[摘要](#摘要)  
 [主题词](#主题词)  
 [第一章 选题依据](#第一章-选题依据)  
 [1.1 选题背景](#X8b11ab26b29137cef5c63f4a455f0260f261b0d)  
 [1.2 工程设计的价值和意义](#X636130287a4fd4298f44567562201fb0c79da1d)  
 [1.3 国内外应用现状和发展趋势](#X8d454b654873a63e705b0b74ca80b7b12bc07a7)  
 [国内公司交付方式](#国内公司交付方式)  
 [敏捷开发国内研究现状](#敏捷开发国内研究现状)  
 [敏捷开发国外研究现状](#敏捷开发国外研究现状)  
 [1.4 选题的先进性和实用性](#X476a60bc7487fdf91d38ed5799b3c2a9ac88014)  
 [参考文献](#参考文献)  
 [第二章 课题内容](#第二章-课题内容)  
 [2.1课题内容](#X7a1b2775970df7f918df8136d2c697a3f2b6ea5)  
 [2.2系统需求分析](#X6b1279f99255f3e619c9a05df619455d0a92511)  
 [2.2.1 概述](#Xb4e4d4292d8b02e1d724a6cc79afd1af78f2740)  
 [2.2.2 用例图描述](#Xec896f7eb8883200982a1fcfefbb5b798f60d45)  
 [3 概要设计](#X1e6333eb87792218c550deaac7fd5b24ae3b73f)  
 [3.1 目标与概述](#Xa84d57a2d72b6a1320f0532eeb8671de36f187a)  
 [3.2 总体架构设计](#X5ba4c3daca77cda728fa303061f84511c189db0)  
 [3.2.1 总体流程](#X789c98777045a7245b308897c310195c73e5281)  
 [3.3功能模块设计](#X4973b03e193f0b8273f391fe3b24b8ce0e8cdea)  
 [3.4.1 需求评审模块设计](#X9ddcb6c0e100bb5974a6ee295c99188b8a3071a)  
 [3.4.2 按需集成模块设计](#X1e5ec973fc2b80ff433f93050d7fcb208a62e3f)  
 [3.4.2 按需测试模块设计](#Xea880116d128668a5ee4be648d64015c1b57628)  
 [拟采用的开发方法，环境，测试方案等](#拟采用的开发方法环境测试方案等)  
 [技术难度及特色分析](#技术难度及特色分析)  
 [本人主要工作描述](#本人主要工作描述)  
 [工作进度的大致安排](#工作进度的大致安排)  
 [预期成果](#预期成果)

### 摘要

当今互联网行业产品迭代更新迅速，业务发展迅速，如何更快更好的交付产品和满足日益增长的业务需求成为首要解决的问题。以往固定的持续交付模式需要做出一些突破。敏捷开发模式打破固有的交付模式，可以持续完成业务需求和交付产品。但是仅仅改变交付模式远远不能满足高速增长的业务需求。

因此，为解决上述问题，本文将设计并实现一站式敏捷协同平台，该平台以敏捷开发模式为基础，通过节点消息触达实现多角色的研发协同，作为枢纽打通上下游内部开发平台，以低成本的方式实现研发链路的自动化、规范化、标准化。核心功能包括需求评审、按需集成和按需测试。

该平台采用前后端分离方式，前端使用React渐近式框架构建用户交互页面，后端使用Egg.js框架来为前端服务，使用MySQL进行数据存储。

本系统应用成功后，将优化现有研发交付链路中的每一点一滴的瓶颈，且贯穿整个研发链路多个环节，支撑产研环节中以需求为事件源包括需求演进、技术研发、效率提升、质量度量等多维度持续追踪，实现缩短产品交付周期、提升组织效能的目标。

### 主题词

中文：敏捷 协同 一站式

英文：agility synergy one-stop

### 第一章 选题依据

#### 1.1 选题背景

过去几年，外卖产研团队已沉淀大量业务规范和技术能力，有效的支撑了外卖业务快速发展。在2020年8月8日已突破4千万单，20年中战略会领导提出了“2025年外卖日均1亿单”目标。未来5年日均1亿单的目标拆分背后强依赖于技术创新、效率提升、流程标准化以及基础架构建设等多个方面。业务目标的进一步提高也会对现阶段的持续交付能力提出了更高的要求，为支撑未来5年外卖日均1亿订单，持续交付系统需要积极主动做一些突破和改进。

终端的持续交付指的是快速将产品经理的需求细化，通过持续的、自动化的流程和技术方案保障，安全、快速的将产品交付到用户终端。按照涉及事务的不同类型，外卖业务需求研发交付过程分为需求管理、需求开发、构建集成、需求测试、灰度发布、线上运维六个阶段。和纯粹的开发工作不同，需求持续交付流程往往对应着复杂的多角色协同问题，主要体现在以下几个方面：

* 流程规范执行不到位，导致沟通成本和业务质量风险较高。
* 人与wiki间触达成本高，信息变动“口口相传”。
* 缺少过程数据记录，问题不可追溯，不利于管理和优化。

现在外卖内部的交付模式是双周迭代交付模式，贯彻的总体原则为：评审、开发、测试完全并行，以两周为固定周期，以需求维度持续交付。但双周迭代存在一些问题：

* 从评审到交付，发布周期还是太长，产品快速上线的诉求还不能得到很好的满足
* 对于特定需求和小需求不是很友好

为了能够更快更好的交付产品和满足日益增长的业务需求，必须改变现有的交付模式,敏捷开发模式就是一个很好的选择。

敏捷开发是轻量型的开发方式，反对传统的、庞大的、重型过程。它提 倡与人交流的重要性，用高质量的软件代替文档，能够快速适应迅速变化的需求 等快速开发软件的能力，他的核心理念是：寻求速度与质量之间的最佳平衡，应对千变万化的 业务需求，始终确保为客户提供最大价值，使得软件真正为业务提供价值最终能 够彻底的改变软件技术为用户业务提供价值的方式[2]

基于以上背景，通过建设一站式敏捷协同平台，打通产品研发交付多个环节标准化、自动化的、数字化、可视化的能力，覆盖需求从生产、集成、部署、测试、上线的全链路，优化现有链路中的一点一滴的瓶颈。

#### 1.2 工程设计的价值和意义

持续交付系统的突破和改进需要产研协同线上化的推动。

产研协同线上化是对过往终端开发、测试、运维能力体系建设的一次升级，通过产研协同线上化，由目前的单一角色、单个节点、非持续、半自动的系统最终演进为多角色参与、多维度、持续、自动化，安全快速的持续交付体系。产研协同线上化解决的不是某一个单点技术或流程问题，而是要优化现有研发交付链路中的每一点一滴的瓶颈，且贯穿整个研发链路多个环节，支撑产研环节中以需求为事件源包括需求演进、技术研发、效率提升、质量度量等多维度持续追踪。终端持续交付解决的不是某一个单点技术或流程问题，而是要优化现有研发交付链路中的每一点一滴的瓶颈。

持续交付是对过往终端开发、测试、运维能力体系建设的一次升级，需要对标业界解决方案，结合外卖业务阶段和终端的技术特点，实现**缩短产品交付周期**、**提升组织效能**的目标。

#### 1.3 国内外应用现状和发展趋势

##### 国内公司交付方式

国内主要参考了现阶段其他公司移动端持续交付的现状。今日头条采用按周发版的交付方式，其中Android采用单周，iOS采用的是双周；滴滴采用的也是双周发版的交付方式；淘宝Android发版交付的方式是一个月两次，iOS是一个月一次；支付宝发版交付方式是平均4周一版；而和外卖业务有竞争关系的饿了么则采用的是双周发版，但是支持临时小版本。可见，目前国内部分公司仍然采用的是瀑布模型的软件开发模式。

介绍完国内各个公司的发版交付方式，再介绍一下敏捷开发的国内外研究现状。

##### 敏捷开发国内研究现状

我国的质量管理发展较晚。当敏捷已经成为一半多国外IT项目的开发方式的时候，国内才刚刚开始尝试敏捷开发。这和国内企业时代背景有很大关系。在对Scrum敏捷方法的研究中，国内的杨帆提出敏捷开发不仅仅是一种开发流程，而足一种方法论，如果仅单纯的学习该开发流程，是无法达到预期效果的。党源源等人通过研究提出，这一敏捷开发之所以可以高效的运行是因为其最大限度地将开发者和用户、计划目标和实际实施情况、投资和投资回报间地距离进行缩短。随着软件行业全球化交流的深入，我国在软件项目管理中对于敏捷开发的使用也越来越多，在了解和学习敏捷开发理论方而的程度逐渐深入，并在我国很多优质的公司中也广泛使用，例如京东的京麦团队。该团队首次使用了敏捷开发，最早在2012年年底引入了Scrum敏捷开发，在后续３年的实践过程中尽力缩短开发周期，并且在线用户数以及活跃用户数也品著提高。与此同时该团队实现了多个敏捷团队相互协作，并且将该敏捷方法推广到很多大中型团队中，为类似的开发团队提供了很好的参考。[3]

##### 敏捷开发国外研究现状

企业在产品或服务的质量上精益求精的态度是其保持可持续发展的关键，相比国内来说，国外先进国家对产品、服务的质量要求更高。在近百年的质量探索实践中，国外发达国家积累了大量优质经验并设立了健全的质量管理体系。

1987年，国际标准化组织（ISO）正式颁布ISO9000系列国际标推，它是在提炼和总结了全球经济发达国家企业质量管理的实践经验基础上制定而成，具备很强的指导性和通用性。通过在实际运行工作中对ISO9000的改进，７年后提出９４版系列，但是此标准的修改更侧重于制造业中保证产品的质量，没有对满足用户耑求提出标准；二十世纪后期在企球经济的推动下，ISO标准也高速变化迎来2000版问世，该版本改善了94版的缺陷且通用性增强，同时强调了对顾客满意度要求的重视，并通过引入PDCA戴明环理论指出质量持续改进的要求，更符合现代企业认证要求。

20世纪70年代，Ｗinston Ｗ.Royce首次在IEEE WcstCorn会议上提出了瀑布模型的概念。在这之后的几十年，瀑布模型一直被各个公司采用来生产软件产品。20世纪90年代，KenSchwaber和Jeff Sutherland提出了Scrum敏捷升发方法并且在OOPSLA工作坊中进行了初步应用。1996年，克莱斯勒支付平台开始应用极限编程。到了21世纪，整个软件界也才开始逐步认识到瀑布模型是有缺陷的，因为爆布模型本身太完美，而实际的工程实践却不可能如此完美。２００１年，由MartinFowlerdun在内的１７人共同成立敏捷联盟。敏捷理论起源于国外，国外很多大公司采用敏捷开发模式很多年，已经有很成熟的敏捷开发管理经验。近些年由传统开发模式转型敏捷开发的企业越来越多，敏捷己经不单单应用于软件行业，甚至在其他行业也有渗透，据统计在美国已经有将近一半的项目在使用敏捷的方法。[3]

#### 1.4 选题的先进性和实用性

相较于其他敏捷平台只关注于需求排期的问题，本平台不仅考虑到了需求排期问题，还考虑到了研发链路中的问题，将需求分配，打包，测试等功能集中到本平台上，打通上下游的内部平台，自动化研发链路中一切可以自动化的环节，大大缩短了交付产品周期，提高了RD开发效率，具有一定的先进性。

本平台集成了需求分配，一键打包，一键提测功能，协同研发流程中的多角色，PM能够在本平台进行需求录入、三方评审；RD能够在本平台进行一键打包、一键提测；QA能够在本平台查看测试报告，来减少后续不必要的测试。综上所述，具有一定的实用性。

### 参考文献

[1]顾勋. 基于Web组件的可视化地图开发平台的设计与实现[D].浙江大学,2021.DOI:10.27461/d.cnki.gzjdx.2021.002485.

[2] 张海潘 软件工程导论(第四版) 清华大学出版社ISNB7-302-07321-x/TP.5312 P179-192, 2003

[3]刘曼. 敏捷开发模式下的质量管理研究[D].北京邮电大学,2021.DOI:10.26969/d.cnki.gbydu.2021.002468.

[4]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.

[5]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.

[6]覃琳,杨程,孙卫宁,覃宏秋.基于敏捷开发下的协同创新管理服务平台能力提升[J].企业科技与发展,2019(02):9-10.

[7]Abdallah Salameh,Julian M. Bass. An architecture governance approach for Agile development by tailoring the Spotify model[J]. AI & SOCIETY,2021(prepublish):

[8]申贵平,陈当阳.一体化敏捷开发和管理平台的实施[J].金融电子化,2013(11):47-49.

[9]Riesener Michael,Doelle Christian,Perau Stefan,Lossie Philipp,Schuh Guenther. Methodology for iterative system modeling in agile product development[J]. Procedia CIRP,2021,100:

[10]Janne Kettunen,Miguel A. Lejeune. Technical Note—Waterfall and Agile Product Development Approaches: Disjunctive Stochastic Programming Formulations[J]. Operations Research,2020,68(5):

[11]遇宇.基于Nodejs的定制化流程引擎设计与实现[J].电脑编程技巧与维护,2020(11):39-40+65.DOI:10.16184/j.cnki.comprg.2020.11.014.

[12]Torgeir Dingsoeyr,Davide Falessi,Ken Power. Agile Development at Scale: The Next Frontier[J]. IEEE Software,2019,36(2):

[13]Nanda C. Surendra,Salman Nazir. Agile Development: Exploring What Practitioners Want to Know[J]. Journal of Software Engineering and Applications,2018,11(1):

[14]李晓纯,孔文熙,朱景福.基于React的在线教学平台设计与实现[J].电脑知识与技术,2021,17(30):95-97.DOI:10.14004/j.cnki.ckt.2021.2891.

[15] Information Technology - Information Software Technology; Studies in the Area of Information Software Technology Reported from University of Oulu (Analyzing the concept of technical debt in the context of agile software development: A systematic literature review)[J]. Computer Weekly News,2017:

[16]刘苗苗. 基于React的智慧管廊平台的前端设计与实现[D].北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.002589.

### 第二章 课题内容

#### 2.1课题内容

本文主要聚焦于如何提升开发人员开发需求的效率，设计并实现一站式敏捷平台，课题主要内容包括以下几点内容:

1) 调查和分析敏捷开发模式在国内外的应用和发展现状。并做好技术的选型工作。  
2) 对用户进行需求分析，确定功能性需求和非功能性需求，并绘制用例图。  
3) 针对用户的需求，明确课题目标，设计系统总体技术架构、功能模块图和主要模块图。  
4) 完成一站式敏捷平台的设计和实现。  
5) 对平台的功能进行全面测试，提高平台的稳定性和可靠性。

#### 2.2系统需求分析

##### 2.2.1 概述

一站式敏捷协同平台主要用户是需求研发上线过程中各协作角色，包括产品、设计、前端RD、前端QA、后端RD、后端QA等。它是一个解决到家C端场景下需求交付的业务层系统服务，包含了以下能力:

* 面向C端的产品侧，整个需求排期，周知，估时，资源耗损的线上化
* 打通需求提测和现有自动化测试，另外建设多端差异性测试在冒烟阶段提醒C端研发同学是否需要冒烟三端
* 需求构建集成提测入版流水线建设，具备需求一键构建、一键提测、一键入版的能力，RD无需关心分支管理规范、分支同步机制、版本集成周期、需求PR合入规范等

平台主要分为三个模块，分别是需求评审，按需集成和按需测试。

##### 2.2.2 用例图描述

本系统主要由项目经理，UI设计师，程序员，测试员使用，不同人员使用的功能不同。总用例图如图2-1所示

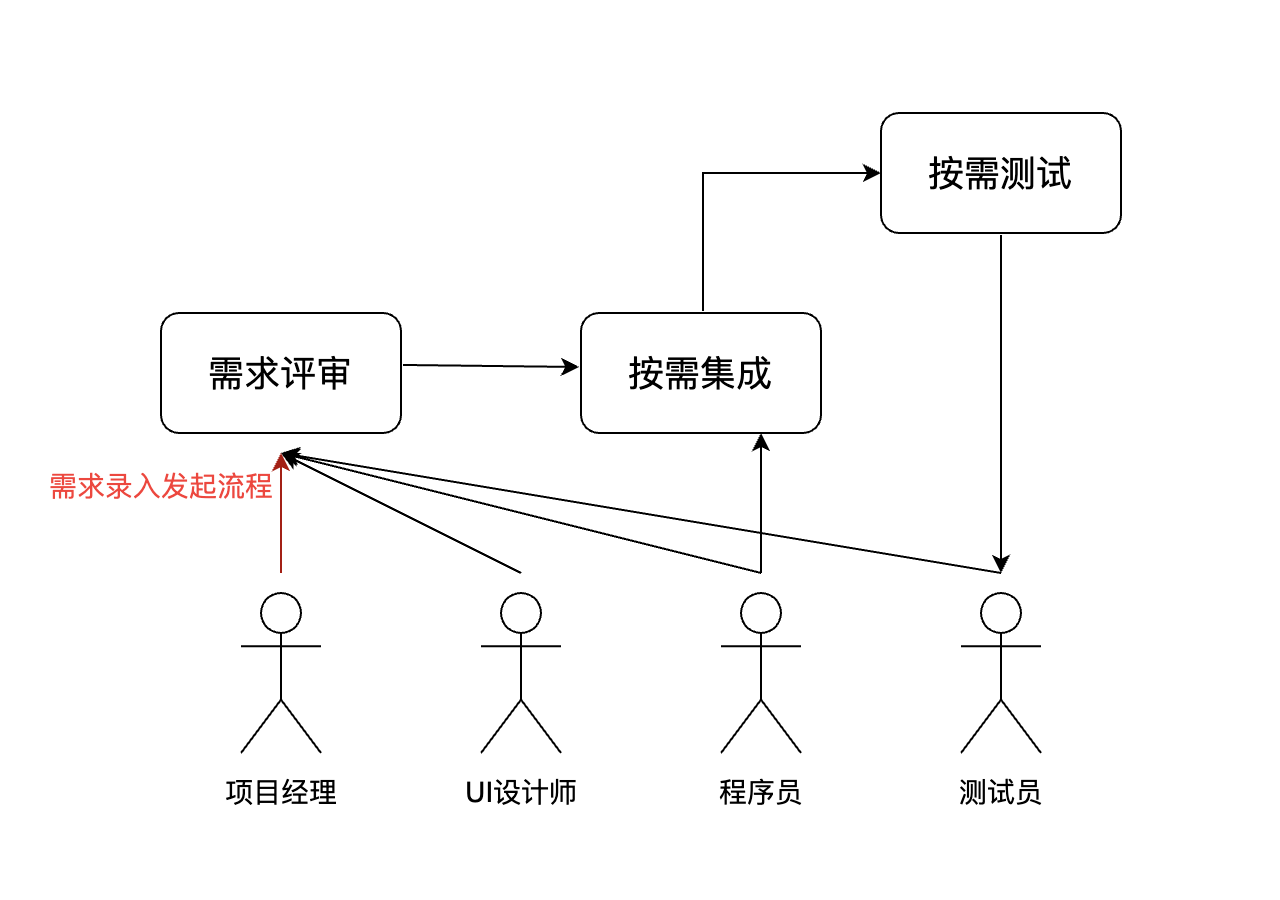


图2-1 总用例图

总用例图包括需求评审，按需集成和按需测试。各功能用例图如下所示：

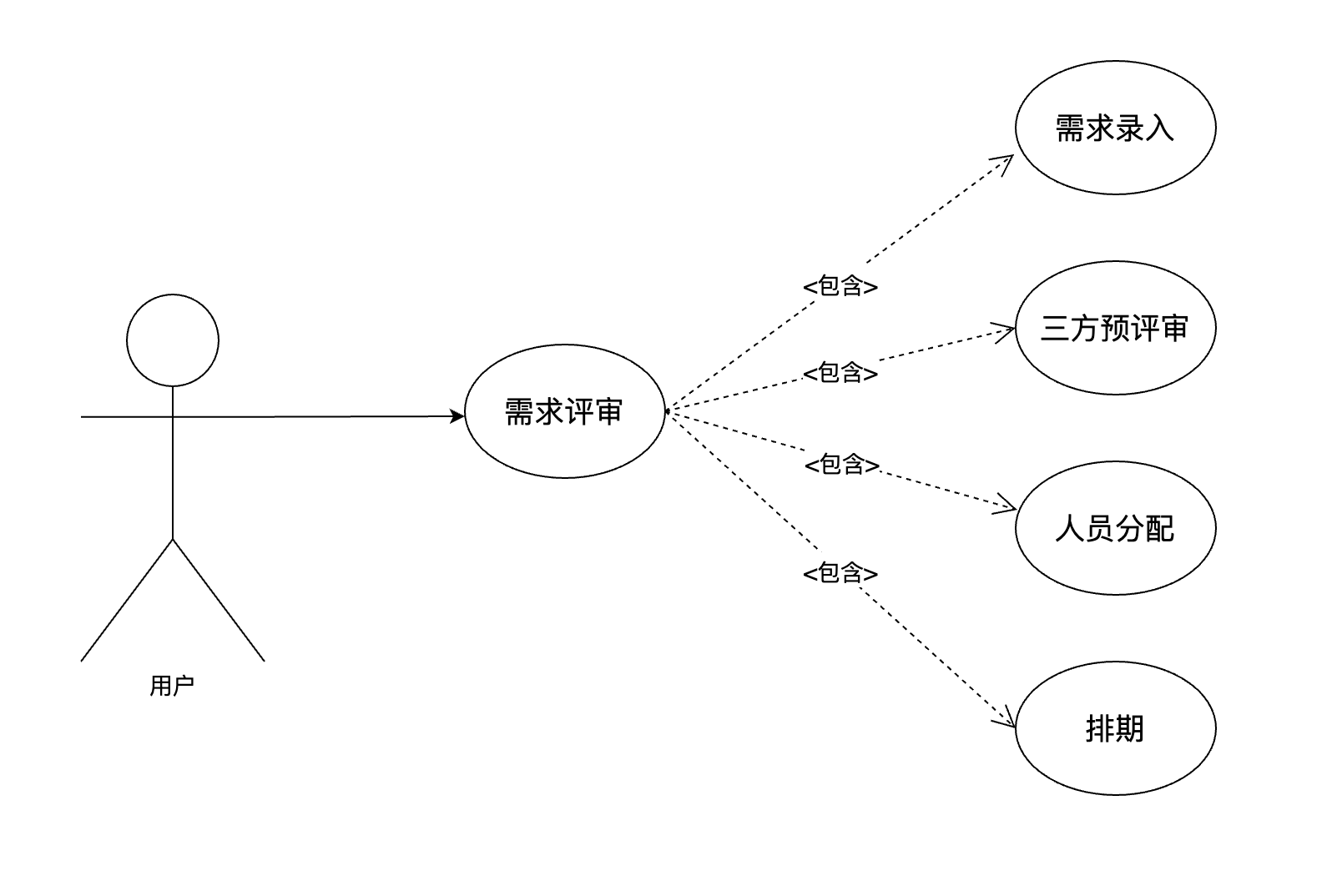
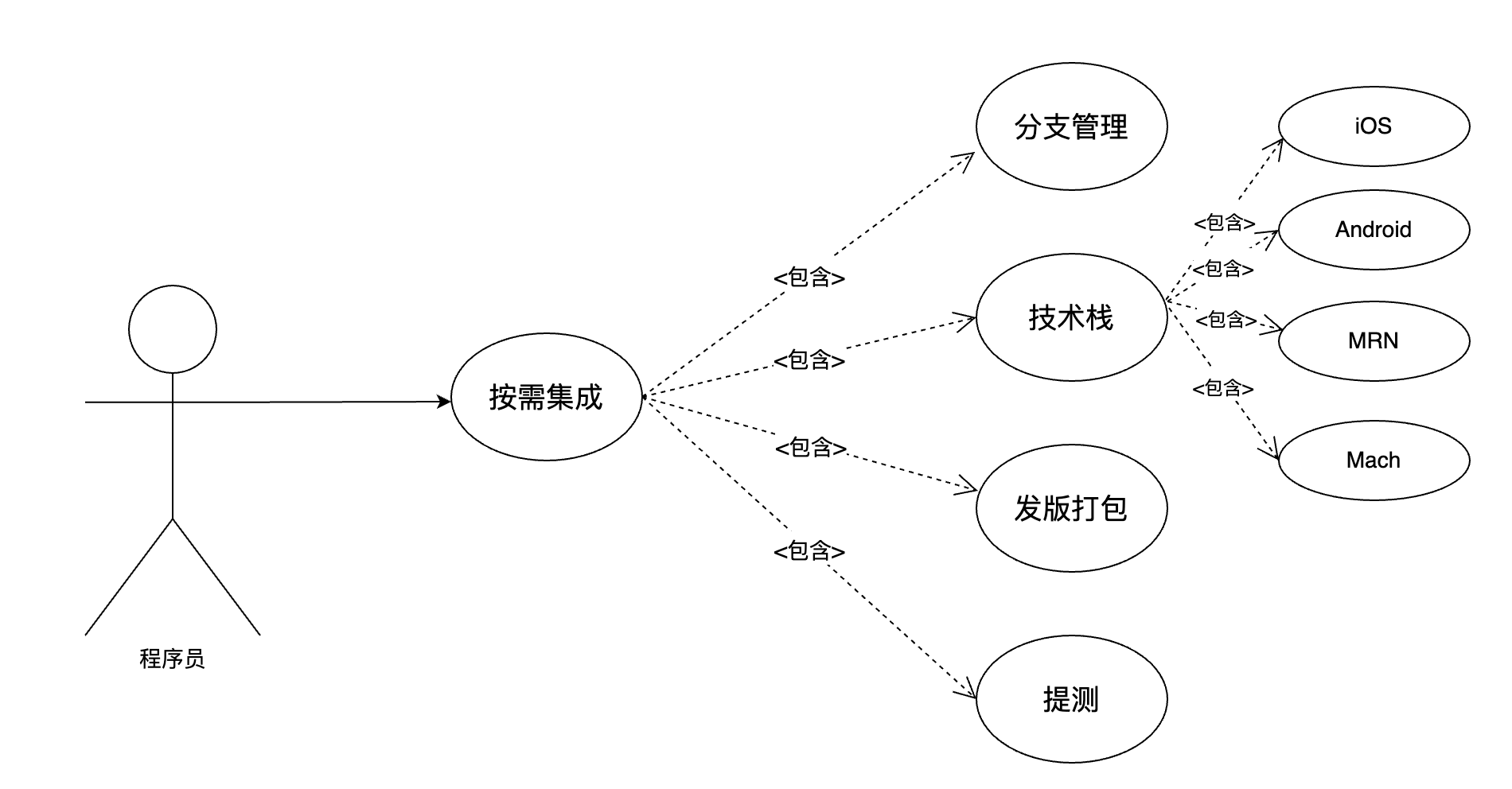
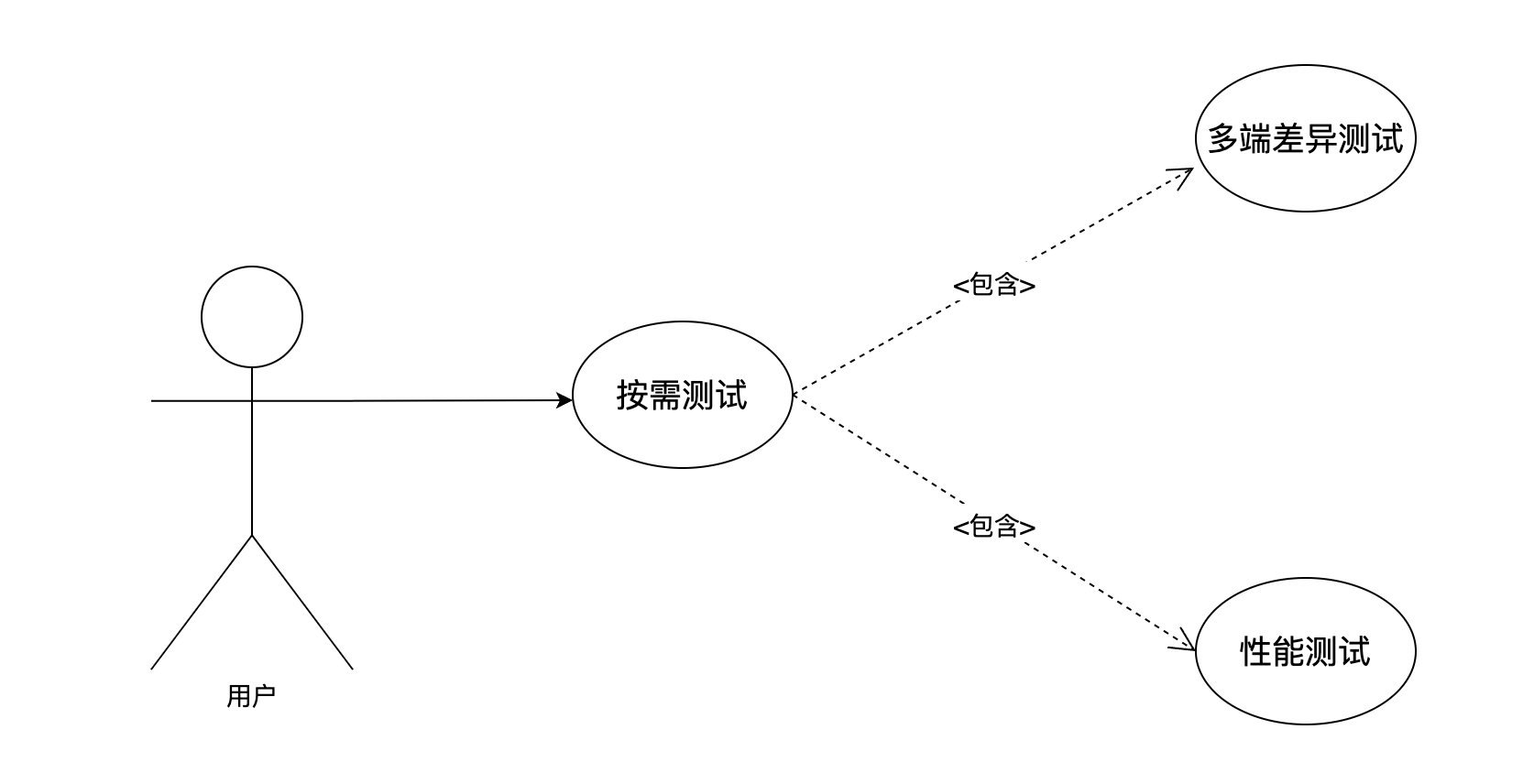


图2-2 需求评审用例图

1）需求评审是所有用户都会使用的，包括项目经理，UI设计师，程序员，测试员。 主要支持需求录入，三方预评审，人员分配，排期功能。



2）按需集成主要提供分支管理、技术栈选择、发版打包和提测等功能,通过提升自动化程度，打通多个公司级服务来降低人力投入，提供集成效率。它的主要的使用者为程序员， 不同于手动去HPX/Talos上触发打包任务, 该平台遵循按需求维度进行打包集成的设计理念针对每个需求，只需要很少或者不用配置，即可持续交付。RD不需要关心使用HPX/Talos的具体细节，甚至不需要关心背后的系统，只需要按照需求维度在平台上一键打包，即可快速获得最新的构建产物。



3）按需测试是对按需集成的产物基础上对接测试服务，通过测试报告评估需求交付的质量，为产研链路的下一个环节RD冒烟、QA测试提供可量化的需求影响范围评估依据。它的用户为程序员和测试员，主要提供多端差异测试和性能测试功能。多端差异测试是指在使用不同技术栈完成需求后的三端代码差异。三端是指外卖APP，美团外卖频道，点评外卖频道。它的主要使用者为RD和QA。

#### 3 概要设计

##### 3.1 目标与概述

根据上文的需求分析，本平台需要在设计完成后，能实现以下功能：

1）用户登陆平台后，可以通过需求评审功能录入需求，通过三方预评审和人员分配功能选择具体完成需求的角色，包括PM，RD以及QA；最后三方协商以后，确定好需求排期，包括需求完成的时间，测试时间以及上线等时间。

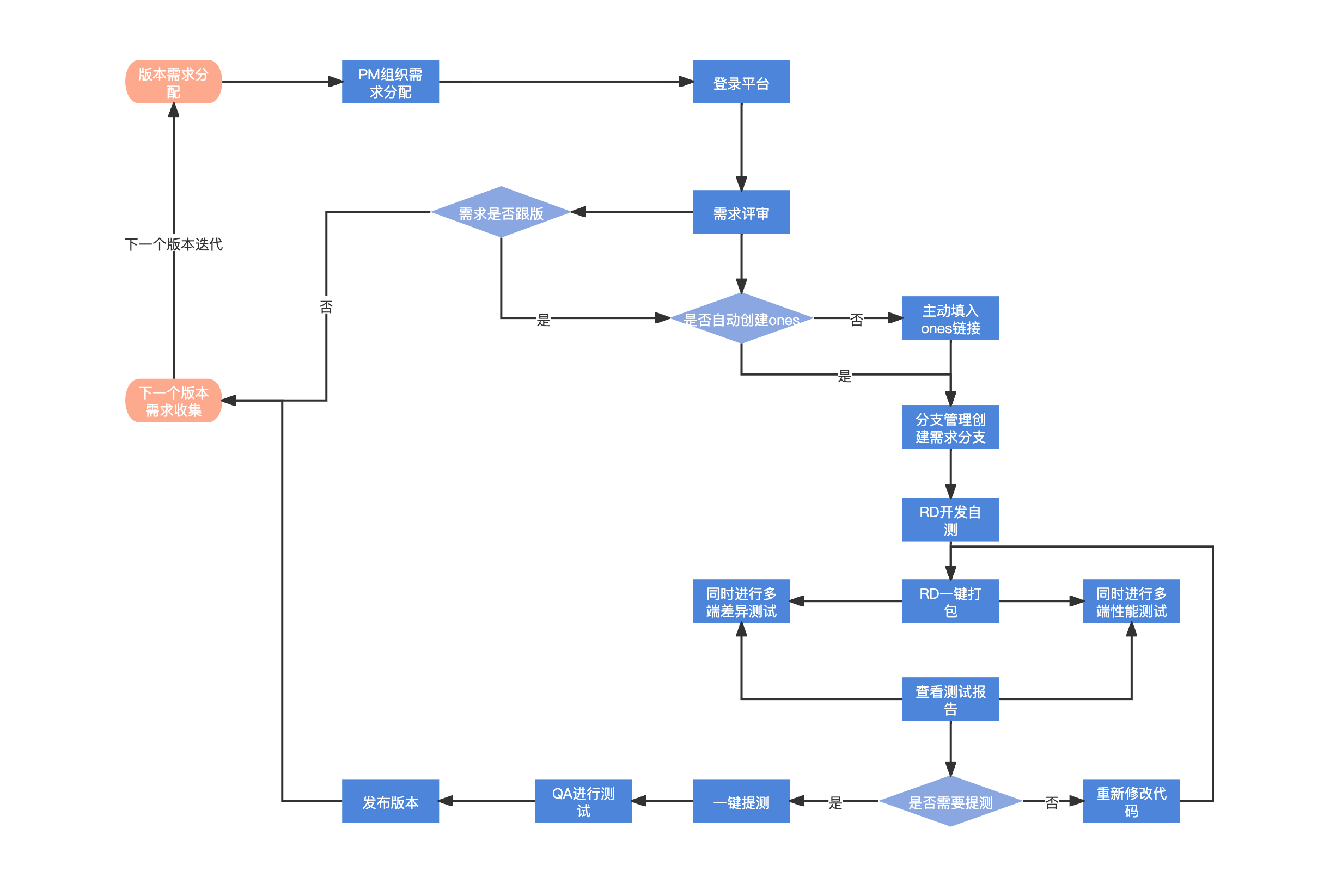
2）在需求开发过程中，RD在分支开发完成以后，可以登陆平台，通过按需集成功能，选择需求分支，集成涉及的技术栈，然后一键打包，等待打包完成以后再通过一键提测功能给QA进行测试。

3）在RD一键打包以后，RD以及QA可以进入平台，查看按需测试的结果，包括多端差异化测试和性能测试。然后再根据相应的结果决定是否要进行冒烟测试。冒烟测试是指在对一个系统进行大规模测试之前，先验证一下软件的基本功能是否实现，是否具备可测试性。

另外，本平台在性能上要达到准确性、可维护性和稳定性三项要求。准确性是指一键打包功能的产物和多端差异分析的结果要准确，因为这个产物最终是要发布到线上来服务外卖用户的，而RD可以根据差异分析结果来决定冒烟测试的次数；可维护性就是要在平台开发过程中，文档、代码遵循软件开发的规范要求。稳定性就是要求平台功能的稳定，不会出现功能使用障碍的情况。

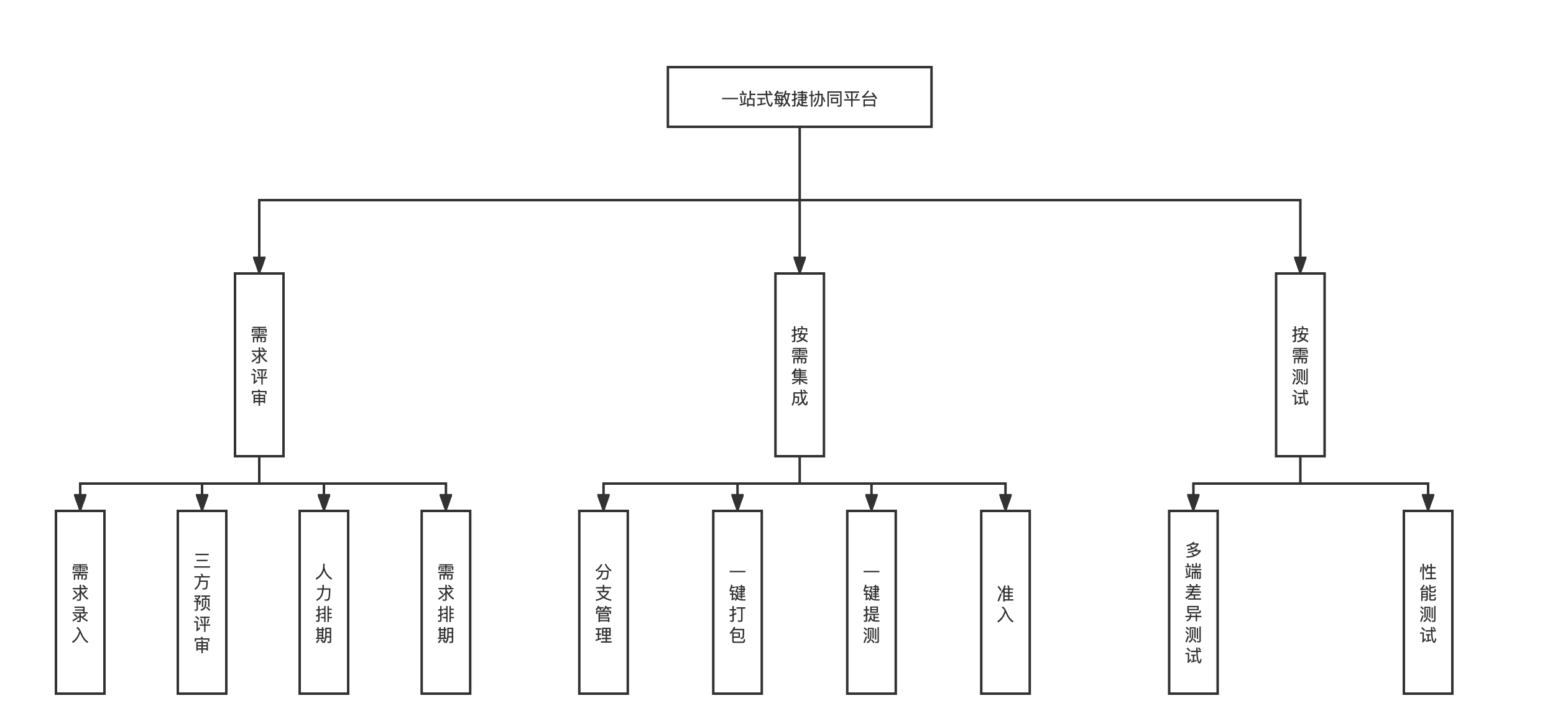
##### 3.2 总体架构设计

##### 3.2.1 总体流程



如上图所示，该平台是一个协同平台，开发流程中的角色通过该平台协同到一起。

##### 3.3功能模块设计



**图 3-2 一站式敏捷系统平台功能模块图**

本平台主要包括需求评审，按需集成和按需测试模块。其中每个模块又分了几个子模块。

需求评审包括需求录入，三方预评审，人力排期和需求排期四个子模块 ；按需集成包括分支管理、一键打包、一键提测和准入四个字模块；按需测试包括多端差异测试和性能测试两个子模块。

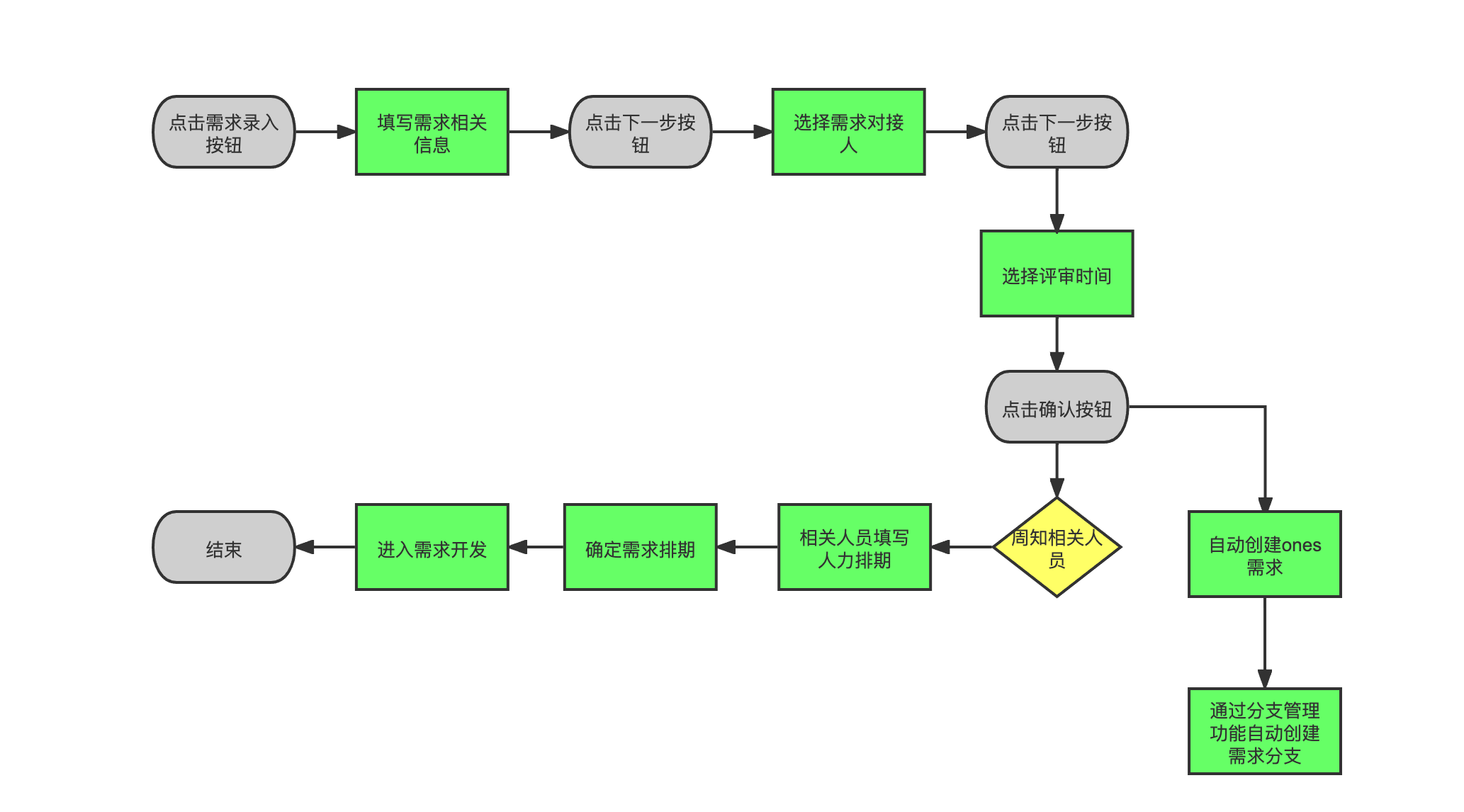
如图3-2平台功能模块图所示，可以得到用户使用平台的主要工作活动如图3-3所示。

从图3-3可以看出，用户需要先登录，在用户登录后，可以看到三大功能模块：需求评审，按需集成以及按需测试。用户先去需求评审模块进行需求录入，然后选择三方预评审时间，接着进行人力排期，最后确定需求排期。而用户开发过程中可以使用按需集成模块的相关功能，包括分支管理、一键打包、一键提测和准入功能；当开发完毕需要进行测试的时候，用户使用按需测试的多端差异测试和性能测试功能进行测试，来判断此次开发完毕的结果。下面详细介绍一下需求评审、按需集成和按需测试主要模块的概要设计。

**画图**。

###### 3.4.1 需求评审模块设计

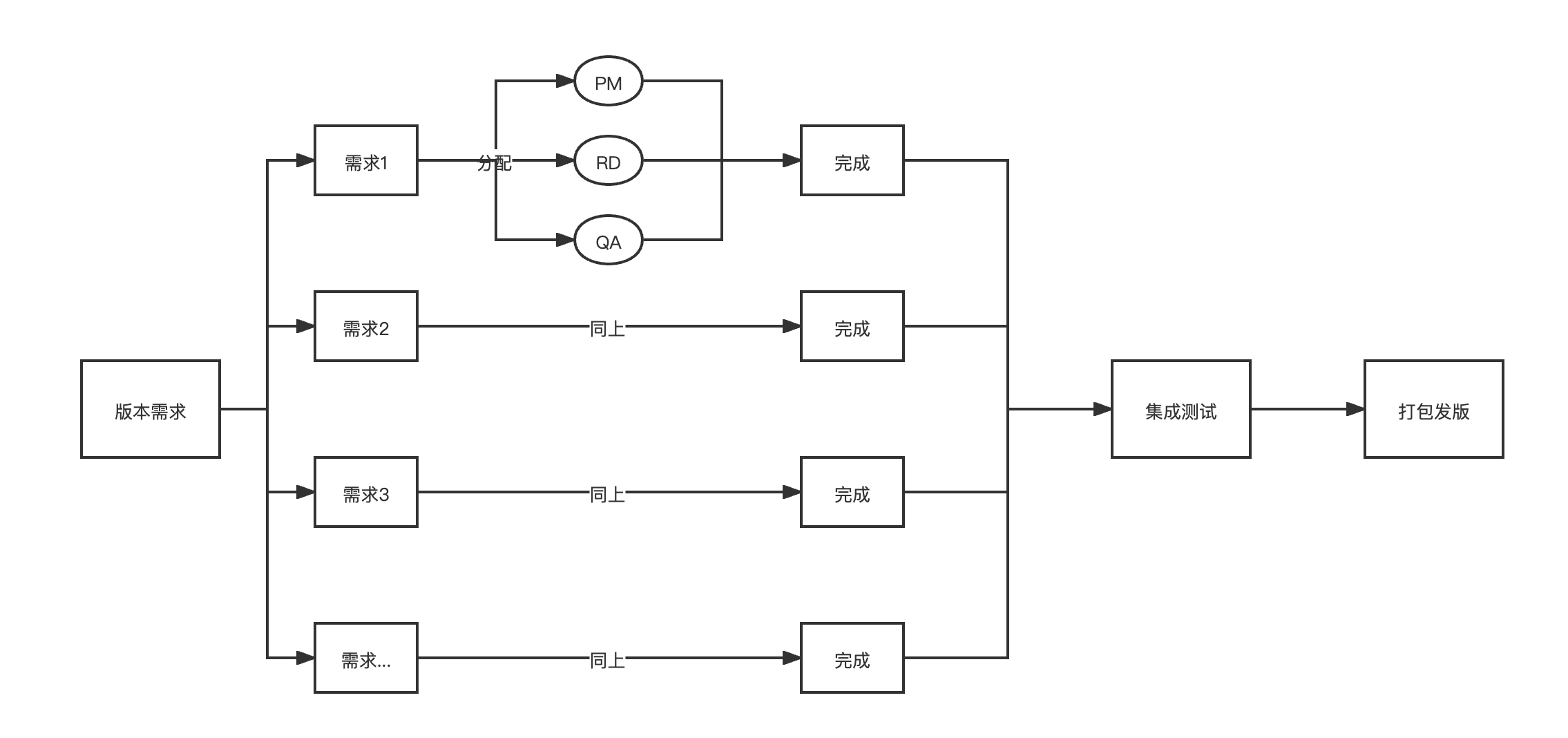
需求评审包括需求录入、三方预评审，人力排期和需求排期。用户点击需求评审按钮后进入需求录入页面，填写需求的文档链接，需求名称，所属的项目空间，需求类型，优先级，敏捷周期；然后点击下一步按钮进入三方预评审页面，选择需求对接人，选择完毕以后，点击下一步，进入选择评审时间设置，为了很好的从原来的双周迭代模式过滤，三方评审时间统一为周一；评审时间设定以后，会通过内部通讯工具周知到相关人员并且在ones上添加需求信息（ones是公司内部进行需求排期的平台）。相关人员点击链接进入人力排期页面，分别填写自己完成该需求的估时，当所有人填完时间以后，整个需求排期就确定了，便进入需求开发期。如图3-3所示。



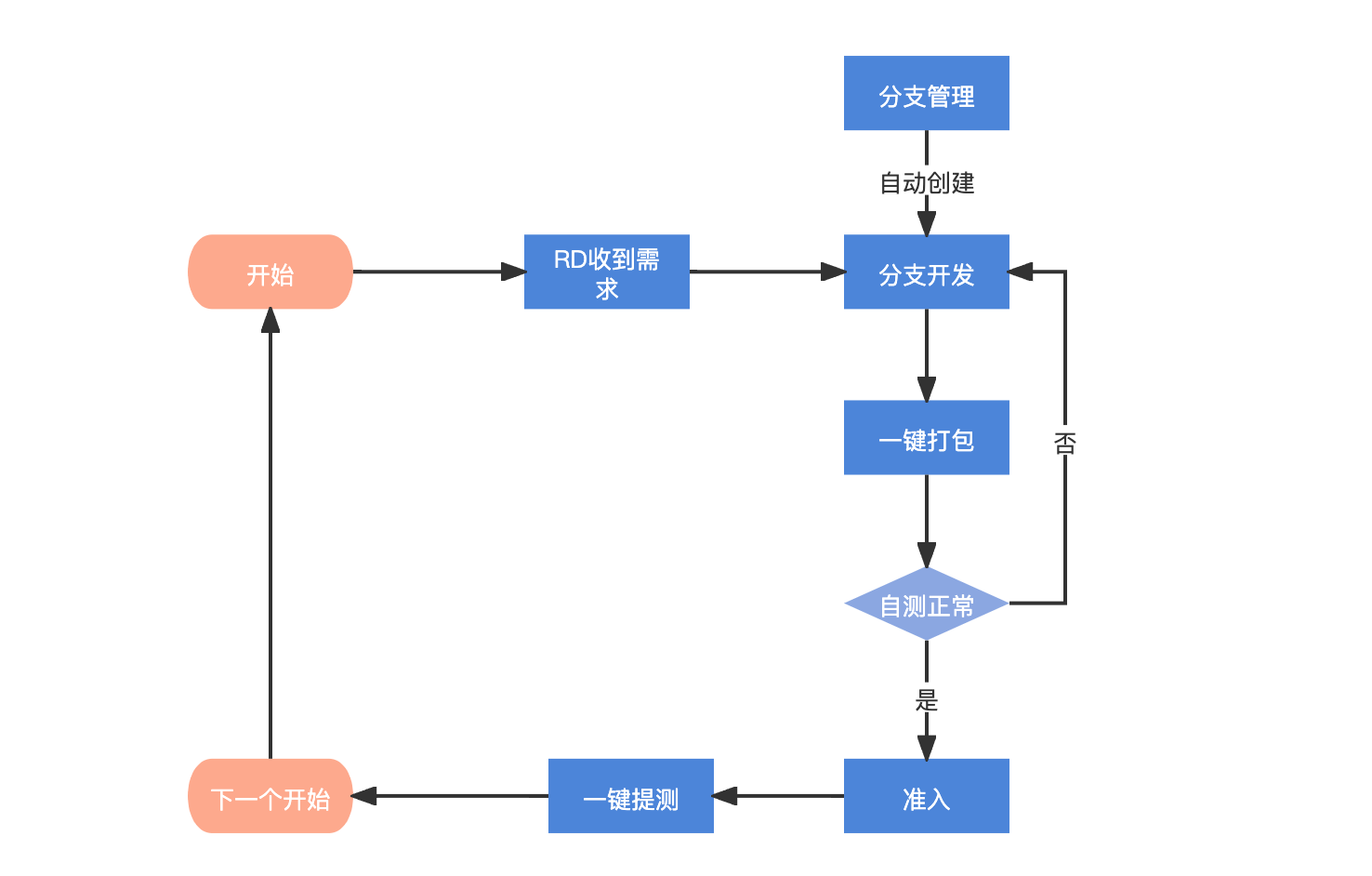
**图3-3 需求评审活动图**

###### 3.4.2 按需集成模块设计

按需集成模块包括分支管理、一键打包、一键盘提测和准入等功能。主要使用者为RD



先叙述一下App发版的过程，如上图所示。一个版本需要上线多个新功能，每个新功能就是一个新需求，大部分需求都会分配相应的PM，RD，QA，三方协同完成需求，然后多个需求集成到一起进行测试，测试通过以后进行打包发版。

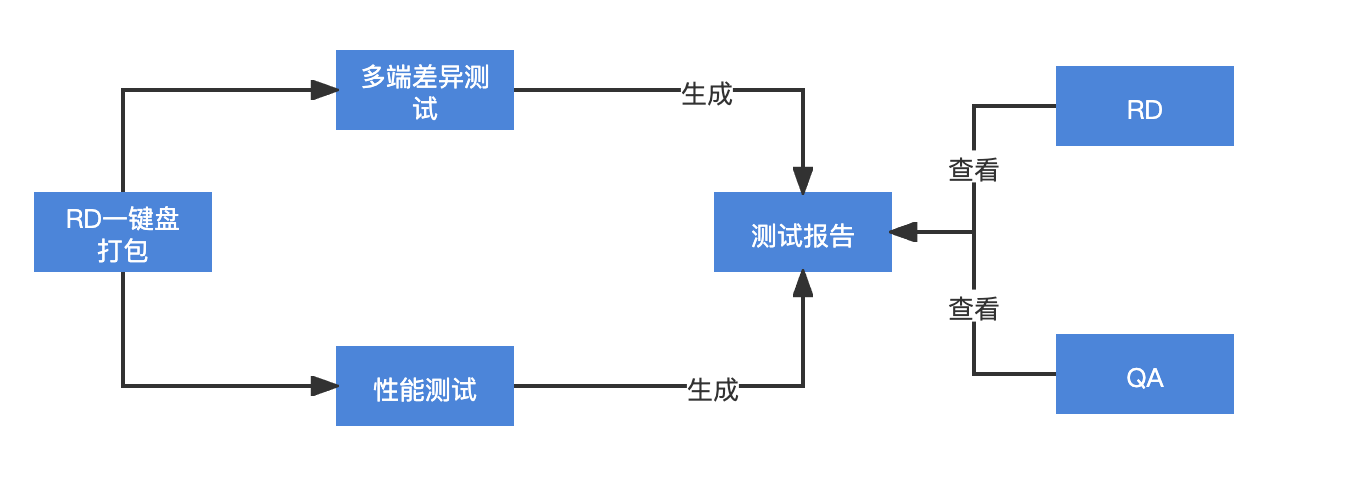


再以一个RD开发的角度阐述开发流程，如上图所示。RD被分配到了一个需求，首先它需要从上一个App版本分支也就是拉取一个新的feature分支作为自己开发需求功能的分支，而这个分支通过分支管理功能自动创建了，然后便在该分支上进行开发和自测。测试需要进行打包，这样就可以使用平台上的一键打包功能，自测完毕以后，通过准入功能将分支代码合进到主仓库。

###### 3.4.2 按需测试模块设计

按需测试模块主要包括多端差异测试和性能测试。主要使用者为RD和QA

多端差异测试是指代码在三端：外卖、美团和点评的差异，性能测试是指用内部测试工具对App的性能进行测试。



如上图流程图所示，多端差异测试和性能测试主要是在打包过程中执行的。

#### 拟采用的开发方法，环境，测试方案等

开发方法：一站式敏捷平台采用前、后端分离的方式进行开发，前端可视化表现层基于HTML, CSS, React(JavaScript)开发，通过网络编程技术实现前台和后台数据逻辑处理和传递。后端开发语言为TypeScript,框架为Egg.js和UmiJS。

开发环境：开发操作系统为macOS，前端开发工具为Visual Studio Code与WebStorm，调试工具为Google Chrome。版本控制工具选择git。后端开发工具选用WebStorm,结构化数据选用MySQL，服务器运行环境使用Linux。

测试方案：对基本功能进行单元测试，参数上验证程序的正确性；集成多个功能进行集成测试，验证模块的正确性。最后阶段进行整个系统的系统测试。

#### 技术难度及特色分析

本平台主要技术难度如下：

1）需求评审：需求评审中的需求排期本身是以敏捷周期模式为基础，考虑到不同业务方原有研发周期模式的不相同，有些业务方仍然是双周迭代等固有模式，而让业务方改成敏捷周期模式需要一个循序渐进的过程，所以在现有情况下如何使得需求排期功能满足各个业务方本来的排期是一个挑战，具有一定的技术难度。

2）多端差异测试：如何保证差异的标准化、差异的完整性是一个技术挑战，可能需要从整个研发链路中去分析，去解决。

本平台主要特色如下：

1）通过打通内部平台、协同研发链路中的多角色，缩短产品交付周期，提高研发效率。

#### 本人主要工作描述

项目开发方面：前端方面主要使用JavaScript语言和React框架编写一些UI组件和功能页面；后端方面使用TypeScript语言进行一些功能接口的编写；

需求分析方面：参与一站式敏捷协同平台的需求分析，调研各个业务方的业务场景，理清共性诉求和个性化诉求。

#### 工作进度的大致安排

2021.8 完成一站式敏捷协同平台的文献调研，了解平台的背景和预期。

选题背景，摘要。

2021.9 ～ 2021.11 学习相关开发技术，确定需求，参与相关功能的架构设计和概要设计。 需求分析报告、概要设计

2021.12 ～ 2022.1 完成项目编码与测试工作 开题报告 各模块代码实现

2022.2 整理相关文档 实习总结报告

1. 2~ 2022.5 根据相关文档撰写毕业论文 论文初稿

2022.6 ～ 2022.8 修改和完善毕业论文、毕业论文答辩 论文定稿

#### 预期成果

1）一站式敏捷协同平台的设计与实现

2）工程硕士学位论文.