3.2.1 项目介绍

3.2.1.1 项目背景

近年来，随着证券公司非现场开户的开放，券商客户的地域分布远远超出营业网点的覆盖范围，这给券商客户的后续服务无形增加了压力；而互联网券商的兴起，使得证券公司在经纪业务领域进入薄利时代，逼迫券商不断缩减运营成本；且随着证券公司新设营业网点的门槛降低，券商营业网点的数量呈现几何倍增加，券商迫切需要解决控制风险的问题。基于以上背景，证券公司在2015年提出将证券经纪业务运营模式由分散运营转为集中运营，以“服务向前、管理集中”为核心，通过“营业部分散受理，总部集中审核”的方式，重塑业务制度，重整内部资源，券商办理业务的流程简化，公司的经营成本逐渐减低，营业部柜员的系统操作简单，提升运营销售渠道和合规风控能力，并通过能够线上化的线下业务尽量线上化来提升客户便捷式服务的体验。系统建设是实现该业务目标的基础，经过行业调研，现有的证券业务办理系统缺乏实现上述目标的基础，决定采用“自主开发、共同建设”的方式构建“新一代证券业务办理平台”。

3.2.1.2 项目目标

本项目的主要实施目标为以下几点：

(1) 建设适用于证券行业的基础服务平台，提供流程引擎、规则引擎、原子业务、电子签名、电子印章、无纸化、视频接入、语音合成、图像识别、人脸识别、档案管理等基础服务，并通过数据服务总线实现与现有业务的无缝对接；

(2) 建设支持集中作业中心（即：审核平台），支持全国业务的集中作业审核。

(3)实现业务流程与外部机构服务时间窗口的解耦,实现 7*24 小时全天候、跨地域的业务通办服务;

(4)实现业务开展的流程化、智能化、无纸化、电子化服务,更好的保证合规风控,降低运营成本;

(5)为互联网等外部渠道提供安全、稳定、可快速扩展的接入服务。

3.2.2 J公司现有敏捷软件项目管理

J公司运营产品事业部的项目主要来自于证券公司的需求,该项目立项后,项目将有运营产品事业部全部完成,即按照需求、开发、测试、运维大致几个阶段来进行。需求分析人员需要实时了解客户的需求,积极同客户保持沟通联系,需求变更需求人员没有及时的进行文档的更新以及组内人员的通知,导致最后项目需求文档与实际不符。

目前公司主要使用公司内部开发的测试管理平台、SVN、JIRA 这几个工具对产品的项目管理周期进行管理。SVN 创建项目目录后,主要是用于提交项目的立项文档、需求文档、开发代码、测试用例、操作手册、工程实施部署文档等。测试管理平台主要用于测试人员提交编写的需求测试用例,在需求执行测试用例的过程中提交缺陷,并且及时跟踪缺陷状态;开发人员也可通过此平台针对提交的缺陷进行修复,以及针对缺陷制作图表。JIRA 用于记录三司三所需求,以及记录客户反馈的需求和他们发现的缺陷,项目经理通过该工具将需求进行版本迭代分类,每个版本迭代的需求分配开发、测试人员以及各自工作量的评估,版本迭代启动后可以通过 Scrum Board 和 Kanban 查看开发测试人员的需求,以及每个需求的状态,项目助理可以通过 Scrum Board 和 Kanban 了解项目的大致进度。JIRA 和测试管理平台没有将需求、测试用例、缺陷三者进行关联。

需求开发和测试阶段,由于该产品对接的系统较多,需求分析有时做的不到位,也需要开发和测试去同其他对接产品的同事进行沟通,每个系统没有一个对接负责人,在咨询业务问题随便找一个人咨询,有些人由于工作忙不回复,就会重新咨询其他人,导致时间的浪费;也由于开发人员未做设计文档以及充足的单元测试,提交的需求到测试这边,测试会因为需求设计不合理以及单元测试不充分导致报错退回需求;测试人员由于项目工期比较赶,有时会不写测试用例直接测试,即使写了测试用例也不会进行用例评审,后期项目上线客户要测试用例和开发设计文档,项目组又安排人补充这些文档,需要重新测试一遍然后再补充用例,导致又需要耗费大量的人力资源成本完善这些文档。

现阶段每个产品都有配备项目助理,且项目助理工作任务不饱和,各项目助理间沟通项目问题也不顺畅,还是需要开发、测试等相互去沟通具体问题,项目

组每日站立会规定时间，小组成员总是拖沓迟到，浪费其他人员的等待时间，也每次站立会每人都是通过口头汇报工作进度，并无具体的文档记录每天的进度，项目经理无法准确评估项目的进度。

项目实施阶段的实施人员是通过总部实施部借调人员，项目结项后实施人员会去进行其他项目实施，后续产品维护困难。

基于以上项目管理现状可以看出，项目组在软件项目管理方面做的并不好，因此亟需对现有的敏捷开发项目管理方法进行优化，并制定可行性方案。

3.2.3 设计问卷调查分析影响 J 公司软件项目管理的因素

针对于公司的软件产品来说，只有在不断的优化了软件项目管理方法，以及开发和测试的技术，才能让公司产品在质量方面有所提升，进而减少项目周期尽早部署生产使用，以此来减少竞争对手，最终达到减少项目成本，这才是软件项目管理发展的最核心内容^[38]。

要想针对现有的敏捷开发项目管理进行合理优化，就需要通过有效的方法进行分析，比如问卷调查方法，通过问卷调查的形式更能让人对结果进行简便的分析，但是要想做好问卷也是一项比较复杂的过程，其过程大致分为以下几个方面：

首先，需要明确影响敏捷开发软件项目管理的因素才能明确应该怎么设计问卷，为了了解这些因素，分别对各个项目的不同角色成员使用单独面对面的方法分析汇总，因为他们都是实际参与项目的人员，更加了解各自参与项目中的问题，最终得出结论影响的因素大致分为 5 大类：组织架构、沟通、进度、成本、质量。此外，他们认为可以通过优化现有的敏捷软件项目管理方法，并将这些方法也记录在了问卷中。

其次，根据上述面谈得出的结果，设计了如附录 A 的问卷，对于问题 1、问题 2 和问题 3 设置当参与人未在项目组超过半年、对现有敏捷管理方法没有优化方案以及完全不了解敏捷开发项目管理时，可以直接结束问卷，以免造成最终统计数据对其进行分析的不准确。

再次，根据上述设计好的问卷，将其发放给每个实际参与项目超过半年的项目组成员，这样可以保证问卷调查对象对项目有一定的了解程度。因此，最终问卷调查对象是运营产品事业部的所有人以及各项目组借调的实施工程师，总计一百多人参与了此次的问卷。我们通过“问卷星”这种小程序发放到运营产品事业部各项目组群，并让每个成员作答，所有人员填写后系统自动统计调查结果。调查问卷发放对应的职务统计情况如下表 3-1 所示：

表 3-1 各职位调查统计表

调查对象	项目经理	项目助理	需求	研发员	测试	实施	总计
人数	4	4	8	59	27	12	118

最后,根据上述问卷调查的结果进行分析,本次问卷调查的填写总计 118 人,且都是参与项目组超过半年的成员,所以所有的问卷调查结果填写都是有效的。

3.3 归纳总结项目管理存在的主要问题

通过对项目组问卷调查结果的综合分析总结,各个项目组在敏捷软件项目管理方面都存在以下相通的问题:

3.3.1 缺少固定工程人员的组织架构问题

一个敏捷团队的成员,应该有项目经理、开发人员、测试人员、实施人员、需求分析成员、评审组成员,并且配备相应的项目助理,目前项目组的实施人员不固定,没有自己内部的运维实施人员。公司有单独的大的工程实施部门,每次借调人员需要汇报给领导,领导再去跟工程实施部门领导沟通协调资源,有时资源还不一定能够及时协调到合适的工程人员,特别是遇到那种紧急的项目,如果没有工程人员就会延期项目的上线。有时协调到资源,但是协调到的工程人员不一定熟悉我们的系统,如果熟悉系统的人会很快上手,直接去客户现场对接部署,如果是新人还需要时间熟悉部署流程,就会浪费大量的沟通协调成本。新人在实施部署前还得花时间去了解项目的系统功能、部署架构,部署环境问题还要咨询项目组的开发、测试人员协助解决,因为没有对该项目的实施经验,也不能够及时准确的定位问题原因,导致花费大量的时间,也会影响生产上线进度或者上线后问题解决的不及时带来的客户投诉。

3.3.2 基于紧前关系绘制图的进度管理问题

项目进度管理是整个项目周期计划的管理,并规定每项计划在什么时间段完成。项目进度管理是将项目的任务拆分成每个子任务,并确定好完成时间,部分子任务还存在各种依存关系,可通过紧前关系绘图法(PDM)详细描述每项子任务的开始时间和结束时间,紧前关系绘图法有 4 中逻辑关系,即:某项子任务完成后下一子任务才能开始(FS),某项子任务结束后下一子任务也跟随结束(FF),某项子任务开始后下一子任务也跟着开始(SS),某项子任务开始后下一子任务才能结束(SF)。这四种逻辑关系的定义如下图 3-3 所示,其中,最常见且最常用的是 SF 关系。

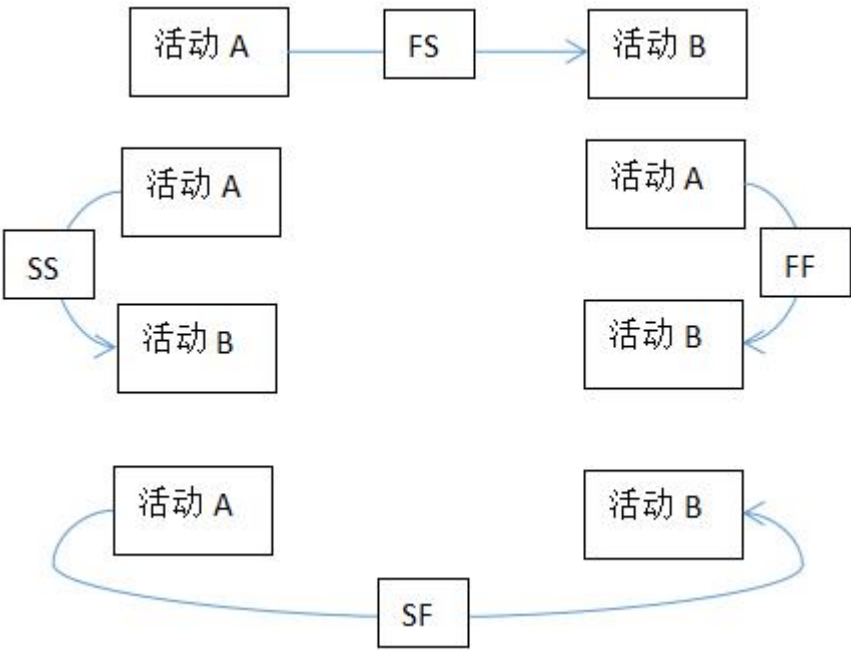


图 3-3 紧前关系绘图法

目前项目组进度计划使用的是里程碑式，是计划时间点，通过燃尽图（如下图 3-4 所示）查看进度，没有详细的进度计划，不能详尽的展示这种子任务之间的依赖关系，也只能看到一个大概整体进度，项目进度管理工具的缺失，项目经理没法预估进度风险，导致未能在客户承诺时间完成而延期交付。

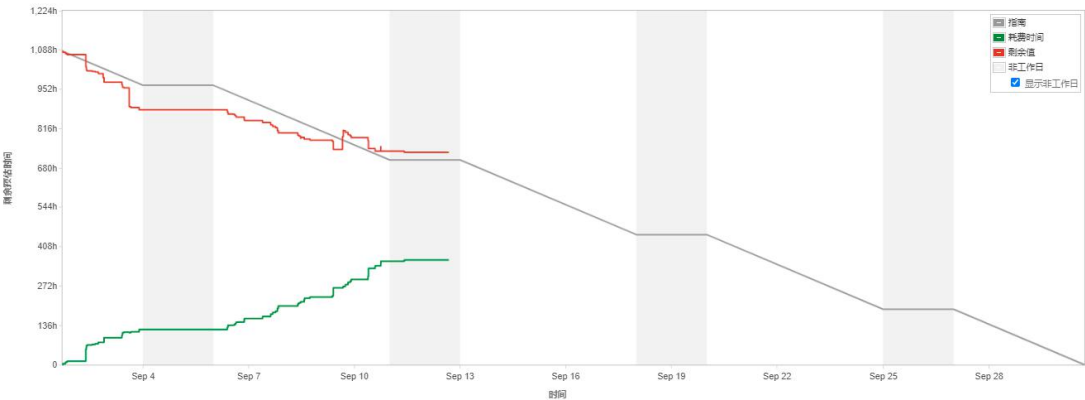


图 3-4 燃尽图

在每个版本项目迭代中，每天早上都会进行站立会，预计十五分钟左右，参与人员包括项目经理、开发、测试和项目助理，每次早会团队成员简述一下工作进度（包括每个需求完成情况以及进度）以及有没有阻塞和需要待讨论的问题，如果有需求不清晰的地方，则会邀请相关的需求人员参与会议；如果存在问题阻塞流程的都会在会上说明，找到对应的解决人员，会后相关人员参与讨论解决；如果存在需求变更的，就需要提出来并记录变更，虽然每次早上都会做个汇报，

但是没有实际的工具去记录这些进度,目前唯一的工具就是 JIRA 上需求的状态,因此项目经理无法准确的看到项目的具体进度,也无法通过这些预估是否能够正常发布版本。具体的项目实施过程中经常会遇到需求研发过程中变更或者前期分析不详细导致研发周期延长,无法按时完成需求开发提交给测试人员,测试人员的预估时间就会被压缩,无法保证版本的质量以及产品的正常交付时间。

据统计,J公司在实施敏捷前未使用 JIRA 工具,大家基本上使用的是 project 工具甘特图记录需求、开发、测试、实施时间,更新进度,一到两天会去维护一下该计划,并上传至 SVN 固定目录,这样项目经理就可以在 SVN 随时看到项目总体进度,并对延迟进度的找到具体延迟原因,并对其进行适当调整;但是,自从进入敏捷开发项目管理后,大家都依赖 JIRA 工具,并未详细的记录每项任务的进度,也不能准确评估具体版本迭代的进度,导致经常在一些项目中出现进度把握失控,延迟交付产品的现象。

因此,整个项目组需要优化现有的敏捷项目进度管理,就需要一套项目进度管理工具来准确的判断项目的进度。

3.3.3 基于挣值管理的项目成本管理问题

成本管理,除了生命周期内的每项工作成本外,还需要考虑的是后期的维护成本,缺一不可。成本管理的过程包括计划成本、和成本估算以及成本控制。它为了让项目在有限的人力资源、配套的软硬件设备和时间范围内,充分利用项目组的人力资源和软硬件设置,降低软件项目的管理成本或者控制在既定范围内,保质保量的交付产品。其中常用的有一种成本估算方法基于 COCOMO 模型的分阶段软件,以工作任务分解(WBS)以及进度作为基础方法提高成本估算的准确性^[29]。成本控制能准确的发现实际成本与预计成本的差额,分析不一样的缘故,使用正确的措施解决。

敏捷方法在成本效益方面非常有帮助软件的发展,将极大地有益于 IT 领域的组织^[40]。挣值管理需要通过进度和成本来综合考虑,在某种特定的范围内达到最优。它可以通过计划价值($PV = \text{计划工作量} \times \text{预算价}$)、挣值($EV = \text{实际工作量} \times \text{预算价}$)、实际成本($AC = \text{实际工作量} \times \text{实际价}$)、完工预算(BAC,成本基准)这些指标计算绩效,主要考察的计算指标如下图 3-5、3-6 所示:

中文名称	含义	计算公式
成本偏差(CV)	截止到某个时间段已经发生的成本偏差, 如果大于0则未超过预估成本, 小于0则已超过预估成本;	$EV-AC$
成本绩效指数(CPI)	截止到某个时间段, 实际已经发生的钱做了多少多少钱的事情, 如果大于1则投入值得, 如果小于1, 则投入比收益少;	EV/AC

图 3-5 成本绩效指标

中文名称	含义	计算公式
进度偏差(SV)	截止到某个时间段已经发生的进度偏差, 如果大于0则表示进度提前, 小于0则表示进度延期;	$EV-PV$
进度绩效指数(SPI)	截止到某个时间段, 实际进度与计划进度的比值, 如果大于1则进度提前, 如果小于1, 则进度延期;	EV/PV

图 3-6 进度绩效指标

目前事业部中每个项目的产品都配备了相应的项目助理, 项目助理的工作范围包括辅导团队正确应用敏捷实践、引导敏捷价值观的实现; 及时发现并排除团队中遇到的困难和障碍; 协助团队持续过程的改进、提高团队的生产力; 跟踪项目进度、更新团队项目进展, 并使之高度可视, 定期发布项目报告; 组织项目组中会议, 包括需求梳理会、项目启动会、每日站立会、评审会等。其工作并不饱和, 因此, 针对这种情况完全可以多个项目组共用一个项目助理, 减少项目助理人员的数量, 即可以减少人力资源。

与此同时, 每个项目组或多或少的都会存在项目变更的情况, 有的是由于需求未分析清楚, 导致开发测试过程中发现问题, 有的是由于开发完成后提交测试但是发现代码提交不符合公司的代码规范要求, 开发需要针对这些地方再次按照代码规范要求改进, 这两种情况的发生就会导致测试、开发时间的延长, 导致项目每个版本迭代周期的延长, 如果不能延长计划, 那么对应的人员就会加班完成任务, 加班过多的人就越容易疲劳, 做事情就没有精力和激情, 大脑思维反应变慢, 因此就会导致事倍功半, 没有达到预期的质量要求, 随之也就会提高项目的成本。因此, 基于这种情况可以通过增加需求设计评审或者需求分析不清晰不开发的措施解决需求分析问题, 可以通过增加代码评审, 严格按照开发规范编码来解决开发的返工。

3.3.4 基于持续增量改进项目质量管理问题

质量管理是软件项目管理过程中至为关键的过程, 他决定着软件应该以什么

样的质量交付，因此，软件项目的质量决定着一个项目的成败。如果一个产品不重视软件的质量，那么就会增加开发和测试的成本，也会让客户对该产品产生质疑，如果是交易类的产品发现缺陷就会导致让客户不能下单成交或者清算错误，让客户造成严重的经济损失利益，有的客户还会往上级投诉，券商收到上级发的警告函，影响券商的评级，进而会导致客户的流失。但是，不是只有质量越高就会越好，质量高那么就会需要高的成本，我们需要考虑项目的成本和收益才能决定项目的质量标准。

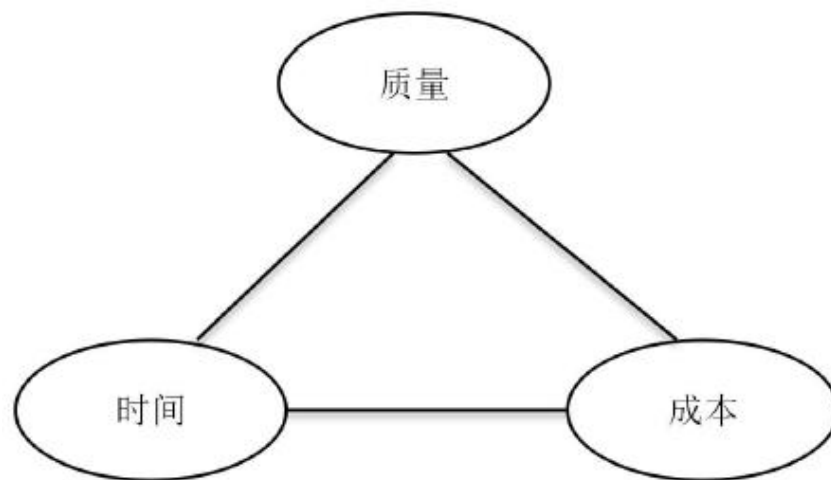


图 3-7 项目管理三角模型

如上图 3-7 所示，早期项目管理中时间、成本、质量是关键的部分，缺一不可的，彼此之间相辅相成，缺少任何一个要素都不行。比如想要质量有保障，那么就必须投入更多的时间，以及加大的成本投资；想要低成本，就需要通过质量的下降和项目周期的减少来完成；想要按时交付产品，那么就需要投入更多的成本保证产品的质量。在此模型下，产品负责人为了准时交付产品以及不超过公司项目成本，经常会下意识的缩减项目资源和时间，导致产品质量大幅下降，不能给予保障，也就给项目带来了一定的风险。敏捷开发项目管理能够通过每个迭代来进行不断交付的模式保证质量，最终保证了产品的质量，可以提高生产率，降低项目的管理成本，一旦质量得不到保证，那么项目的管理成本就会相应增加，项目的风险就会随之增加。

在质量管理环节中，需要针对每个环节去分析，通过使用流畅、稳定的持续增量改进的方法以致达到质量管理的要求，可以使用六西格玛管理（六西格玛管理就是 6 个标准差管理）、精益生产等方法实现。

目前事业部项目组中，针对测试这块，如果时间充裕则会编写测试用例，如果时间不充裕则不会编写测试用例，导致部分需求的测试用例缺失，即使有测试用例也未对其进行评审，无法判断测试用例的覆盖度是否全面，因此，没法保证

产品质量。为了解决这一问题，针对测试用例这块如果时间不充裕可以使用思维导图编写对应用例，并对测试用例和编写的思维导入进行评审，评审人员包括需求、开发、测试人员，这样就可以提前发现测试用例是否符合需求，以及开发人员在代码开发过程中某些逻辑是否考虑的不全或者同测试人员的逻辑有偏差。

另外，为了能够让每个项目的版本迭代质量得到改善，可以增加一个项目结束后的回顾会议，发现项目中的好的管理规范，并对好的管理规范提倡并继续执行，指出项目中不良的恶习，制定相应的方针改掉不良的规范，避免下次再次出现。从而间接的提升项目的管理。

3.3.5 基于 5C 原则的项目沟通问题

项目的沟通管理就是积极与项目组中团队成员以及客户相关人员的密切联系。沟通的方式有多种，比如书面、口头、会议、邮件等等方式。通过简洁的语言表达、面部表情、身体动作等来发送和接收信息。一个成功的沟通需要根据项目组成员的不同制定适合自己的沟通策略，然后形成对应的沟通管理计划通过各种方式把正确的信息传递给对应的人，避免出现理解错误信息。

成功的沟通需要遵循 5C 原则，即言语表达要正确（correct）、文字表述要简洁（concise）、结果表述要清晰（clear）、逻辑思维要连贯（Coherent）、条件语句要受控（control），基于这些原则也需要通过一些技巧配合才能达到，比如沟通过程中需要认真聆听，减少因个人理解偏差导致的误解，并强化自己的沟通技巧等。因此，有效的沟通需要确定沟通的目的，尽量了解对方理解能力，通过通俗易懂的方式表达，尽量让每个人都能接收到信息。

因为项目的特殊性，项目对接了很多系统，包括内部的综合运营系统、影像系统、手机、VTM、PAD 项目组的相互对接，还有其他部门的系统对接，包括 JZJY 系统、RZRQ 系统、ZH 系统、GGQQ 系统、OTC 系统、资金管理系统等，内部系统对接相互之间共同成长和进步，业务相对较熟悉，所以沟通会较顺畅，但是针对其他部门的系统对接，由于业务较多，很多业务不熟悉其业务设计逻辑，只能去咨询对应系统的相关人员，但是很多人员都不是很熟悉，且别人工作量也比较饱和，也没有具体的项目对接人去解答问题，导致经常咨询问题随意找他们的开发、测试人员，很多人也不愿意回复，因此只能一个一个的去咨询，如果有人回复就好，没人回复只能自己去他们系统摸索。

新产品一般周期较长，且也会存在三司三所需求变更的情况，需求的变更有时会重新提交新的需求文档，有时只是口头通知开发和测试，所以没有一个最新的全量需求文档，导致后续其他需要了解这个需求的找不到最新的全量需求文档，需要自己去摸索熟悉以及咨询对应的开发和测试人员；

项目助理记录问题需要同其他项目组协商讨论，不能具体到问题点，还得相应的人员参与，导致浪费了大量的沟通时间。

第4章 基于敏捷开发J公司项目管理优化方案设计

通过上一章节分析了J公司敏捷软件开发项目管理的问题，主要是组织结构、交流不顺畅、进度工具不完善、未避免不必要成本、质量控制不规范引起。这些问题是现有的敏捷开发模式方法需要解决的，因此，本章节希望优化现有的敏捷项目管理方法，全面将敏捷的软件项目管理理论体系运用到J公司的软件项目管理中，并快速解决现有的敏捷软件项目管理问题，后续J公司其他软件项目管理方法可以参考并借鉴使用，提升公司所有项目的软件管理水平。

4.1 基于敏捷开发优化方案设计原则与思路

在针对现有公司敏捷项目管理模式优化的基础上，不能脱离《世界敏捷开发联盟》的总体原则与思路，现阶段J公司优化的原则与思路也要按照这些进行，对现有的敏捷开发方案进行优化调整。

4.1.1 敏捷开发优化方案设计的基本原则

基于《世界敏捷开发联盟》提出的12项原则，再结合J公司敏捷软件项目软里的现状，本次优化的原则如下：

- (1)、优化组织架构，对现有架构不做大的变动；
- (2)、优化团队的沟通，明确团队内各成员职责；
- (3)、优化项目进度工具，提升敏捷的效率；
- (4)、优化项目的成本，减少项目成本的支出；
- (5)、优化项目的质量，提高项目的质量；

4.1.2 敏捷开发优化方案设计的总体思路

从上一章节分析可以得出，J公司的现有敏捷软件开发项目管理存在组织架构、沟通、进度、成本、质量的缺陷，导致了开发存在明显的返工、需求变更频繁、项目延期交付、质量无法保障的问题，J公司针对这些项目使用的敏捷方法需要对他们进行优化，优化总体思路如下：

- (1)、优化组织架构，增加对应项目的工程实施人员，并确保其稳定性；
- (2)、优化团队的沟通问题，建立各项目的具体对接人，问题汇总由个对接人沟通协调；需求的变更需要做好文档维护工作，并通知团队的对应人员；
- (3)、优化项目进度工具，支持 Scrum 与 Kanban 的使用，并且项目助理增加使用甘比特图记录每次站立会的进度，并汇总发送给产品经理、项目经理等负责人；

- (4)、优化项目的成本，减少项目助理的数量，可以多个产品使用同一个项目助理；
- (5)、优化项目的质量，提高测试团队的用例覆盖度，增加测试用例评审环节；需求做好评审分析，记录问题具体到对应的人后找他们相应解决；增加开发设计分析评审，提前了解开发的思路是否符合要求，这样可以避免在测试阶段或开发一半发现缺陷后重新返工，以此来提高开发效率给测试流出充足的时间测试，保障了产品质量。

4.2 基于敏捷开发软件项目管理优化的方案设计

4.2.1 增加固定工程人员的组织架构方案设计

在敏捷开发团队中，需要需求人员、开发人员、测试人员以及实施人员，他们是缺一不可的，每个岗位都非常重要，他们需要团结一致专注于每个迭代结束时交互有价值的功能数量。但是在现有模式下，项目团队的实施人员不属于同一个部门，沟通困难且人员不稳定，在项目亟需用人时还需同另外一个部门领导协商，如果当前时间段工程部没有多余人员借用，那么只能延期，如果工程部借调的是新人，那么还需对他进行项目的培训，讲解相关的系统业务，这些问题的存在都势必会影响项目的进度，使产品延迟交付。为了解决这一问题，现将公司的组织架构做如下调整，撤除原有独立的工程部，将每个工程实施人员划分到每个产品事业部中，成立包含需求、实施的产品交付部，具体调整结构图如下图 4-1 所示：

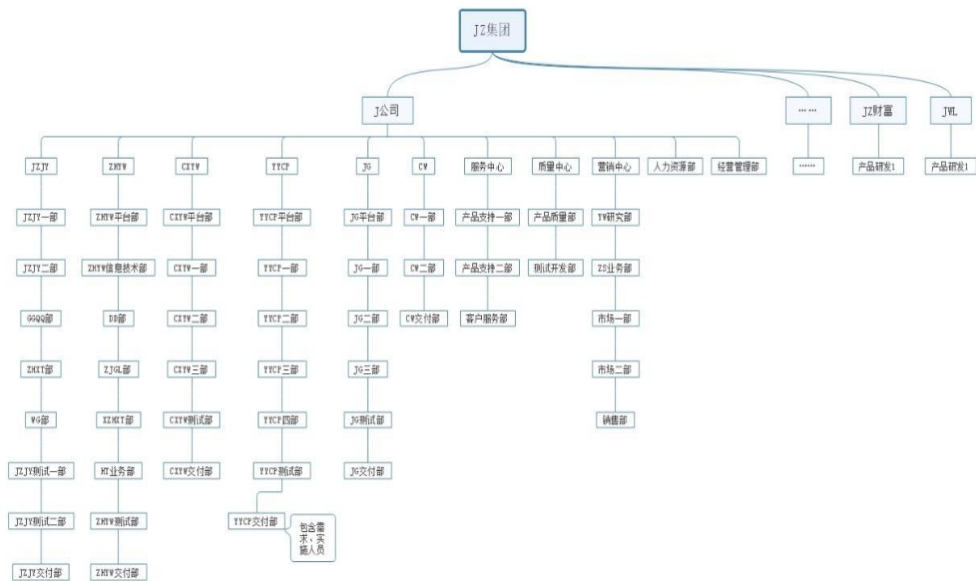


图 4-1 优化后的组织架构图

工程项目实施过程中会产出各种类型的交附件，主要会产出以下几种类型交附件：文档类、脚本类、工具类、源代码类（包含产品源代码、工具源代码）、邮件类、视频类、语音类。

项目实施过程中，各阶段的交付件是项目的重要产出和资产，同时也各阶段工作完成的标志。交付件管理覆盖到项目实施全过程，项目经理对所有交付件的产出的及时性与质量负责。

管理要求：

（1）、项目交付件管理在整个项目期间开展。项目经理对所有交付件的产出的及时性与交付质量负责；

（2）、项目过程中的所有交付件均需及时归档 SVN 配置库对应目录；

（3）、项目组各成员按时完成项目经理分配的任务，并及时产出任务所涉及的交付件。项目经理及时检查项目组成员的工作成果，审核任务产出交付件质量；

（4）、正式提交给客户的交付件均需经过项目经理审核且带有版本号，后续更新做好版本更新记录；

（5）、脚本交付件在每个文件备注其含义并且备注描述要清晰，脚本交付件的命名需对脚本的使用场景或解决的问题清晰展示；

（6）、项目助理定期对项目的交付件提交情况及进行检查；

（7）、源码及重要设计文档不可私自提供给客户，具体参考《J公司源代码与重要文档的管理制度》。

4.2.2 敏捷开发基于进度网络分析的进度管理方案设计

进度管理一直是很多软件公司做项目的一个重大难题，如果太粗犷会造成项目管理的脱节，项目后期问题频繁，太细致会造成管理消耗成本升高。进度管理通过进度网络分析方法制定，它包含了关键路径法、资源优化、数据分析、提前量与滞后量、进度压缩这5种技术。其中，最常用的是关键路径法，其他的几种技术都是在它的基础上衍生出来的新方法。关键路径法是在已知计划必须完成时间以及人员资源充足的情况下，预计项目中各子任务开始、结束日期。根据已知的完成时间逆推出每项任务的完成时间，再通过预期时间计算出最晚的开始时间。此方法使用的前提就是需要有充足的人力资源，如果人力资源不充足，那么就会导致项目的交付时间延长。在每个迭代版中，项目经理需要与客户或其他需求提出人员确定每个需求的版本发布时间，制定项目的里程碑进度计划，然后再同项目组中的团队成员确定每个迭代的需求数和日期编写计划。计划制定好了后需要对该计划的执行进行控制，避免出现偏差。通过关键路径法来控制进度的

绩效,然后通过查看每个迭代的燃尽图,迭代启动后有一个指南,可以通过指南作为基线,了解每个子任务的耗时时间和已经剩余时间。

软件项目进度的管理需要通过各种有效的方法对实际的项目开发的进度进行管控。比如,软件的项目助理每日组织站立会,对每个迭代的任务进行汇总统计,及时发现项目过程中的风险和存在的问题,问题责任到具体的人,一旦出现开发或测试资源不足导致延期等问题,可将问题收集后反馈给部门领导寻求解决方案,协调资源来保证每个迭代的正常交付。

J公司运营产品事业部在运用了敏捷方法的 Scrum Board 和 Kanban 工具后,在每个版本项目迭代中,团队成员的工作任务被分解成每个小的事故或子任务,通过 Scrum Board 和 Kanban 来记录,这样能够方便开发和测试人员更好的配合每个需求,也省去了一些沟通上的问题,每天早上都会进行站立会,每个参与人员汇报一下当前的工作情况,包括每个需求完成情况以及进度,但是这些不能详细的记录每项任务的进度,也不能准确评估具体版本迭代的进度,导致经常在一些项目中出现进度把握失控,延迟交付产品的形势。为了避免这些情况的出现,可以在现有的敏捷项目管理中增加使用甘比特图记录每次站立会的进度,并汇总发送给产品经理、项目经理等负责人。详细步骤如下:

项目助理组织项目经理、团队成员参加每日站立会议,站立会的原则有:

- (1)、所有参会人员都必须站立开会;
- (2)、会议时间最好不要超过 15 分钟;
- (3)、每日站立会固定时间、固定地点,不允许迟到,针对迟到会有相应的惩罚机制;
- (4)、每个人员都必须简洁描述前一工作日的内容,当天的工作计划,当前工作中有什么困难需要解决;
- (5)、使用物理看板的团队操作原则:

①建立物理看板的团队在每日站立会开始前需更新看板中迭代中每个需求的状态;

②物理看板中超过在制品上限的需警示,不能再往此状态中拉入需求;

③团队中每个成员需关注过程中的阻塞问题,并及时反馈。

团队成员在工具 JIRA 中填写子任务描述、分析、实现概要等,每日编写工作日志,估算剩余工时;

项目助理每日统计迭代任务列表、燃尽图、缺陷解决情况,记录迭代任务列表的进度,从bug类型、根源、单个需求bug数等方面进行分析,及时发现迭代风险及问题并揭示,发布敏捷项目日报。

每日完成标准:

(1) 所有提交的代码均已通过单元测试或开发之间相互交叉测试，并出具相应的单元测试报告；

(2) 每日在 JIRA 中更新工作日志，记录工作时长，描述工作内容，预估剩余所需工时；

(3) 当天发现的缺陷开发必须当天解决，遗留最长不超过 24 小时。

管理流程指导：
所有的风险问题需合并至日报中，并重点体现出来，日报发给项目组所有参与人员。

4.2.3 敏捷开发基于净值管理的成本管理方案设计

项目一启动就会有成本的存在，产品的研发、使用、维护、管理是个大工程，需要大量的人、财、物支持，这些都应该在成本管理中被关注，并通过各种途径和方法降低项目的成本。在软件项目管理中，因为软件开发有很多不确定因素，项目周期长导致软件开发成本不可预估，所以项目经理要对项目中各项成本进行及时高效的严格控制，确保项目按计划进行，不会造成项目亏损或者项目失败。

4.2.3.1 人力资源的调整

软件项目的核心就是在既定的时间和预估成本内，将有限的人、财、物重组在一起，用最少的成本，发挥最大的作用实现最初的目标。软件开发行业主要依靠智慧来创造价值，大多数的开发人员都渴望能够挖掘个人潜力，并能够得以超常施展出来实现存在的价值，这也是鼓舞员工的最灵验方式^[41]。

敏捷开发团队成员以 5-9 人为最佳，由产品经理确定团队的项目经理、开发、测试、需求分析成员，然后讨论确定项目助理，项目敏捷团队组建后，由项目助理申请开通团队相关权限，比如 SVN、JIRA、测试管理平台等。在敏捷项目中，每个成员都有不同的分工，并且他们都有对应清晰职责范围，每个项目组成员都应该保质保量的完成指派的计划任务，保证按时完成验收，如下表 4-1 所示：

表 4-1 项目团队角色以及对应职责表		
角色	角色定义	职责
项目经理	确保团队做正确的事情，产品与团队的价值合理化，管理产品待办列表的负责人	1. 清晰的表达产品待办列表，对其用子任务优先级划分，明确验收标准。 2. 规划迭代目标、迭代版本发布的内容、日期 3. 确保团队对迭代待办列表中的子任务认知一致。 4. 负责迭代用户故事验收。 5. 主持迭代启动会、评审、迭代验收会议。 6. 参与过程改进任务，如度量、重构、项目自动化等。

表 4-1 项目团队角色以及对应职责表(续表)

角色	角色定义	职责
开发负责人	负责版本整体开发质量	1、对升级包所有开发过程和产出负责，包括设计方案、单元测试、代码审核、缺陷修复、发包后质量等。
测试负责人	负责版本整体测试质量	1、对升级包所有测试过程和产出负责，包括测试方案、测试用例、测试用例评审、测试执行、发包后质量等。
项目助理	负责确保团队正确的做事。所有人都能够理解并实施敏捷 Scrum 与 Kanban，是团队中的服务型领导	1. 辅导团队正确应用敏捷实践，引导敏捷价值观的实现。 2. 及时发现并排除团队中遇到的困难和障碍。 3. 协助团队持续过程改进，提高团队生产力 4. 跟踪项目进度，更新团队项目进展，并使之高度可视，定期发布项目报告。 5. 组织项目中会议包含需求梳理会，启动会，站立会，评审会、回顾会。
需求人员	负责产品需求整理	1.负责迭代中客户需求的背景、原因、详细内容讨论和确定，以确保迭代内需求清晰明确。
实施人员	负责产品上线部署以及维护	1.负责系统安装部署、联调测试、全网测试、性能测试等实施工作； 2.负责实施缺陷收集、确认、反馈和进度跟踪。
团队成员	负责产品需求实现	1. 理解并实践敏捷。 2. 负责子任务的故事点估算 3. 认领迭代任务列表中的任务并按能力估算工时，找到最合适的计划保质保量交付任务。 4. 向项目利益相关人演示工作成果。 5. 团队自组织自我管理、持续改进专业技能与团队沟通。

除此之外，敏捷开发项目管理中新增了过程组：包括需求分析组、评审专家组，详细情况如下表 4-2 所示：

表 4-2 项目过程组以及对应职责表

过程组	组成成员	过程组职责
需求分析组	产品经理、项目经理、产品线服务人员、客户代表（如有）	1、负责迭代中客户需求的背景、原因、详细内容、验收标准的讨论和确定，以确保迭代内需求均清晰明确 2、讨论确定迭代需求的分类及优先级 3、有权剔除内容不明确、范围蔓延等需求
评审专家组	产品、项目、开发、测试负责人及其他相关专家	1、参加项目的所有评审活动，包括设计、代码、用例、发布评审等，并提出有效的评审意见； 2、组织或参加缺陷分析、质量回溯、迭代回顾等过程，并提出有效的评审、改进意见。 3、确定过程中的完成定义工作是否符合要求，如：测试工作、迭代需求、发布评审等是否符合要求。

正如前文提到的，现每个项目中都配备了项目助理，项目助理也就是辅助团队使用敏捷实践，跟踪项目进度，参与项目会议，记录每次会议纪要，并将会议纪要整理出来发送邮件，项目助理的工作不饱和，因此针对这种情况完全可以多个项目组共用一个项目助理，减少项目助理人员的数量，即可以减少人力资源或者将减少的项目助理人员用于增加初级的开发或测试人员。

与此同时，每个项目成员中的人员结构配比也要合理，每种类型的人员其中级别应该合理配置，尽量遵循普通人员多余高级和助理级别。开发和测试的配置比例应该是 2:1 或 3:1。

4.2.3.2 减少需求的返工

敏捷开发的重点是需求优化，采用迭代、循序渐进的方法开发。敏捷开发能够灵活接受需求的变更的，那么就需要做相应的需求管理，需求的管理能够保障团队成员都能熟悉需求的目的和实现方式，其处理流程下图 4-2 所示：

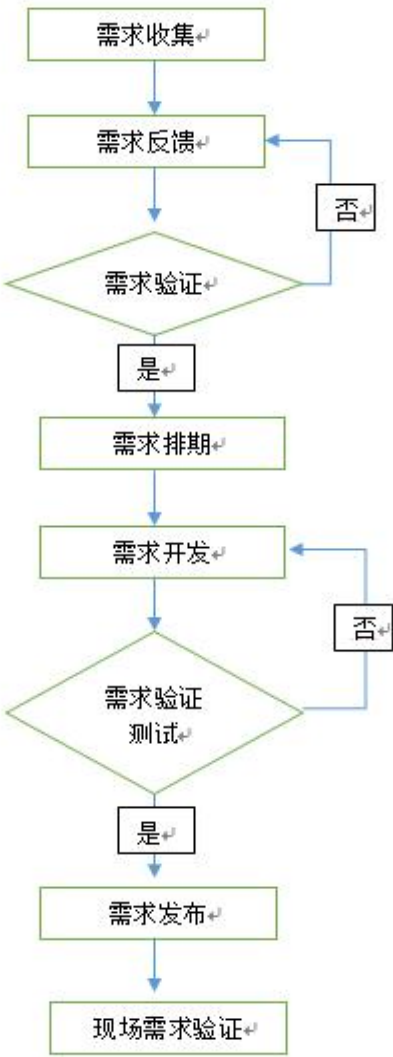


图 4-2 需求处理流程图

从原则上讲，所有的版本计划制定后，其需求是不可变更的，但是不排除一些特殊情况的存在。

首先，每个项目组都或多或少的会存在需求改动的现象，主要有两个原因，一个是由于需求分析不到位，在开发部分功能时遇到某些逻辑不合理需要重新梳理需求，因此就会出现需求的返工，这种情况的存在会造成开发和测试人员对需求变更的厌恶，同时也会提升项目成本。另外一个是客户的需求变更导致在原有功能上新增功能，造成版本迭代延期交付、增加项目成本。在每个迭代计划进行过程中，对于改动的部分，就需要对原有计划再次预估，变更后的风险点，项目成本、开发、测试人员的工作量、是否能够如期交付等。经权衡利弊后，项目经理可以依据该需求的紧急程度判别，如若需求不紧急可将此需求剔除本次迭代放入下一迭代实现，如若需求紧急必须要此次迭代完成，那么可以考虑剔除本次迭代中优先级低的未完成需求并重新调整迭代计划。除此之外，需求的变更需要需

求分析人员更新文档，记录变更时间、内容，这样可以做到需求的可追溯性。

其次，每个项目组中也会存在开发完成后提交测试但是发现代码提交不符合规范，又重新修改代码提交，如果测试已经完成了之前的测试，那么针对新的修改就会重新投入人力去测试，必将导致开发、测试周期增加。因此，基于这种情况通过增加代码评审需求，严格按照开发规范编写代码，减少需求的返工。针对开发不规范的设置惩罚机制，首次可以给予警告或提醒，如果多次出现就要求处罚，将此处罚作为项目组经费。

4.2.3.3 减少评审环节出现低级错误

敏捷开发的执行过程中会出现计划、需求、设计、代码、用例、发布等一系列的评审工作，为了避免评审中出现零容忍度的差错而浪费时间，在每项评审工作前需相关的被评审人员提前按照评审检测清单列表检测各自文档是否存在这种低级错误。

(1) 项目计划评审，如图 4-3 所示：

重要性	checklist	是否符合该要求 (作者自检)
必选	是否覆盖模板要求要素编写立项申请报告？项目名称是否恰当？	
必选	是否对客户、市场前景、同类产品、竞争对手做了充分调研？	
必选	公司是否已有同类产品，是否重复性开发？	
必选	项目的目标是否清晰，是否可达？	
必选	项目范围是否合理？	
必选	验收标准是否清晰并且是否可行？	
必选	技术方案是否合理，是否充分考虑公司已有平台、架构及组件的复用可能？	
必选	是否识别了项目的风险，并指出其影响程度，影响程度的判断是否合理。	
必选	项目计划及进度方案粒度是否合适，并有各阶段和活动对应的交付要求？	
必选	资源安排是否恰当，是否存在大量的不确定人员？	
必选	项目工作量估算是否合理？	
必选	项目成本是否按照公司要求进行计算	
必选	是否识别了项目的关键问题，关键问题前置条件是否合理，拟解决时间是否恰当？	
必选	是否调查与本产品相关的政策，分析业务开展的合法、合规性（有无政策“支持”或者“限制”）？	
必选	是否已经存在某些专利将妨碍本产品的开发与推广，是否识别本产品中无新的专利点可申请得到知识产权保护。	

图 4-3 评审计划检查图

(2) 需求评审，如图 4-4 所示：

重要性	checklist	是否符合该要求（作者自检）	是否符合该要求（评审组长检查）
必选	是否覆盖模板要求要素编写业务需求说明书？		
必选	是否界定了系统范围？		
必选	是否清晰介绍了需求的背景和价值？		
必选	是否是从客户的角度来描述对系统的需求？		
必选	是否对客户业务需求的功能进行描述、需求是否完备？		
必选	是否对客户非功能性业务需求进行了描述？		
必选	系统输入和输出是否定义充分？		
必选	是否对要处理的主要数据对象进行了描述？		
必选	是否对外界的接口进行了描述？		
必选	是否对业务环境进行了说明？		
必选	评估是否所有需求点都可实现？		

图 4-4 需求评审检查图

(3) 开发设计评审，如图 4-5 所示：

重要性	checklist	是否符合该要求（作者自检）	是否符合该要求（评审组长检查）
必选	1、是否覆盖模板要求要素编写详细设计说明书？		
必选	2、是否清晰的描述了单元设计信息，包括数据流程、控制流程、接口？		
必选	3、是否说明了单元功能的目的？		
必选	4、是否对单元功能进行了描述？		
必选	5、是否说明了用于实现该单元的算法？		
必选	6、设计是否能实现特定的需求和目标？		
必选	7、详细设计中的每一函数单元/数据元是否有唯一标识，标识是否合理？		
必选	8、是否存在逻辑上的问题？		
必选	9、详细设计中的每一函数单元是否有相应的软件功能项对应？		
必选	10、是否对每个子模块的各个关键过程/函数给出了伪代码或者描述？		
必选	11、设计实现上是否与《概要设计说明书》保持一致？		
必选	12、参数的数量、类型和顺序是否与接口匹配？		
必选	13、是否正确的定义了输入输出数据？		
必选	14、是否清晰的描述了传递参数的顺序？		
必选	15、是否识别了传递参数的机制？		
必选	16、设计单元是否具有高内聚度低耦合度？（即该单元的变化不会对本单元造成不可预料的影响，对其他单元的影响达到最小）		
必选	17、是否对输入、输出、接口和结果进行了错误检查？		
必选	18、是否对错误情况给出了有意义的信息提示？		
必选	19、是否考虑了意外情况？		

图 4-5 设计评审检查图

(4) 开发代码评审，如图 4-6 所示：

重要性	checklist	是否符合该要求 (作者自检)
必选	1、变量和常量的命名、方法名的描述方法是否与约定保持一致?	
必选	2、变量和属性是否书写正确?	
必选	3、变量和属性是否被正确的初始化?	
必选	4、非局部变量是否能用局部变量替换?	
必选	5、变量和属性是否可以用常量替换?	
必选	6、所有的属性是否都有正确的访问限制符(private, protected, public)?	
必选	7、每一个方法, 类和文件是否都有适当的头注释?	
必选	8、每一个属性, 变量和常量的声明是否都有注释?	
必选	9、方法和类的头注释是否和它们的功能保持一致? 注释和代码是否保持一致?	
必选	10、对于每一个方法, 它是否都返回了正确的值?	
必选	11、每一个方法是否都结束?	
必选	12、每种方法是否都有正确的访问限制符(private, protected, public)?	
必选	13、所有的for循环的控制变量是否都在循环顶部被声明?	
必选	14、在子类中是否有应该放到父类中的通用成员?	
必选	15、对于每一个数组引用, 下标值是否在定义的范围之内?	
必选	16、对于对象和数组引用, 是否应确定其值应为非空?	
必选	17、是否存在不同类型数据之间的混合计算?	
必选	18、在计算中是否存在上溢或下溢的可能?	
必选	19、关于数值计算的顺序和优先级的假设是否正确?	
必选	20、是否用了括号来避免模糊不清?	
必选	21、对每一个布尔测试, 正确条件是否被检查?	
必选	22、比较操作符是否正确?	
必选	23、每个布尔表达式是否都正确?	
必选	24、所有的循环是否都能结束?	

图 4-6 代码评审检查图

(5) 测试用例评审，如图 4-7 所示：

重要性	checklist	是否符合该要求 (作者自检)	是否符合该要求 (评审 组长检查)
必选	1、是否覆盖模板要求要素编写测试用例?		
必选	2、是否每个测试用例的预置条件都被描述清楚?		
必选	3、测试用例的“执行步骤”是否清晰，正确?		
必选	4、测试用例本身的描述是否清晰，不存在二义性?		
必选	5、是否包含异常用例?		
必选	6、是否程序中的每个需求都有对应的测试用例?		
必选	7、测试用例的覆盖范围是否全面，是否覆盖项目需求中的所有功能点?		
可选	测试用例是否复用性强? (例如，可将重复度高的步骤或过程抽取出来定义为一些可复用标准步骤)		
可选	是否明确测试用例的设计方法? 如等价类法、边界值法等		
可选	每个测试用例的“输入”中是否列出了所有测试的输入数据?		
可选	测试用例的“预期结果”是否有明显的验证方法?		
可选	优先级安排是否合理?		

图 4-7 用例评审检查图

(6) 版本发布评审，如图4-8所示：

重要性	checklist	是否符合该要求 (作者自检)	是否符合该要求(评审 组长检查)
必选	1、是否包含代码评审、测试用例评审报告？		
必选	2、是否包含测试报告？		
必选	3、《升级包检查表》不合格项是否已修改完毕？		
必选	4、升级包内是否有需求修改影响其他产品，需要关联修改？		
必选	5、升级包是否做了回归测试检查？		
必选	6、是否包含程序升级说明文档？		
必选	7、文档是否归档并上传SVN？		

图 4-8 发布评审检查图

4.2.4 敏捷开发基于持续增量改进的质量管理方案设计

软件项目管理过程中，产品能否成功质量起到了至关重要的作用，如果产品的质量很差客户也不会使用该产品，他们可能寻找其他质量好的产品替代。质量，是高科技企业非常核心的竞争力，质量跟不上，所有的付出都会打折扣，甚至归零。一个企业能不能产出优秀的产品、有用的产品、能让客户放心的产品，不仅仅是跟人的技能本身相关的，还需要有质量意识，还有体制、机制去进行控制和管理。做质量管理，有技术的成分在里面，但是最核心的是我们有没有以产品质量为生命的意识。由此可见，质量是非常重要的，也需要整个团队中的每个成员共同来提升。

“质量是生命”是J公司内部也一直倡导的文化，强调‘技术为本、过程为器、质胜之道、工匠精神’的质量口号，技术是企业发展的根本，技术为本就是要求日常的工作中必须重视技术。过程是我们做事的步骤方法，公司制定质量管理白皮书，是结合了业界的最佳实践和我们自身的特点制定的，是我们每个人都应遵循执行的过程规范。在过程执行中我们要做好预防工作，做好事前控制，和质量管理部一起，在过程和预防方面做更多的事情，提升我们的产品质量。工匠精神来自于我们对产品的完美追求。它的内在精神是：完美、严谨、一丝不苟、耐心、踏实、专心、持之以恒、专业、兢兢业业、创新、拼搏等各种可贵的品质。这种精神需要每个人都能有心、用心去做好每件事，要向德国制造学习，信奉标准、完美、精准、专注和实用主义。影响软件质量的三要素：人、技术、过程，质量改进，就是为了不断提高我们的效率，用更小的成本去收获更大的效果。而最终收获和业绩有两个方式达成：一个是提升人员的能力，一个是改进我们的方

法。依靠人的能力称之为**人控**，改进方法称之为**技控**。人，有智慧，有情绪，会追求成就感，但是会有惰性，所以方法（技术和过程）会更加重要一些。技术是一个支点，通过这个杠杆作用，我们可以很省力地撬动起更大的绩效。

为了更好地提升客户满意度和细化管理，夯实质量基础，编写改进开发质量重点要求，优先补足软件研发和工程实施上的短板，建立有效的研发流程和工具，实现全过程质量控制和敏捷交付，如下表 4-3 所示：

表 4-3 研发质量改进重点要求表

专项	目标	关键改进型	度量指标
需求 响应	1、业务团队及时响应客户需求，推动遗留需求关闭率和问题及时解决率达到目标要求，提升客户满意度；	1、梳理遗留需求，并制定遗留需求改进计划，所有客户需求均需要与客户沟通确认，承诺完成日期和计划修复版本，确保按客户承诺的时间完成发布并关闭。	1、需求响应周期：需求关闭时间-需求提出时间
	2、完善需求管理过程，需求数据的集中统一管理和跟踪分析	2、对于新增需求均需要与客户沟通确认，并确定承诺完成日期和计划修复版本，需求实现后，需要与客户反馈解决结果，获得客户认可后才可关闭需求。	2、需求完成率：当年已关闭需求除当年创建需求（JIRA 流程中需求最终状态为已关闭）
事件		3、每周四前或事件驱动汇报进展及需求数据。按产品或状态或按年统计需求数，解决率等。	
		4、发现质量审计不符、需求处理过程中异常，需及时出具改进措施并按目标日期解决并关闭。	
		5、QA 部门负责需求管理办法拟定和优化，执行的监督检查、数据定期分析晾晒和汇报、推动改进	
	1、完善事件管理办法，依据证券期货行业标准修订事件定级标准，并发布；	1、按事件管理办法执行，所有事件统一上报到公司级统一管理；	1、事件瞒报和漏报个数，与黑事件挂钩
	2、客户事件集中统一收集、跟踪。	2、负责事件数据收集和分析（包含：部门、级别、总数量、事故来源、处理时长等）；	2、事件及时上报率：24 小时内上报的事件占比达 100%
		3、质量管理总部负责事件管理规范修订和发布，执行的监督检查、晾晒数据、参与事件调查，组织培训等	3、改进措施及时完成率：回溯制定的改进措施 100%关闭。
			4、各级事件处理时长：恢复时间-发生时间

表 4-3 研发质量改进重点要求表（续表）

专项	目标	关键改进型	度量指标
设计 评审	所有立项的研发项目都应该输出设计文档并经过评审。重大项目设计文档发技术与产品委员会评审，设计评审有效落地。	<p>1、《软件设计说明书》需按照同行评审要求来执行，对于重大项目设计文档提交技术与产品委员会评审。</p> <p>2、设计均须评审，需要项目经理、需求、设计、开发人员、测试参与。项目经理以及开发负责人负责把控设计质量。</p> <p>3、评审后，需出评审报告，并将记录都登记到 JIRA 系统并解决关闭。</p> <p>4、评审贡献（如：发现问题、改进意见或分享等）作为晋级或评优参考</p> <p>质量管理总部：</p> <p>1、评审数据晾晒（包含：按产品-评审缺陷数、评审质量），推动分析改进；</p> <p>2、评审相关过程或评审清单优化。</p>	<p>1、按评审人员、产品等维度统计评审问题数据；</p> <p>2、不合格项数据统计，如未做评审活动数据，重大项目设计文档未报送技术与产品管理委员会评审数据</p> <p>3、前三阶段缺陷比达到 40%以上。</p> <p>4、设计方案不足引入的内部测试问题/生产问题/事件</p>

表 4-3 研发质量改进重点要求表（续表）

专项	目标	关键改进型	度量指标
代码 评审	所有立项的研发项目都应该开展代码评审。内部组织代码评审经验分享，分享材料沉淀归档到 Confluence。	<p>1、研发负责人或高级别开发工程师检查开发所提交的代码，确保代码的质量。</p> <p>2、所有的高级及以上级别人员都需要进行代码评审并进行总结分享，分享记录按团队形式输出，并至少推选一人做公司级分享。</p> <p>3、开发工程师要做自测，保证产出代码的质量。</p> <p>4、关键、核心模块均需要开展正式的会议评审并使用代码评审清单。评审角色覆盖到项目每个人且有高级别人员参加，出具对应报告且登记到 JIRA 系统并解决关闭。</p> <p>5、评审贡献（如：发现问题、改进意见或分享等）作为晋级或评优参考。</p> <p>质量管理总部：</p> <p>1、质量管理总部负责对评审数据晾晒（包含：按人员评审缺陷数、分享记录）。提交优化需求-使用 KTMP 代码评审功能，评审问题反馈到工具中，每周统计代码评审数据</p>	<p>1、从多个维度统计问题数据</p> <p>2、度量研发项目代码评审覆盖率。</p> <p>3、统计每个高级别开发人员分享次数（分享形式不限，分享文档或 ppt 或 kdop 优秀代码分享总结等）。</p> <p>4、前三阶段缺陷比达到 40%以上。</p>

表 4-3 研发质量改进重点要求表（续表）

专项	目标	关键改进型	度量指标
测试过程	系统测试：所有交付的项目都经过测试，促进测试质量提升	<p>1、系统测试需转测申请，达到转测试标准方可转测。</p> <p>2、设计测试用例要按照公司定义测试用例模板规范，包含所有功能，跟需求关联^[42]。</p> <p>3、测试报告：测试完测试人员需按照公司定义测试报告模板填写，并输出测试报告，功能测试要给出测试质量等价评价（ABCD）和测试结论（测试是否通过或是否满足发布要求或是否能够发布）。</p> <p>4、缺陷记录和统计分析：测试发现缺陷录入缺陷平台并把缺陷关联需求或版本。测试完成之后，对于发现的缺陷要做分析。</p> <p>5、发布：对提交测试版本测试完成后，测试人员按照发布标准对该版本质量进行评价，并以此作为判断是否可发布的依据。具体评价标准请参见“测试质量评价标准”。</p> <p>质量管理总部：</p> <p>1、重新审视测试过程并强化测试过程执行：明确测试用例、测试报告（功能、性能、安全等）标准。</p> <p>2、质量管理总部负责对测试数据晾晒，推动分析改进。</p>	<p>1、提交测试打回次数或冒烟测试缺陷数；</p> <p>2、按版本维度统计测试用例个数和测试用例执行通过率（含自动化测试数据）；测试用例评审发现问题^[43]；</p> <p>3、遗留缺陷数；有效的缺陷数；缺陷解决率；有效缺陷致命+严重缺陷占比；</p> <p>4、已执行测试质量等级评价 A\B\C\D 项目数；测试质量等级评价 A+B 等级占比</p> <p>5、发布后的质量：（生产问题+事件）除内部测试发现缺陷占比。</p>

表 4-3 研发质量改进重点要求表（续表）

专项	目标	关键改进型	度量指标
测试 管理 平台	落地使用测试管理工具平台，并结合研发全流程（需求/缺陷/流水线/测试）等深入应用。	1、明确试点范围及计划：各试点团队明确具体的试点范围及试点时间计划	1、需求或测试或缺陷使用覆盖率 = 已使用平台需求或测试或缺陷模块的项目数除部门总项目数
		2、持续信息反馈同步：配合测试管理平台专项推动人员，做好试点过程中的需求问题反馈、解决方案探讨、进展情况收集反馈等工作	2、流水线使用频率 = 当月流水线构建次数除天数
		3、各试点团队输出试用总结，内容含但不限于：工具本身的可用性、便捷性、试用效果、建议等	3、流水线构建成功率 = 当月流水线构建成功次数除当月流水线构建总次数
		4、各模块使用基本要求	
		1) 需求模块成功与 JIRA 实现需求数据同步、流程能正常使用流转	
		2) 流水线遵守命名要求、流程建设要求、使用管理要求	
		3) 测试管理：RF、手工用例在测试管理平台管理，按计划测试，产出 RF、手工执行报告	

为了提升项目管理过程中每个迭代的质量，可针对过程中的每个环节增加评审，详情如下表 4-4 所示：

表 4-4 评审要求表

类型	时机	内容	参与人	产出
设计评审	迭代启动会、需求开发前	方案、概设、详设、用户界面设计等	评审专家组	设计评审报告,且项目经理确认问题解决
代码评审	迭代过程中单次或多次	源代码	评审专家组、开发人员	代码评审报告,且开发负责人、项目经理确认问题解决
用例评审	思维导图或用例编写完成后(时间充裕可以完整编写测试用例,若时间紧急仅需思维导图)	手工用例或思维导图、自动化测试用例	评审专家组、团队成员	思维导图或测试用例评审报告,且测试负责人确认问题解决
测试评审	功能测试完成后,研发项目迭代结束并不发版本情况下	需求测试过程及结果	评审专家组、团队成员	测试评审报告,且项目经理、测试负责人确认问题解决
发布评审	测试阶段结束进入发布状态	迭代中已完成的用户故事、任务与计划解决版本,对升级说明文档、操作手册、接口文档、测试报告、脚本、xml	评审专家组、团队成员	发布评审报告,且项目经理确认问题解决
回顾会议	测试版本发布后	对已完成的版本进行回顾,包括需求分析问题、开发质量问题、进度问题等都可以提出,针对问题提出解决方案,针对好的需要继续保持	团队成员	回顾会议报告,项目经理需对问题进行处理解决

4.2.5 敏捷开发基于 5C 原则的项目沟通方案设计

在项目的每个迭代启动会前,项目经理首先要对他在此次迭代版本中的需要实现的需求任务清单进行认真的确认,对其进行明确的分析,每个成员要清楚了解每个需求的目的以及如何实现。开发、测试人员依据需求文档仔细阅读,并提出需求中存在的问题,将问题汇总给对应的需求人员进行解答,并进行对应的需求调整。

项目经理在项目启动时,需与客户、对内明确联系的主题、方式、间隔时间等,在项目启动会时宣导给整个项目组相关组员。实施过程中,根据与客户的约定组织双方项目干系人召开项目会议,产出会议纪要。根据客户要求及内部管

理要求发布周报、日报。遇到项目问题或风险时，根据项目问题的严重级别，反馈给相对应的人员。

(1)、项目会议

①在项目计划阶段，项目经理需和客户沟通确定项目例会规范，确认例会召开时间、方式、地点、参与人员要求等，同时确定项目组内部例会规范。

②项目经理与客户的项目会议内容包括但不限于以下事项：汇报项目进展，讨论并决策项目中出现的缺陷、风险及处理措施，制定相关计划。项目会议结束后，项目组及时产出会议纪要。

③项目经理按照计划组织项目组内部项目例会，通报项目相关决策事项，同时了解项目组成员近期工作进展，安排新工作任务，产出会议纪要。

(2)、项目周报、日报

项目经理定期通过项目周报或日报形式给项目干系人汇报项目进展，揭示项目风险、问题及解决进展和状态。

①工程项目常规管理只需要定期发布项目周报，在项目重要时间节点或客户特殊要求下才发项目日报。

②项目经理每周发布项目周报给项目干系人即客户及公司相关人员，定期将项目情况汇报给工程管理组，以便工程管理组对项目作出及时、合理决策。

③项目周报、日报必须抄送给项目管理部

由于项目的特殊性，项目对接的系统较多，能够完全熟悉每个系统的人员基本没有，由于业务较多，很多业务不熟悉其业务设计逻辑，在遇到每个系统问题时，随机的询问对应系统的开发和测试，有时候他们忙没时间回复，有时他们也不知道原因就直接不回复，导致问题被搁置、盲目的咨询多个人员寻求问题答案浪费时间或者自己去他们系统操作熟悉业务需求，为了解决这一问题，特要求项目组以及各对接系统设立一个负责人，由项目负责人将项目组的问题汇总，然后针对对接不同系统的问题找对应的系统负责人进行问题的沟通。

第 5 章 基于敏捷开发 J 公司项目管理优化方案实施与保障措施

5.1 敏捷开发优化方案实施

5.1.1 应用敏捷开发方法的项目特点

敏捷方法用迭代周期短的优势应对需求的变更，需要反复试验的，为针对项目中的交流问题、需求频繁变更等提供了轻量级的项目管理和开发、维护的思路，它结合了之前开发方法的特点。

敏捷方法的整体目标是以人为主和欢迎变化，敏捷方法目的是成为适应变化的过程^[44]。

因此，敏捷软件开发项目管理适用于维护类（升级包）、研发类，其特点包括人数最好不要低于 5 人和不要超过 9 人，项目需求变更频繁，人员比较固定。

5.1.2 应用敏捷开发实施方案

本项目遵循“整体规划、风险可控、分步实施、平稳过渡”的建设原则，主要分为两个阶段：

第一阶段：项目初步探索阶段，投入需求、开发、测试、实施开展此项目，构建集中作业中心，确定业务模式、制定权限体系及业务规则，实现高频柜面业务的集中运营；

第二阶段：项目成熟期，项目团队成员继续跟进此项目，将账户、集中交易、两融、期权等系统菜单按照标准的业务流程及规则上线，最终实现“一套终端、一个账号、一次登录”全业务通办的目的；同时将临柜业务移动化，增强客户服务体验能力。

5.1.3 应用敏捷开发实施过程

5.1.3.1 项目立项

（1）敏捷项目团队组建

由产品经理确定敏捷团队的项目经理、开发、测试、实施、需求、专家成员，然后确定项目助理，并共同组成敏捷团队。团队组建后需明确各成员的职责范围，其中，由于该产品的特殊性，对接系统较多，需要确定与其他产品对接的责任人，便于后续系统问题找对应责任人沟通，敏捷团队组建后由项目助理申请开通团队成员的相关权限，比如 SVN、JIRA、测试管理平台等。

（2）敏捷项目立项

维护产品的敏捷线确定后，由项目管理部向质量管理部接口人报备，由项目助理在度量系统中创建相应的敏捷项目；研发项目按照总部研发管理办法完成项目立项工作，同时由项目管理部向质量管理部接口人报备，由项目助理在度量系统中创建相应的敏捷项目，可采用小型迭代形式完成待办列表、概设、详设、识别技术难点、人员协调等启动前的准备工作。

5.1.3.2 需求梳理

(1)维护需求列表

产品经理或者项目经理结合服务部收集到的需求相关信息，以及对客户已有的承诺等，梳理产品需求，在每次迭代启动前确定迭代的开发内容以及详细开发计划，确定产品待办列表中的需求优先级^[45]。

(2)产品需求分析

产品经理根据待办列表优先级由高到低进行版本规划，需涵盖常规升级包、一对一升级包以及补充包；

产品需求分析完成标准：1. 明确近期 1-2 个迭代的版本规划；2. 版本规划需涵盖已明确迭代周期内全部升级包需求；3. 产品需求分析需定义到接口级（包含出入参）、功能号及对应菜单，并填写 JIRA 升级说明、需求分析结果内容；

(3)迭代规划

产品经理或项目经理组织确定本轮迭代的需求清单及迭代周期，将需求加入 JIRA 的迭代清单中。需注意的是：规划的迭代内容可不限于 1 轮迭代，若条件允许，时间要求明确，支持同时规划下 N 轮迭代的内容；迭代周期建议最长不超过一个月，最佳周期为 2 周时间。

5.1.3.3 迭代需求分析会

(1)迭代需求分析会

由项目经理组织需求分析组针对迭代清单中需求进行讨论，详细了解需求背景、原因、内容、范围、验收标准等，以确保迭代内需求均清晰明确，并且确定迭代中需求的分类以及优先级，在此过程中项目经理有权剔除需求内容不明确、范围蔓延等的需求，最后产出迭代的清单。

(2)会后需求宣导

迭代清单确定好后，需要发邮件知会所有成员，他们接收到通知后关注 JIRA 查看本次迭代需求列表，熟悉了解每个需求的内容，如有疑问或意见可以备注在每个 JIRA 需求中，如果需要确认可以直接找对应的人员，JIRA 系统会自动发送邮件给对应的人，对应的人看到此邮件会在 JIRA 解答此问题，每个人看完需求后，开发组、测试组分别各自组织一次需求宣导会，解读每个需求并对需求提出

各自团队的意见和疑问，也将这些内容备注在 JIRA。最后项目助理协助项目经理组织一次需求宣导会，做需求第一次团队内部讲解，针对有疑问的需求做一次解答。

(3)需求分析完成标准

①需求分析包含四要素：客户、完成标准、产品意义、验收标准

- a.客户：为谁开发的系统；
- b.完成标准：实现该功能的标准；
- c.产品意义：做这个功能的收获；
- d.验收标准：团队目标基于客户需求希望，同时可进行细化和量化。

②需求完成了拆分、分类和优先级的定义，具体内容在JIRA中填写；

③需求完成了概要分析设计（设计程度由团队项目经理自行把控）；

④需求符合INVEST（I dependent(独立的)、N egotiable(便于沟通的)、V aluable(有价值的)、E stimable(可估计的)、S mall(短小)、T estable(可测试的)）原则^[46]。

(4)本阶段输出，如下表 5-1 所示：

表 5-1 迭代需求分析会输出内容表

输出项	责任人	备注
产品待办列表、迭代清单列表	产品经理、项目经理	在 JIRA 中登记
产品维护解答裁剪表（scrum）	项目助理	在公司质量白皮书取模板 （适用于研发项目）

5.1.3.4 迭代启动会

迭代启动会由项目经理（PO）发起并主持，项目助理组织协调，会议参与人员为：产品经理、项目经理、项目助理、团队成员、产品相关的服务、工程人员，迭代计划会议内容包括“做什么，怎么做，谁来做”^[47]。其目的是确保团队对本轮迭代清单的内容和验收有足够理解，团队共同评估迭代故事点和工作量，共同承诺迭代产品交付的时间和标准。

迭代启动会的流程如下图5-1所示：



图5-1 迭代启动会流程

（1）需求讲解

由项目经理清楚表达并讲解迭代清单需求的背景、内容，以及初步的修改方案，使大家都清楚该需求的来龙去脉，也可由对应的需求扩展到相应的业务讲解，最终与团队成员达成对需求的验收标准理解一致。

（2）故事点估算

当团队成员确认已经对需求完全了解、无任何重大问题时，大家开始对需求进行故事点估算。

估算方法：估算故事点相对值。团队共同选取出故事点为 1 的需求，作为评估故事点的标准，然后再将其他需求和这个标准进行比较，得出其他需求的相对点数，故事点可选点数为：1、2、3、5、8、13；

估算工具：敏捷估算扑克牌；

（3）需求认领

所有的迭代清单中的需求都评估了故事点之后，团队成员对需求进行认领^[48]。

（4）需求设计和工作量预估

需求认领完成后，相应的开发人员和测试人员对迭代清单中每个需求进行开发设计和测试设计，团队对需求修改方案、影响范围和验收标准达成共识；由团队成员评估完成该需求的开发测试工作量，单位为小时（即 JIRA 中的原始估算时间）。

（5）确定迭代计划

团队根据历史团队生产力和周期，与本轮迭代估算的工作量和周期进行对比，评估本轮迭代计划是否合理，是否需要计划的需求进行增减。团队共同确定本轮迭代版本的发布时间，以及质量保证所必需的评审活动的时间计划，例如：设计评审、代码评审、第一批需求转测、测试用例评审、回归测试、发布评审等时间，并由项目经理指定人员形成会议记录。

（6）确定迭代中进行需求数量

团队成员针对【每日测试中需求数】的最大值达成共识，过程中若处于测试中需求数超过最大值，则 JIRA 会标红报警，需团队及时作出调整决策。控制进行中需求数量可以暴露产品在开发过程中的问题和阻碍，帮助团队及时发现瓶颈。

(7) 迭代启动

项目助理开启 JIRA 上的迭代，迭代启动后，JIRA 上有对应的 Kanban，一般为横向 Kanban，从左到右为：已登记、已受理、已分配、开发中、开发完成待测试、测试中、已解决、已关闭列，如下图 5-2 所示：



图 5-2 迭代控制看板

此外，可以通过 JIRA 的 Scrum Board 的 Backlog，选择该迭代中的需求导出成 excel 列表（右键-在 excel 中查看-保存）。

(8) 本阶段输出，如下表 5-2 所示：

表 5-2 迭代启动会输出内容表

输出项	责任人	备注
项目迭代计划	项目助理	通过 JIRA 导出
迭代清单列表	项目经理	在 JIRA 中登记

5.1.3.5 迭代进行阶段

(1) 需求处理

团队开发人员对需求进行开发，开发完成后进行单元测试，单元测试实行交叉测试，并出具单元测试报告，并上传至 JIRA，单元测试通过后先代码自查，自查完成后组织代码评审，需参考代码编码规范以及需求逻辑点进行，代码评审问题较多修改后可组织再次评审；

团队测试人员编写测试用例，自动化测试用例，当项目时间紧迫时，可使用 XMIND 工具编写需求思维导图，用例或思修改点较多可适当组织二次评审，评审通过后方可执行测试用例；

团队测试人员依照用例对需求进行测试，发现的 bug 在测试管理平台记录跟踪，若有些问题影响范围较大，此次版本可以不修改，可在 JIRA 上创建需求列表，将测试管理平台中的缺陷备注清楚 JIRA 需求编号，并关闭此缺陷缺陷。此外，针对提交的缺陷，解决时间最好不要超过 24 小时，为了方便测试回归验证，开发人员需要在测试管理平台针对每个缺陷记录缺陷产生的问题原因、修改方

法、修改文件、单元测试点以及SVN提交版本号记录。

此外，测试人员在进行版本迭代过程中，需要对升级包进行回归测试、自动化测试、压力测试等，其中回归测试实行交叉测试，可以通过多个人的差异化思维发现bug。

(2)持续集成

为了软件开发处于正常运行可通过持续集成完成，使用持续集成能更快地交付软件^[49]，较早找到bug，减少解决问题的时间和成本，产品缺陷更少。

版本控制：产品代码、测试代码、数据库脚本、构建与部署脚本等所有用于创建、安装、运行和测试程序的内容都需提交到SVN中^[50]。

自动化构建：代码提交SVN程序库后，能通过命令行启动自动构建。若构建失败，需立即修复。

团队共识：持续集成需每个人都能以缓慢增长方式多次将代码提交到代码库，团队需达成共识：一旦持续集成环境发现问题，必须立即修复，优先级最高；构建失败后不要提交新的代码。

(3)每日沟通

项目助理组织项目经理、团队成员积极参加每日站立会议，认真反馈产品问题，站立会的原则有：参会人员站立开会，会议时间不超过15分钟，会议时间、地点固定，每人都必须简洁描述前一工作日的内容，当天的工作计划，当前工作中有什么困难需要解决。

项目的团队成员在JIRA中填写任务描述、分析、实现概要等，每日编写工作日志以及估算剩余工时，项目助理每日统计迭代清单、燃尽图、缺陷解决情况，从缺陷类型、根源、单个需求缺陷数等方面进行缺陷分析，及时发现迭代中的风险并揭示，通过使用甘比特图针对每个需求开发、测试完成百分比进行统计，最后发布敏捷项目日报。

(4)迭代质量保证，如下表5-3所示：

表 5-3 迭代质量保证表

评审类型	评审时机	评审内容	参与人	产出
设计评审	迭代启动会、需求开发前	开发方案、概设、详设、用户界面设计等	评审专家组	设计评审报告，且项目经理确认问题解决
代码评审	迭代过程中单次或多次	源代码	评审专家组、开发人员	代码评审报告，且开发负责人、项目经理确认问题解决
用例评审	思维导图或用例完成后	手工用例或思维导图、RF 用例	评审专家组、团队成员	思维导图或测试用例评审报告，且测试负责人确认问题解决
测试评审	功能测试完成后，研发项目迭代结束并不发版本情况下	需求测试过程及结果	评审专家组、团队成员	测试评审报告，且项目经理、测试负责人确认问题解决
发布评审	测试阶段结束进入发布状态	迭代中已完成的用户故事、任务与计划解决版本，对升级说明文档、操作手册、接口文档、测试报告、脚本、xml	评审专家组、团队成员	发布评审报告，且项目经理确认问题解决

(5)迭代发布

测试人员经 SVN 打包，并在测试环境进行升级验证，并检查程序相关配置文件，确保产品版本正确。同时检查产品文档，包含外围接口文档、测试用例、升级说明文档、操作手册、测试报告，以及代码评审报告、测试用例评审报告（确认评审问题是否已经全部解决并为关闭状态）；项目助理组织项目经理、开发人员、测试人员、实施人员召开发布评审会议，发布评审内容，需包含脚本、私有 xml 配置文件、JIRA 升级内容、升级说明、接口文档、测试报告、操作手册、未修复的缺陷。评审过程中工程实施人员需要针对升级说明以及升级内容的每一项都需要了解，以便后续生产部署不会出现问题。评审过程中由测试负责人记录发布评审问题，形成发布评审报告，具体到每个责任人，会后责任人根据发布评审问题进行修改，测试负责人进行检查确保问题全部关闭。最后，测试负责人通过 SVN 打包，放在 Release 指定目录下，发送邮件给领导审批，领导审批通过后，经公司项目管理部指定人员发放到公司网站供客户下载使用。

(6)本阶段输出，如下表 5-4 所示：

表 5-4 迭代进行阶段输出内容表

输出项	责任人	备注
迭代清单列表、燃尽图	项目助理	在 JIRA 中查看
敏捷日报	项目助理	每日邮件发送项目组成员
设计文档	开发人员	提交 SVN 对应需求目录
设计文档评审报告、单元测试报告、代码评审报告	开发人员	针对评审的问题修改对应的设计文档
测试用例或思维导图	测试人员	提交到 SVN 版本迭代目录
测试用例评审报告、测试报告、升级说明文档、操作说明、发布评审报告	测试负责人	提交到 SVN 版本迭代目录
升级包	测试负责人	包含代码、脚本、升级文档等

5.1.3.6 迭代回顾会

(1) 会前准备

项目助理在升级包发布后对迭代过程数据进行统计分析以及跟历史版本对比, 包含缺陷、版本质量、需求规模、人力投入、工期等维度; 在升级包发布后一周内组织升级包项目回顾会议, 并在会前发送迭代过程统计分析数据;

①缺陷分析:

项目组中缺陷分析主要是缺陷, 按照缺陷类型分为编码、概要/详细设计、脚本/配置、接口协议、历史遗留、业务/软件需求分析、用户指引类文档, 可根据编码占比统计每个版本的开发质量问题, 根据每个版本的不同缺陷级别来得到加权缺陷数, 以及严重级别以上缺陷占比进行统计分析质量是否有提升。

项目中将一般的加权数定义为 1, 致命的加权数定义为 10, 严重的加权数定义为 3, 轻微的加权数定义为 0.3, 其加权缺陷数计算方法如式 (5-1-a) 所示:

式 (5-1-a): $\text{加权缺陷数} = 10 \times \text{致命数} + 3 \times \text{严重数} + 1 \times \text{一般数} + 0.3 \times \text{轻微数}$ 。

②迭代趋势:

迭代趋势是分析各迭代版本的计划和实际需求规模、实际和计划工期、生产力、需求稳定度、缺陷密度等, 生产力差异的分析来提升生产率的效率; 通过对需求稳定度的分析用来准确合理的安排每个版本的需求; 通过对缺陷密度的分析可以对开发人员的开发质量进行分析, 并针对这些缺陷分析来提高产品的开发质量。

其中, 生产力、需求稳定度、缺陷密度的计算公式如式 (5-2-a)、式 (5-2-b)、式 (5-2-c) 所示:

式 (5-2-a): $\text{生产力} = \text{实际需求规模} / \text{实际投入工时 (人日)}$;

式 (5-2-b): $\text{需求稳定度} = 1 - (\text{计划需求规模} - \text{实际需求规模}) / \text{实际需求规模}$;

式 (5-2-c): $\text{缺陷密度} = \text{加权缺陷数} / \text{实际需求规模}$ 。

(2) 回顾会内容

迭代版本发布后，团队开展对已结束迭代的总结，回顾已结束的迭代开发效果和开发效率，根据度量结果分析已结束迭代的过程目标偏差，总结迭代的实践经验，针对好的工作规范要提倡并继续保持执行，针对不好的规范要指出并加以改进，以便在下一个迭代改进，来提高团队的生产率。

回顾会可结合缺陷分析、迭代趋势等内容一起总结，得出优化措施并按照优化措施严格执行。下一轮迭代的回顾会中需检验之前迭代问题的完成情况，已完成可关闭，未完成备注并继续执行。

(3) 会后工作

①项目助理通过邮件方式发送给项目组成员，内容包括回顾会会议纪要，下轮迭代跟踪改进措施执行情况；

②检查迭代所有交附件是否已归档上传到 SVN，未上传的需要及时上传；

项目助理需要在升级包发布一周内完成度量数据提交，并收集总用例数、自动化用例数、代码行数、计划需求规模、需求规模变化量（新增+减少需求规模）等度量数据，提交至度量管理系统。

(4) 本阶段输出，如下表 5-5 所示：

表 5-5 迭代回顾会输出内容表

输出项	责任人	备注
项目敏捷数据统计及度量	项目助理	包括各版本透视图、缺陷分析、迭代趋势等数据
项目回顾报告	项目助理	包括迭代的总结,迭代问题以及解决方案

5.1.4 应用敏捷开发优化预期效果

在敏捷开发项目管理优化方案得到改进后，针对 J 公司存在的组织架构、沟通、进度、成本、质量的问题优化预期效果如下表 5-6 所示：

表 5-6 敏捷优化方案预期效果表

	存在的问题	优化后实施预期效果
组织机构	项目组缺失内部实施人员	增加了项目实施人员后，实施人员专心学习项目系统，了解项目产品的所有功能，积极解决项目中的环境问题，现场客户问题，从而减少了开发、测试重现问题的时间。
项目沟通	增加对接项目负责人	每个对接项目增加对接人，将问题收集汇总后发给对应的负责人，负责人针对这些问题一一解答，减少了项目中盲目咨询人员的沟通时间，也大大降低了项目的沟通成本
项目进度	对接项目问题要求对接负责人参会 使用项目进度度量工具	针对每次会议中存在的问题在会前提出，并要求相应的项目对接人参会，解答这些疑点。 同步增加使用甘比特图，准确的反应了项目的进度，精准的判断项目是否存在延期，提前做好风险把控。
项目成本	减少项目助理	项目助理减少后，2个及以上的项目使用同一个项目助理，这样针对项目间存在的问题也能准确找对应的人沟通，减少了项目间的沟通成本，并且人员的减少也降低了项目的成本。
	增加设计评审环节，减少需求返工	增加开发设计文档的评审，可以提前了解设计逻辑，并且能够了解设计逻辑是否与系统开发规范一致，提前预防了返工的风险。
项目质量	加强用例评审，提高测试质量	增加测试用例的评审，提高了需求用例覆盖度，大大的提示了测试质量。
	工程维护对接测试环境	测试人员不维护测试环境，这样大大增加了测试时间，可以有更充分的时间去测试需求，提高质量。
	增加回顾会，持续提升质量	项目组成员针对已完成迭代回顾，匿名指出不足，加以改进，得到成长，提升质量。

5.2 敏捷开发优化方案保障措施

5.2.1 组织架构调整

正如上文中提到的优化了J公司现有的公司组织架构，要优化是一件伤筋动骨的大事，适合的架构能带来更大收益，调整不好将会影响公司的收益，因此，必须要严格的分析架构调整带来的影响和风险，这样才能更好的整合资源调整组织架构。将工程实施人员部门拆散到每个具体的产品中，这样保证了每个产品有对应的熟悉的工程实施人员，可以专门负责该项目的实施工作，也不用因为借调人员而苦恼，也帮助了项目组的开发、测试人员节省了很多客户答疑的时间以及测试环境维护的时间，这样就有更多的时间去开发和测试，大大提升了开发、测试的效率，可以恰当的运作现有资源，为项目成长提供有利空间，优化能够顺利的实践与公司上级领导的大力支持密切相关。

5.2.2 制度规范与绩效相结合

团队成员一旦习惯了现有的工作模式后，如果再对现有模式进行优化，增加一些重要环节，将会大大的增加他们的工作量，需要挤出更多的时间编写各种文档以及参加各种会议，那么他们将会抗拒，不愿接受新的敏捷流程模式，理应产生严重的排斥心理。因此，如果需要对团队成员接受新的流程模式，那么就要建立一些规章制度，并让大家都有时间慢慢接受变化的适应过程。

流程优化的具体实施关系到公司产品的是否能够顺利上线并保证质量，公司需要制定相应的规范制度，列出每一项的检查内容，针对每一项提出具体的要求并设置相应的分值，不符合要求的需要说明并扣除相应的分值等，最后对所有的分值汇总得出各版本迭代的规范度，并将此规范度纳入绩效考核，针对做的好的项目给予嘉奖，针对不好的批判改正，具体规范制度如下表 5-7 所示：

表 5-7 敏捷开发优化方案制度规范保障措施表

XXX 产品维护阶段裁剪表及检查单									
升级包未扣分前总得分					50				
扣分总数									
升级包规范度									
检查日期									
					备注				
编号	节点	活动	检查内容	裁剪指南	活动是否裁剪	标准分	扣分	不符合项	
1	单个升级包规划	计划	未在 jira 上进行升级包的计划	不可裁剪	否	2	0		
		需求分析和管	维护阶段的缺陷和新需求未全部采用 JIRA 进行管理	不可裁剪	否	4	0		
2	升级包制作	需求评审	未填需求说明书评审报告；报告内容不完整或评审问题未解决。	不可裁剪	否	2	0		
		需求跟踪	需求未与对应的测试用例关联，需求状态未及时刷新	不可裁剪	否	2	0		

表 5-7 敏捷开发优化方案制度规范保障措施表（续表）

XXX 产品维护阶段裁剪表及检查单							
3	设计以及评审	未对客户提交大的需求进行相关的流程设计并提交文档	可裁剪	是	2	0	
4	程序修改	程序修改完后未对每个需求/缺陷描述清楚	不可裁剪	否	2	0	
	单元测试	未按照流程规定的方式以及要求填写单元测试结果	不可裁剪	否	2	0	
5	代码评审	项目负责人未进行代码走查规范检查	不可裁剪	否	3	0	
	转测试	未在 JIRA 上修改成待状态	不可裁剪	否	2	0	
	升级包制作	新需求未设计测试用例	不可裁剪	否	4	0	
		未开展安全测试，未提交测试报告	可裁剪	是	2	0	
6		未提交缺陷到缺陷平台	不可裁剪	否	3	0	
		未输出测试报告	不可裁剪	否	3	0	
	配置管理	测试负责人未对测试用例进行评审	不可裁剪	否	3	0	
7		升级包发布当天配置项文档未归档配置库	不可裁剪	否	2	0	
	发布评审	升级程序发布前未召集相关人员进行发布评审（评审项包括：测试报告、升级说明书、程序、脚本、遗留缺陷等）	不可裁剪	否	2	0	
8							
	升级包发布	基线	不可裁剪	否	3	0	
	发布审批	未对需要发布的内容进行基线	不可裁剪	否	2	0	
9		升级程序发布前未填写“升级程序发布审批表”	不可裁剪	否	2	0	
	发布通知	已审核的程序未由需求管理员通过专用邮箱统一发布	不可裁剪	否	3	0	
10	需求关闭	需求确认与关闭	不可裁剪	否	2	0	

每个团队的成员都需要积极参与并按照每一项要求严格执行，针对流程的优化而导致工作量的增加肯定会引起员工的抵触，因此需要针对这些变化适当放宽需求、开发、测试时间，并酌情拉长项目周期，并且有效的对团队成员进行沟通和培训，降低其抵触心理并慢慢习惯新的流程模式；

5.2.3 公司高层的支持

针对现有已经形成固化思维的流程以及组织架构去做改变，一定会遇到很多困难以及团队成员的排斥，因此就需要公司高层领导们的大力支持。正如文中的保障措施，没有公司高层领导的鼎力支持，那可能就没有办法进行有效的实施，尤其是公司组织架构的变更，组织架构的变更关系到公司的发展，更加需要高层领导们针对变更做各种讨论沟通，以及变更会不会带来不好的影响等，因此，就必须要有公司高层的支持。

第 6 章 结论与展望

6.1 研究结论

本文将 J 公司的运营产品事业部的综合运营产品项目作为研究对象, 首先介绍了该产品项目的发展背景、目标以及目前该产品的敏捷软件开发项目管理现状, 然后阐述了论文国内外现状以及研究思路和内容等, 将其作为论文的基本信息参考。

然后通过介绍 J 公司的概况、组织结构、软件项目管理流程, 以及公司现阶段软件项目管理的现状, 并对影响公司软件项目管理的因素进行分析, 总结出了现有敏捷软件开发项目管理存在架构、沟通、进度、成本、质量管理的问题;

接着, 针对 J 公司现有的敏捷项目管理方案遇到的实际问题, 通过对现有模式进行深入的探究和研讨, 并结合敏捷开发项目管理模型以及敏捷开发重要的思路进行融会贯通, 增加固定的工程人员; 基于 5C 的沟通管理原则固定各系统对接人以及明确联系细则规范; 基于紧前关系绘制图的进度管理原则增加进度管理工具, 晨会汇报各自进度和遇到的困难; 基于挣值管理的成本管理原则减少非必要的资源, 增加必要资源做到物尽其用, 减少需求返工导致的时间成本增加, 增加评审前的互查避免评审过程中出现低级错误浪费参与人时间; 基于持续增量改进的质量管理原则设计质量改进要求表严格执行, 增加评审环节问题责任到人, 产出对应报告。再此基础上提出优化的设计实施方案和实施过程, 让敏捷开发体系能够淋漓尽致的体现, 以此来进一步提升公司的敏捷项目管理方案, 降低公司的成本, 提升产品质量能够抢占市场以此来提升客户的信赖。

最后, 通过后续具体实施检验优化方案的预期效果, 针对方案的有效实施也配套提供了组织架构调整、制度规范与绩效相结合、公司高层大力支持的保障措施, 来保障新的优化方案能够顺利开展实施, 并通过优化的预期效果验证其是有效的, 具有可行性的。

在当今科技快速发展的时代, 敏捷开发项目管理是目前阶段软件开发运用最为广泛的模型, 除了要求团队成员有良好的与人交流能力外, 还需要理解能力强, 有责任心, 团队协作能力强等, 这样才能够快速提升项目组成员的工作能力和经验。身处软件行业, 就应该知道软件项目管理是一个持续不断优化改进的过程, 我们都要持续改善现有的项目管理方案, 找到适合自己产品以及公司的项目管理方法, 降低项目的成本, 提升产品的质量, 为客户提供高品质的产品, 赢取客户的信任, 抢占市场份额, 为公司实现利益最大化。

6.2 研究展望

本文探讨对象仅以 J 公司金融软件系统为例，作为公司的重点项目，引起了公司大领导的重视，以及客户的大力支持，但是由于前期项目管理问题，导致项目没有达到预期效果，通过对项目管理的优化，使其得到了一定的提升，但是由于时间的限制，本文的研究因此还存在着一些缺陷，比如：（1）研究对象单一，本文针对运营产品事业部的产品敏捷项目管理进行了优化，收集的信息也是该部门的信息，但是公司的软件产品非常多，产品的客户群体以及项目成员能力各不相同，如果要将此优化方案推广给公司各个产品使用，需要有更多的研究对象。

（2）敏捷模式多样化，还需要更加深化探究，不确定可以用其他更优模式来替代。

由于本文中优化的敏捷项目管理方案单一，不一定适用于公司的所有产品项目，若其他产品项目要使用该项目管理方案，则需要根据其实际情况进行考量，此方案可作为一个参考模板，其他产品项目要想有符合自己管理方式可以在此基础上延伸或变更。虽然还有部分研究缺陷需要进行优化，但是对于公司其他项目管理不完善或未使用敏捷开发的团队如何建立高效的、适合自己项目组的项目管理方法还是有很多值得参考学习的地方，希望能够通过这些方法能够帮助他们提升软件产品的质量。

J 公司的敏捷开发软件项目管理优化方案从实践出发，结果实际问题分析，得出结论最终应用于实践中，具有可行性，为公司其他的产品项目优化敏捷开发项目管理提供参考模板，具有实际的指导意义。但是在实际的敏捷项目管理过程中肯定还有一些更好的项目管理方法，本文的研究希望有更多人提出指导建议。此外，由于本人知识面的局限性，欢迎更多的人指出本文有问题的地方，并积极改正。

参考文献

- [1]杜海颖. 一种基于敏捷开发的软件项目管理方法与实践[A]. Information Engineering Research Institute,USA、Asia Pacific Human-Computer Interaction Research Center,Hong Kong.Proceedings of 2018 4th International Conference on Mechatronic Systems and Materials Application(ICMSMA 2018)[C].Information Engineering Research Institute,USA 、Asia Pacific Human-Computer Interaction Research Center,Hong Kong:智能信息技术应用学会,2018:5.
- [2]朱晓赞,许民利.项目管理研究综述[J].价值工程,2008(11):128-131.
- [3]王怀民,吴文峻,毛新军,丁博,郭长国,李未.复杂软件系统的成长性构造与适应性演化[J].中国科学:信息科学,2014,44(06):743-761.
- [4]李红艳. 计算机软件开发项目管理阶段应遵循的原则 [J]. 信息通信,2014(11):167-168.
- [5]陈文楷.软件工程发展现状与趋势综述[J].信息记录材料,2018,19(06):6-8.
- [6]A. Garg,K. Tai,M.M. Savalani. State-of-the-art in empirical modelling of rapid prototyping processes[J]. Rapid Prototyping Journal,2014,20(2).
- [7]施秉东.谈计算机软件开发中的敏捷方法[J].通讯世界,2016(03):229.
- [8]靳萍,谷秀洁.敏捷项目管理在高校图书馆的应用研究——以西安工程大学图书馆微信建设为例[J].山东图书馆学刊,2019(01):82-86.
- [9]KIM G, HUMBLE J, DEBOIS P, et al. Dev Ops 实践指南 [M] .刘征, 王磊, 马博文, 等译. 北京: 人民邮电出版社, 2018.
- [10]胡盛行. 敏捷软件开发在非软件研发企业的应用探索与实践[A]. 中国科学技术协会、吉林省人民政府.第十九届中国科协年会——分 4 信息新技术 东北新工业论坛论文集[C].中国科学技术协会、吉林省人民政府:中国科学技术协会学会学术部,2017:7.
- [11]Jan Bosch,Petra M. Bosch Sijtsema. Introducing agile customerâ centered development in a legacy software product line[J]. Software: Practice and Experience,2011,41(8).
- [12]季晓宇.软件工程的历史与发展趋势研究[J].网友世界,2014(15):22.
- [13]Ahmet Efe,Nisanur Mühürdaroglu. Secure Software Development in Agile Development Processes of E-Government Applications[J]. Journal of International Scientific Researches,2018,3(1).

- [14]高稳稳. 基于定制化的 Android 自动化测试工具的设计与实现[D].中国科学院大学(工程管理与信息技术学院),2014.
- [15]毕猛. 软件项目管理现状分析和对策[J]. 计算机光盘软件与应用,2014,17(21):71+73.
- [16]苏莉. 软件项目管理现状与对策分析[J]. 中国战略新兴产业,2017(48):101.
- [17]刘恩泽,谭欣杰. 中国软件项目管理现状分析[J]. 科技资讯,2019,17(16):253-256.
- [18]闫帅,许鹏翔. 基于瀑布模型与敏捷开发相结合的项目管理方法探讨[J]. 电子技术与软件工程,2013(18):67.
- [19]田莉. 论如何借鉴敏捷软件开发方法来改进软件项目管理[A]. 天津市电子学会、天津市仪器仪表学会. 第三十二届中国(天津)2018'IT、网络、信息技术、电子、仪器仪表创新学术会议论文集[C]. 天津市电子学会、天津市仪器仪表学会:天津市电子学会,2018:4.
- [20]梁瑾. 敏捷软件开发项目的快速启动方法[J]. 电子技术与软件工程,2017(01):52-53.
- [21]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.
- [22]丁志平. 敏捷软件开发中的风险管理[J]. 信息与电脑(理论版),2010(02):76+78.
- [23]David Notkin. Software, Software Engineering and Software Engineering Research: Some Unconventional Thoughts[J]. Journal of Computer Science & Technology,2009,24(02):189-197.
- [24]张彦. 数据治理背景下我国公共部门项目管理模型设计[J]. 中国管理信息化,2020,23(13):189-193.
- [25]张运嵩,蒋建峰. 从 PMBOK 第六版看项目管理发展新趋势[J]. 价值工程,2020,39(06):7-9.
- [26]马丁·塞德米耶,左剑. 全球视野下项目管理持续发展的十大理由[J]. 项目管理评论,2017(02):14-19+7.
- [27]曾忠诚,李壮相,江志炎,张全地. 敏捷项目管理方法在云化的软件开发架构中的应用探讨[J]. 计算机产品与流通,2018(01):22+39.
- [28]Niemi-Grundström, Minna. Developing, evaluating and managing library with agile methods[J]. Library Management,2014,35(6/7):
- [29]顾家铭. 基于敏捷开发的物联网工程实践教学探讨[J]. 科教导刊(中旬刊),2020(05):105-107.)
- [30]樊亮,阚青. 基于信息化的软件开发项目管理[J]. 项目管理技术,2018,16(11):96-99.

- [31]潘立琼,朱学颖,余久久,肖刚.基于 Scrum 敏捷方法自主学习系统的设计[J].黑河学院学报,2020,11(01):215-217.
- [32]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.
- [33]刘倩.敏捷开发与瀑布开发相结合的方法研究及优势分析[J].南方农机,2017,48(19):89+91.
- [34]崔硕.敏捷开发在北京市疫情决策分析平台项目管理中的应用[J].信息技术与信息化,2021(01):91-92+95.
- [35]肖耀涛.基于敏捷方法的软件研发项目管理系统的设计与实现[J].信息与电脑(理论版),2020,32(04):83-85.
- [36]王秋实.对基于信息化的软件开发项目管理分析[J].计算机产品与流通,2020(02):25.
- [37]王磊.软件项目开发模型的优劣势分析[J].科技传播,2017,9(22):104-105.
- [38]芮斌.信息化背景下软件开发项目管理[J].电子技术与软件工程,2019(05):42.
- [39]Hai Yang. Research on Improved Staged Software Cost Estimation Method Based on COCOMO Model[J]. Advanced Materials Research,2014,3326.
- [40]Mayanka Sharma,Iqbal Kotwal. A Concept Note on Execution and Impact of Agile Software Project Management Methods in Midsized it Product Development Companies[J]. We'Ken- International Journal of Basic and Applied Sciences,2016,1(3).
- [41]梁永幸.浅谈敏捷开发与其他传统开发方式的区别[J].电子世界,2012(24):128.
- [42]陈翔,顾庆,陈道蓄,蒋峥峥.回归测试中测试用例集缩减问题的研究[J].计算机科学,2014,41(09):196-204.
- [43]王宇,韩苗.基于信息系统的软件项目管理[J].中国新通信,2021,23(05):78-79.
- [44]Barry Boehm.Get ready for agile methods , with Care[J].IEEE Computer,2002,35(1):64-69.
- [45]刘卉.敏捷项目管理在遥感信息系统开发中的实践应用[J].中小企业管理与科技(中旬刊),2021(02):166-168.
- [46]Kenneth S.Rubin , A Practical Guide to the Most Popular Agile Process[M] , Addison-Wesley Professional,2012:89-93
- [47]李志凌,王先科.基于 PDCA 原理的京东敏捷软件项目管理研究[J].项目管理技术,2018,16(05):60-64.
- [48]刘华清.敏捷开发中进度管理的策略[J].数字通信,2013,40(04):78-80+94.
- [49]商永巧.DevOps 之持续集成实践之路[J].通讯世界,2018,25(12):266-267.

- [50] 段琳, 孙恒一. 基于 Bamboo 的持续集成系统[J]. 电子技术与软件工程, 2016(20):74-76.

附录：敏捷开发软件项目管理优化问卷调查

敏捷开发软件项目管理优化问卷调查

大家好，针对金融软件行业来说，软件项目经常会出现需求变更以及三司三所出紧急上线新规的情况，现有的敏捷软件开发未能够按时的完成产品的交付，需要对其进行优化。为了解决这一问题，设计了此问卷，请各位真实的作答。

1、您是否在项目组工作超过半年时间？【单选题】

- ☐ 是 ☐ 否

2、您是否觉得现有的敏捷项目管理还需优化？【单选题】

- ☐ 有必要 ☐ 没有必要，项目管理非常合理

3、您是否完全了解敏捷开发流程【单选题】

- ☐ 完全不了解（请跳转至问卷末位，提交答案）
☐ 了解一些，但是不全面
☐ 很熟悉，并能熟练掌握其流程

4、您认为影响现有敏捷项目管理的因素有哪些？【多选题】

- ☐ 公司组织架构问题
☐ 项目组成员间沟通问题
☐ 项目进度管理问题
☐ 项目成本控制问题
☐ 项目需求变更频繁问题
☐ 项目需求分析问题
☐ 项目人力资源问题

5、敏捷软件开发是一种采用迭代管理的软件项目管理方法，您认为采用这种方法影响公司组织结构问题的原因是什么？【多选题】

- ☐ 部门实施人员缺失
☐ 其他，_____

6、您认为采用这种方法影响项目沟通问题的原因是什么？【多选题】

- ☐ 对接项目较多，没有具体的对接人
- ☐ 项目助理间问题沟通不顺畅
- ☐ 其他，_____

7、 您认为采用这种方法影响项目进度管理问题的原因是什么？【多选题】

- ☐ 进度管理工具的缺失
- ☐ 其他，_____

8、 您认为采用这种方法影响项目成本管理问题的原因是什么？【多选题】

- ☐ 人员结构不合理
- ☐ 需求分析不明确导致返工
- ☐ 其他，_____

9、 您认为采用这种方法影响项目质量管理问题的原因是什么？【多选题】

- ☐ 开发代码提交不规范
- ☐ 测试用例缺失或不评审，用例覆盖度不高
- ☐ 需求返工压缩工期，测试不充分
- ☐ 其他，_____

10、 您认为通过哪些方法可以优化现有敏捷项目管理？【多选题】

- ☐ 增加固定项目实施人员
- ☐ 调整人力资源
- ☐ 增加设计评审
- ☐ 增加开发代码规范评审
- ☐ 增加测试用例评审
- ☐ 增加版本回归会议
- ☐ 其他，_____

11、 您所在的项目？【单选题】

- ☐ YYCP 一部
- ☐ YYCP 二部
- ☐ YYCP 三部
- ☐ YYCP 四部

12、 您的名字? 【填空题】

13、 您的岗位? 【单选题】

- ☐ 项目经理
- ☐ 需求人员
- ☐ 开发人员
- ☐ 测试人员
- ☐ 工程人员