

应用软件项目的迭代开发与测试

姜 文,刘立康

(西安电子科技大学 通信工程学院,陕西 西安 710071)

摘 要:随着软件开发技术的发展,敏捷开发的理论和方法获得广泛应用。迭代开发模式是敏捷开发的核心内容。迭代开发模式将软件项目分解为多个迭代阶段,每轮迭代都进行测试并提供输出版本。依据软件迭代开发实践,归纳总结出一种新的迭代开发模式,其主要特点是迭代开发与迭代测试同时进行。首先介绍了软件迭代开发过程中四类人员的职责;详细叙述了迭代开发的模式和工作流程,迭代开发期间编码、持续集成和测试之间的关系;叙述了迭代开发的具体实施;最后介绍了一个软件迭代开发案例。工作实践表明,软件项目采用新的迭代开发模式,可以更好地适应当前开发组和测试组分为两组的实际情况,加强软件开发工程师和软件测试工程师之间的交流和沟通,灵活应对软件开发过程中的各种技术变更;使软件整体开发进度和质量得到有效控制。

关键词:敏捷开发;迭代开发;编码;测试;持续集成;基线

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2019)04-0007-06

doi:10.3969/j.issn.1673-629X.2019.04.002

Iterative Development and Testing in Application Software Project

JIANG Wen, LIU Li-kang

(School of Telecommunication Engineering, Xidian University, Xi'an 710071, China)

Abstract: With the development of the software development technology, the theory and methods of agile development is applied widely. Iterative development model is the core content of agile development. Software project is decomposed into multiple iterative steps by iterative development model, each of which is tested with output version providing. According to the practice of software iterative development, a new iterative development mode is summarized, which is characterized by the simultaneous development of iterative development and iterative testing. First we introduce four types of responsibilities to four types of role in iterative development. Then we detail the iterative development model and work process of iterative development, the relationship between coding, continuous integration and testing during iterative development process, and describe the specific implementation of iterative development. Finally a software iterative development case is given. Practice shows that by adopting new iterative development mode, the software project can adapt to the actual situation better, which is divided into two groups, including development group and test group, strengthen communication and exchange between software development engineers and software test engineers, and flexibly respond to various technology changes in software development process. The overall software development schedule and quality are effectively controlled.

Key words: agile development; iterative development; coding; test; continuous integration; baseline

0 引 言

随着软件开发技术的发展,敏捷开发^[1-8]的理论和方法在软件行业获得广泛运用。敏捷开发包括的方法比较多,主要包括 Scrum 方法、Crystal 方法、极限编程(extreme programming, XP)、自适应软件开发、功能驱动软件开发等。虽然它们在特定的活动和工作方面有所差异,但是它们都采用迭代和增量开发(iterative

and incremental development, IID) 模式,通常简称为“迭代开发”。IID 是所有敏捷方法的核心。

依据软件迭代开发实践,归纳总结出一种新的迭代开发模式,其主要特点是迭代开发与迭代测试同时进行,详细叙述了迭代开发的模式和工作流程;叙述了迭代开发的具体实施,并且给出了一个软件迭代开发案例。

收稿日期:2018-05-30

修回日期:2018-09-12

网络出版时间:2018-12-20

基金项目:国家部委基础科研计划资助项目(A1120132007)

作者简介:姜 文(1986-),女,工程师,硕士,CCF 会员(E200032324M),研究方向为图像处理、数据库应用和软件工程;刘立康,副教授,研究方向为数字通信、图像传输与处理和软件工程。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20181220.1109.066.html>

1 迭代开发和测试

迭代开发^[9-10],也被称作迭代增量式开发或迭代进化式开发,软件迭代开发过程中,整个软件项目根据开发计划被分解为多个迭代周期(一般 2 至 4 周),每一次迭代都由需求分析、设计、编码实现和测试在内的多个活动组成,每轮迭代结束后都会生成一个稳定的、经过验证的软件版本。

迭代测试^[11-16]根据每一轮迭代开发包含的特性数量,确定待测试的用例数量,将整个测试工作分割成一系列小型测试任务,并根据测试用例的数量与难度来安排测试工程师参与迭代测试工作。

- 迭代开发和测试的优点如下:
- (1) 每一轮迭代都为下一轮迭代提供开发基线,软件开发建立在稳定的质量基础之上;
 - (2) 有助于早期发现各种技术问题,及时灵活应对各种技术变更;
 - (3) 有助于增强各类人员之间的交流、沟通,减少在开发和测试工作中的理解偏差;
 - (4) 有利于合理安排工作任务,确保软件版本计划按时顺利完成。

2 软件迭代开发过程中涉及的角色

软件迭代开发过程中涉及四类人员,如图 1 所示,包括软件管理、软件开发、软件测试和其他相关人员;各类人员各司其职,共同完成软件迭代开发工作。

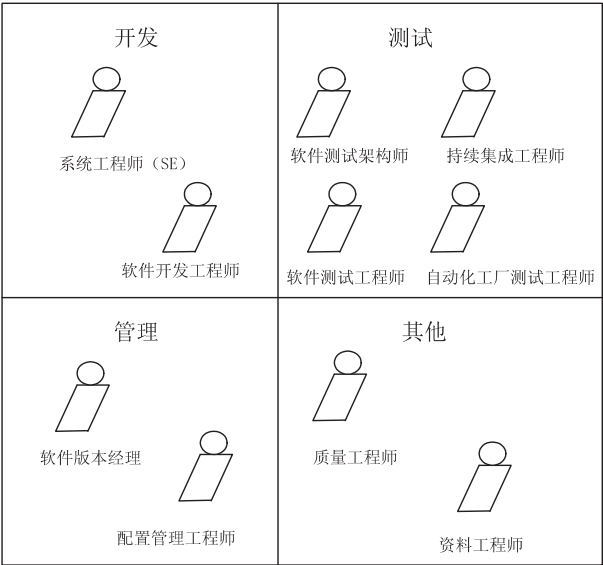


图 1 软件迭代开发角色分类

2.1 管 理

2.1.1 软件版本经理

软件版本经理是软件项目的负责人,负责软件开发的技术管理和人员组织安排。

- (1) 负责主持制定软件项目开发计划,召开软件版本开工会,确保软件迭代开发、迭代测试以及软件产

- 品资料开发按计划顺利开展工作;
- (2) 任命软件开发组负责人、测试组负责人,调配软件开发工程师和软件测试工程师;
- (3) 及时了解迭代开发和测试工作进度,确保软件版本计划顺利进行;
- (4) 了解每个人的工作情况,表彰相关人员,激励大家的工作热情;
- (5) 每轮迭代结束后,组织相关人员进行项目验收和评审工作。

2.1.2 配置管理工程师

配置管理工程师负责软件成品、半成品、代码、脚本和技术文档的管理工作。

- (1) 管理各种文档资料:软件产品的各种技术性资料;
- (2) 管理软件代码:软件产品代码和测试用例自动化脚本代码;
- (3) 软件产品版本管理:保存软件产品开发过程中的所有版本,需要时可以回溯查询;
- (4) 版本库中目录和文件的读、写、浏览的权限管理:给相关人员分配权限;
- (5) 开发基线管理:确定软件开发基线,软件开发基线的更新;
- (6) 开发过程中的变更管理:主要包括需求变更、计划变更、设计变更以及工程变更,每次变更都需要经过严格的评审程序。

2.2 开 发

2.2.1 软件系统工程师

系统工程师(SE)负责完成软件需求设计、架构设计和详细设计,将设计场景文档转换为开发任务,参与处理软件开发过程中遇到的各种问题。

2.2.2 软件开发工程师

软件开发工程师负责编写代码、源代码调试,自验之后,将源代码合入版本库;处理问题单,修改代码。

2.3 测 试

2.3.1 软件测试架构师

测试架构师(TSE)负责编写软件产品测试策略;细化与分析软件测试点,编写软件测试用例;参与处理软件测试过程中遇到的各种问题。

2.3.2 软件测试工程师

软件测试工程师根据设计场景文档、测试策略和测试用例编写自动化测试脚本。搭建测试环境,进行软件测试。对检测到的软件缺陷提交问题单,跟踪问题单处理的全过程。

2.3.3 持续集成工程师

持续集成工程师在软件迭代开发期间,每天从版本库下载最新代码,定时完成持续集成工作。在工作

中发现的问题,及时反馈给软件开发工程师处理。

2.3.4 自动化工厂测试工程师

自动化工厂测试工程师在迭代开发期间和持续集成工程师共同完成冒烟测试任务。

2.4 其他

2.4.1 质量工程师

在迭代开发过程中,质量工程师全程监控软件开发和测试的工作流程,监控和审核迭代过程中各项度量数据是否达标。迭代开发完成之后,参与迭代开发与测试工作的质量评审。

2.4.2 资料工程师

资料工程师根据系统工程师提供的设计场景文档,编写产品文档。

3 迭代开发模式和 workflows

依据软件迭代开发实践,归纳总结出一种新的迭

代开发模式。其主要特点是迭代开发与迭代测试同时进行。这样处理的优点包括:

- (1)适应当前软件开发与测试分为两个不同组的组织形式;
- (2)有利于人员的调度安排,充分发挥软件开发工程师和软件测试工程师各自的优势;
- (3)软件编码和测试用例脚本编写同时进行,提高测试效率,缩短开发周期;
- (4)软件开发工程师在编写代码和修改代码时,可以使用测试用例自动化脚本测试自己编写的代码,促进开展测试驱动开发工作。

3.1 迭代开发模式和流程

新的迭代开发模式和流程如图2所示。

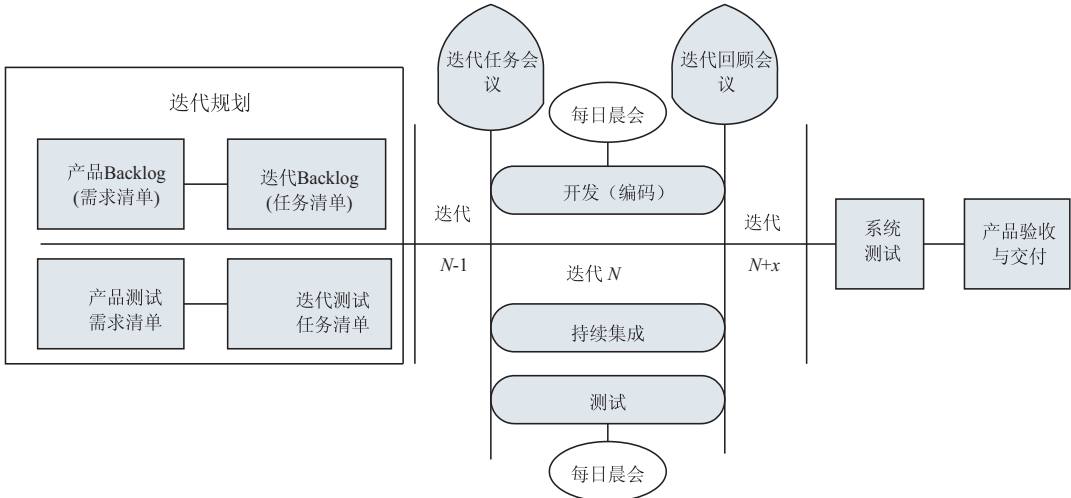


图2 迭代开发模式和流程

3.1.1 迭代规划

SE 根据软件产品需求规格文档确定产品 Backlog (需求清单),按照用户需求和模块之间的关联程度划分迭代个数。确定迭代 Backlog (任务清单),分配每个迭代需要完成的任务清单(迭代开发中,一个任务相当于一个功能点,一个小的需求)。迭代 Backlog 是团队在一轮迭代中的任务清单,是团队的详细迭代开发计划

TSE 根据软件产品需求规格文档和产品 Backlog (需求清单)编写测试策略,确定产品测试需求清单;分配每个迭代需要完成的测试任务清单,编写测试用例。

3.1.2 迭代任务会议

迭代启动前,SE 组织软件开发工程师详细讨论本轮迭代开发计划,澄清需求、对完成标准达成一致;确定迭代任务的优先级,细化工作计划和分配迭代开发

任务。

TSE 组织软件测试工程师讨论本轮迭代测试计划,分配测试任务;确定测试用例自动化脚本开发的优先级。

3.1.3 每日晨会

每日晨会通常需要 15 分钟左右。SE 与软件开发工程师开会,了解项目的进度,及时解决实际工作时遇到的问题。TSE 召集软件测试工程师开会,了解工作进度,及时处理遇到的各种问题。

3.1.4 迭代回顾会议

迭代回顾会议在每轮迭代结束后举行,相互交流、总结经验和教训,讨论开发与测试工作存在的各种问题,将其中未完成的任务转移到下个迭代中。

3.1.5 系统测试

最后一轮迭代工作完成之后,需要对软件进行系统测试。自动化工厂测试工程师连跑测试用例自动化

脚本对软件进行冒烟测试(预测试),然后软件测试工程师对软件进行全面系统测试。之后软件进入产品验收与交付阶段。

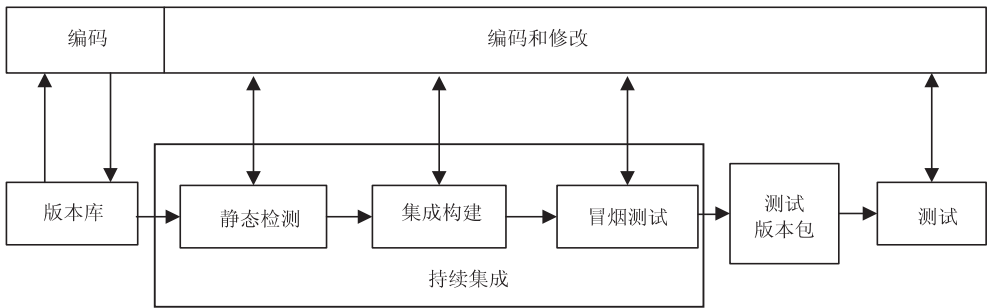


图3 编码、持续集成和测试三者的关系

3.2.1 编 码

软件开发工程师主要负责迭代开发过程中的编码工作。

- (1)每天登陆版本库,根据版本库提供的新基线更新工作副本;
- (2)检出需要的文件开始工作,编写代码,对编写好的代码进行单元测试,静态走码检查,冲突处理和本地构建工作。可以采用软件测试工程师编写的测试用例自动化脚本测试自己编写的代码,最后将评审过的代码提交到版本库;
- (3)处理持续集成和测试中发现的各种问题,定位问题原因,修改软件代码,提交版本库。

3.2.2 持续集成

持续集成工程师搭建的集成构建工程,通过制定定时任务来自动完成从版本库更新代码,开展持续集成工作。集成构建工程包括静态检查、集成构建(编译、出包)、冒烟测试三项任务。

- (1)静态检查:调用静态测试工具分析检查代码,静态测试工具对代码进行语法扫描,找出不符合编码规范的地方;同时检查多个文件,检查当前文件的同时还会检查所有与之相关的文件。
- (2)集成构建:对从版本库下载的代码进行编译、链接,生成新的版本包。通常集成构建失败属于相当严重的错误,需要及时定位处理。
- (3)冒烟测试:冒烟测试是对软件新的版本包基本功能进行测试验证,自动化工厂的测试工程师负责完成集成构建工程中的冒烟测试任务。
- (4)为软件测试工程师提供测试版本包。

3.2.3 测 试

- 软件测试工程师根据 TSE 安排的任务开展工作:
- (1)按照优先级顺序编写测试用例自动化脚本;
 - (2)搭建测试环境,安装部署测试版本包,开展测试工作;
 - (3)对于测试中出现的问题提交问题单,与软件

3.2 迭代周期内的开发测试活动

迭代周期内,编码、持续集成和测试三者的关系如图3所示。

- 开发工程师共同定位处理,跟踪问题单处理的全过程;
- (4)为自动化工厂提供测试用例自动化脚本。

4 迭代开发工作的具体实施

SE 和 TSE 完成迭代规划之后,开始迭代开发和测试的具体实施。

4.1 软件开发

4.1.1 开发串讲

版本经理公布软件产品版本计划后,系统工程师根据需求规格文档,完成软件产品的概要设计和详细模块设计,输出设计场景文档;召集软件开发工程师进行产品特性串讲,对软件开发工程师进行产品业务培训;介绍软件产品需求背景、软件产品架构、业务流程、模块功能、各模块之间的调用关系和接口。

4.1.2 开发澄清

开发串讲之后,进行开发澄清讨论,其特点在于双向交流。开发澄清的主体是软件开发工程师,系统工程师、测试架构师与测试工程师也参加澄清讨论。软件开发工程师按设计场景和业务流程讲解自己负责的软件模块功能特性、输入和输出处理;针对需求提出问题,保证对需求的理解一致。SE 在澄清过程中,就一些关键点(容易遗漏、犯错的地方)提问,确认开发工程师与系统工程师的理解一致。在开发澄清讨论中,如果发现对软件特性的理解,有不确定的地方由系统工程师进行确认;如果发现系统工程师给出的设计场景文档中有不一致或遗漏的地方,需要系统工程师申请 CCB(change control board)裁决,软件开发工程师根据 CCB 裁决的结论来进行特性开发。

4.1.3 迭代开发

- 迭代开发的具体过程如下:
- (1)通过迭代任务会议,系统工程师给软件开发工程师分配开发任务,进一步澄清需求,讨论和处理解决相关的技术问题;
 - (2)通过每日晨会,系统工程师了解软件开发工

工程师的工作进展情况,并讨论处理工作中遇到的各种问题;

(3)软件开发工程师负责软件产品的编码工作,同时与系统工程师共同处理持续集成工程师与软件测试工程师提交的问题单,定位问题产生的原因,修改软件代码。

4.1.4 迭代回顾

每轮迭代结束后系统工程师召集软件开发工程师,举行回顾会议,总结经验和提出改进点,最后形成迭代回顾纪要。

4.2 软件测试

4.2.1 用例串讲

测试架构师把测试策略与测试用例等技术文档,提供给软件测试工程师,让软件测试工程师熟悉软件特性;测试架构师给软件测试工程师分析测试策略中的测试点,进行测试用例讲解。进行用例串讲时,通常软件开发工程师也参与,确认测试用例设计无理解错误和遗漏点。用例串讲之后,软件测试工程师对测试用例存在的问题与测试架构师、开发工程师进行澄清讨论。

4.2.2 测试反串讲

测试工程师根据对测试用例的理解,对分配给自己的测试任务,编写软件特性测试点分析文档;文档需要测试架构师与相关的软件开发工程师进行评审。评审通过后,测试工程师对测试场景用例进行反串讲,通过这种方式确认测试工程师能够正确理解测试用例。测试架构师与软件开发工程师参与测试反串讲。

4.2.3 迭代测试

- 迭代测试具体过程如下:
- (1)通过迭代任务会议,测试架构师给软件测试工程师分配测试任务,并讨论和处理解决相关的技术问题。
 - (2)通过每日晨会,测试架构师了解软件测试工程师的工作进展情况,并讨论处理工作中遇到的各种问题。
 - (3)测试工程师编写测试用例自动化脚本。
 - (4)测试工程师搭建测试环境,安装测试版本包,开始测试工作。
 - (5)对于自动化测试,测试工程师通过自动化脚本进行测试,测试完成之后,参于测试任务的测试工程师进行脚本交叉检视,并根据检视意见修改各自编写的脚本;测试架构师检视所有执行测试的自动化脚本,测试工程师根据测试架构师的检视意见修改脚本;之后将脚本合入版本库。
 - (6)对于不能编写自动化测试脚本的测试用例,测试工程师手动执行测试用例,测试的过程中将相关

的数据保存下来,由测试架构师评审测试过程。

4.2.4 问题单处理

测试过程中,测试工程师将发现的软件缺陷,提交问题单到缺陷管理系统中。问题单走到对应的软件开发工程师名下,软件开发工程师处理问题单,修改后的代码重新合入版本库,验证通过之后问题单走到回归状态。根据问题单所示的内容启动回归测试,如果回归测试的结果和预期结果一致,表明问题单回归测试通过,可以关闭问题单,完成问题单处理流程。

4.2.5 迭代回顾

迭代测试完成之后,测试架构师组织软件测试工程师召开测试迭代回顾会议。会议主要关注每个测试场景的缺陷密度(即问题单数/代码量),每个场景需要补充的测试点,本轮测试执行过程中需要改进的地方。最后形成迭代回顾纪要。

4.3 迭代验收和质量评估

每次迭代结束时需要对本轮迭代工作进行验收和质量评估。

4.3.1 迭代验收

迭代结束之后,系统工程师与测试架构师组织软件开发工程师和软件测试工程师进行迭代 SE 验收,SE 验收过程中,测试工程师讲解各测试场景典型用例的测试过程,演示脚本的运行过程与运行结果。可以由系统工程师给出验收用例,测试工程师进行验收测试,将验收用例的测试结果反馈给系统工程师。验收结束之后,由系统工程师和测试架构师总结验收结果,提交迭代验收纪要。

4.3.2 迭代质量评估

迭代质量评估由版本经理、系统工程师、测试架构师、开发组组长、测试组组长以及质量工程师参与,属于软件版本层面的迭代过程的总结。主要根据迭代测试过程中缺陷问题级别和数量的分析结果、各测试场景中需要补充的测试点,进行软件版本质量评估。重点是分析与安排下一步如何改进与提升软件产品质量。

5 在软件产品开发中的应用

文中提到的迭代开发与测试方法,在公司软件产品开发中应用多年。在大型软件的开发与测试过程中广泛应用,取得了良好的效果。采用该迭代开发模式,提高了软件开发与测试的效率和质量,也为后期软件维护工作打下良好的基础。

工作实践表明,开发串讲、测试串讲、开发澄清以及测试反串讲使软件开发工程师和测试工程师能够快速理解软件特性,减少对于软件特性的理解偏差。迭代回顾可以及时总结在上一轮迭代过程中需要改进的

地方,为后续的工作提供改进的方向。

下面给出了一个通信软件产品迭代开发与测试的案例。

5.1 案 例

K 产品接受客户局点 W 的需求,开发产品的一个新特性,新特性的迭代开发和测试时间将近 2 个月。根据版本经理制定的迭代计划,整个迭代过程分为 3 轮迭代,具体的安排如表 1 所示。

表 1 迭代开发和测试工作统计表

迭代序号	时间/天	软件开发	软件测试	测试用例数
		工程师人数	工程师人数	
迭代一	14	2	2	109
迭代二	28	4	3	189
迭代三	16	4	4	223

在 2 个月的迭代开发测试过程中,软件开发工程师总共完成了 3 万行的代码开发;软件测试工程师共提交问题单 39 张;测试用例总数 521 个,编写测试用例自动化脚本 487 个,编写测试套 8 个;整个迭代过程中测试用例执行率达到了 100%,测试自动化率为 93.47%。

5.2 技术问题处理

K 产品在原有软件功能上增加新特性,在新特性开发过程中处理了大量的技术问题。下面介绍一个具有代表性的问题。

在软件开发初期,开发工程师将软件代码合入版本库之后,持续集成生成版本包;执行冒烟测试,验证新版本包的基本功能时出现 5 个测试用例自动化脚本执行失败。测试工程师初步分析之后确认不是自动化测试脚本的问题,开发工程师分析之后发现是由于对原有功能的代码进行修改引入的缺陷。测试工程师提交问题单;开发工程师修改代码,重新将代码合入版本库,对代码执行回归测试,修复了该缺陷。

6 结束语

软件迭代开发的理论和方法在软件项目开发中获得广泛应用,尤其是对于开发大型复杂软件系统具有重要价值。长期的工作实践表明,采用迭代开发模式,可以加强软件开发工程师和软件测试工程师之间的交流和沟通,灵活应对软件开发过程中的各种技术变更,从而降低软件产品开发风险。迭代开发模式有利于软件开发计划的管理和实施,使软件开发进度和质量得

到有效的控制。

参考文献:

[1] BITTNER K, SPENCE L. 迭代软件开发项目管理[M]. 罗景文,译. 北京:清华大学出版社,2010.

[2] 李 宾. 基于敏捷开发流程文档管理系统的设计与实现[D]. 西安:西安电子科技大学,2010.

[3] 葛 岳. 基于敏捷开发方法的软件项目管理研究[D]. 天津:天津大学,2014.

[4] HODA R, NOBLE J, MARSHALL S. Self-organizing roles on agile software development teams[J]. IEEE Transactions on Software Engineering, 2013, 39(3): 422-444.

[5] KETTUNEN P, LAANTI M. Combining agile software projects and large-scale organizational agility[J]. Software Process: Improvement and Practice, 2013, 13(2): 183-193.

[6] PETERSEN K, WOHLIN C. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case[J]. Journal of Systems and Software, 2009, 82(9): 1479-1490.

[7] 明恒统. A 公司软件开发模式研究[D]. 武汉:武汉工程大学,2017.

[8] DINGSΦYR T, NERUR S, BALIJEPALLY V G, et al. A decade of agile methodologies: towards explaining agile software development[J]. Journal of Systems & Software, 2012, 85(6): 1213-1221.

[9] STEPHENS R. 软件工程入门经典[M]. 明道洋,曾庆红,译. 北京:清华大学出版社,2016.

[10] LARMAN C. 敏捷迭代开发:管理者指南[M]. 北京:人民邮电出版社,2013.

[11] CRISPIN L, GREGORY J. 敏捷软件测试:软件测试工程师与敏捷团队的实践指南[M]. 孙伟峰,崔 康,译. 北京:清华大学出版社,2010.

[12] GREGORY J, CRISPIN L. 深入敏捷测试—整个敏捷团队的学习之旅[M]. 徐 毅,夏 雪,译. 北京:清华大学出版社,2017.

[13] 杨丹君. 迭代测试流程在软件测试中的应用[J]. 工业控制计算机, 2016, 29(12): 21-22.

[14] 尚 洁,李春雷. 快速迭代开发模式下系统测试方法[J]. 指挥信息系统与技术, 2017, 8(3): 93-98.

[15] 吴 俊. 敏捷测试在 S 银行软件项目中的应用研究[D]. 上海:东华大学,2017.

[16] CARRERA A, IGLESIAS C A, GARIJO M. Beast methodology: an agile testing methodology for multi-agent systems based on behaviour driven development[J]. Information Systems Frontiers, 2013, 16(2): 169-182.