

敏捷开发方法的实践与思考

郭靖 孙艺通 黎中有 肖硕承

近年来,为适应快速变化的软件需求,提高软件交付速度、质量和客户满意度,敏捷开发方法引起了广泛关注。本文将敏捷方法归纳为两类,分别选取每类中具有代表性的方法,即极限编程(Extreme Programming,XP)和Scrum进行阐述。通过对这两种方法进行详细地剖析,能够使读者更深刻地理解敏捷原则,只有在实践中对敏捷模式进行适当的改进和灵活运用,才能达到预期的目标和效果。

在早期软件规模化和工程化过程中,由于缺少规划和设计环节,软件结构随着软件规模的膨胀变得无法维护,并且由于忽视需求的管理和资产的沉淀,软件开发变得混乱无序,同时,也由于未考虑软件的可测试性和可维护性,导致软件的交付失败率居高不下。在这种背景下,瀑布模型被提出,通过系统化、规范化、按顺序递进的软件开发流程来保障交付质量。瀑布模型根据软件生命周期,将软件开发过程主要分为问题定义、可行性研究、需求分析、设计、软件编码、测试、运行和维护等阶段,各阶段要求具备明确的结果、规范的文档以及严格的评审,且各个阶段之间的依赖关系非常强烈,只有上一阶段结束才可进入下一阶段,这是标准的顺序模型。然而,随着信息技术的不断发展,由于软件需求的不确定性、软件产品响应的能力以及系统的复杂性等因素,导致瀑布模型无法满足软件发展的需要,越来越多的中小软件开发团队采用敏捷开发方法,以更加灵活和迭代的方式来推进项目。

一、敏捷开发方法概述

敏捷开发方法是一种迭代、增量式的软件开发方法,旨在通过密集的团队合作和快速响应变化来交付高质量的软件产品。该方法强调客户参与、自组织团队、适应性规划和频繁交付价值,将软件开发项目分解为小的模块,并通过逐步迭代的方式进行解决,这些模块逐渐构建起整个应用软件。敏捷开发方法强调高效、快速和有效,是一种轻型的科学方法,相比于重型方法,敏捷开发避免了不必要的中间环节,节省了团队的时间和精力。此方法在过程的设计上寻求平衡,以简洁的步骤获得令人满意的结果,摒弃了烦琐的流程,

提供了一种更加灵活、高效的开发方式。敏捷开发主要有以下几个特点^[1]。

第一,迭代式开发。整个开发过程分为多个独立的迭代周期,每个周期包括设计、开发、测试和交付等阶段。第二,增量交付。产品是在每个迭代周期结束时被逐步交付使用,而不是在整个开发过程结束的时候一次性交付使用。每次交付的产品都是可以部署到用户应用环境中被用户使用,能给用户带来即时效益和价值。第三,及时反馈。敏捷开发方法主张用户能够全程参与到整个开发过程中,强调用户和利益相关者的持续合作,表现为需求变化和用户反馈被动态管理并及时集成到产品中,确保项目朝着正确的方向发展,满足用户需求。第四,持续集成。它是开发团队将代码频繁集成到主干(主要代码仓库)的过程,其目的是确保团队能够快速、频繁地检查和验证代码的质量,减少集成问题以及解决问题的时间。第五,自我管理。敏捷开发强调团队的自我管理和自主决策,团队需要有足够的决策权和自由度来进行项目管理和产品开发。团队成员也会主动发现能力短板和效率瓶颈,并及时自主改进。

在软件项目开发中,比较常见的敏捷方法有Scrum、精益、看板、XP等,前3种偏向于过程管理,属于敏捷方法论中的一个派系,最后一种偏向于工程实践,代表着敏捷领域的另一派系。下文将对具有代表性的两种方法——极限编程和Scrum进行阐述。

二、常见的敏捷开发方法

(一) 极限编程

1. 概述

极限编程^[2]提出的目的是找到快速编写高质量软件的方法,并能够适应客户不断变化的需求。XP是一种专注于编程技术、团队协作的软件工程方法学,由价值观、原则、实践和行为4部分组成,如图1所示,它们彼此相互依赖、关联,并通过“行为”贯穿于整个生命周期。

2. 价值观

极限编程中有5个核心价值观:沟通、简单、反馈、尊重和勇气。这些核心价值观代表了团队成员的

一种积极进取的特定心态，使他们在实现共同目标的道路上竭尽全力。

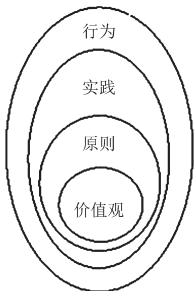


图1 XP四个组成部分

第一，沟通。成员之间面对面“沟通”，通过交流解决问题，提高效率。第二，简单。寻找最简单的方法来完成工作，避免过度设计，造成资源浪费。第三，反馈。收集系统、客户和团队各方面的反馈信息，通过“反馈”及时调整战略方针及项目方向。第四，尊重。通过“尊重”团队成员所付出的努力，来达到客户、开发人员、管理层三方互相信任与理解的目的。第五，勇气。团队成员可以客观地评估由于自身失误造成的后果，并敢于接受批评和不断改进。

3. 原则

第一，快速反馈。团队成员理解客户在开发过程中给出的反馈，并能够迅速作出反应。第二，假设简单。开发人员需要专注于当前的重要工作，遵循所有问题都可以简单解决的原则。第三，增量变化。对产品一小步的改变比一次性的大改变效果更好，任何问题都应该通过一系列能够带来差异的微小改动来解决。第四，拥抱变化。团队成员需要做好心理准备，以接受产品不断变化的需求。第五，高质量的工作。通常采用测试先行的编码方式提供质量保障。

4. 实践

XP是描述软件开发工程具体实践的敏捷框架，如图2所示，包含13个实践，具体分析如下。

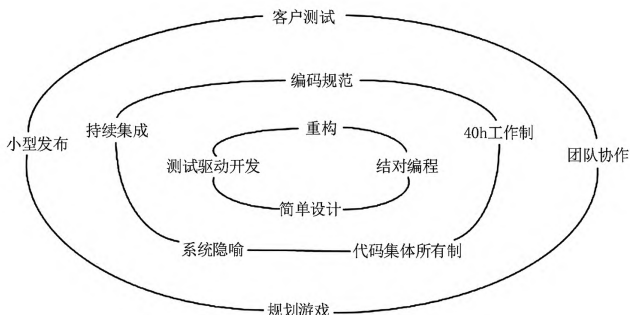


图2 XP实践

第一，团队协作（完整团队）。强调所有参与项

目的团队成员，包括开发人员、测试人员、客户代表、项目经理等，都是开发过程中不可或缺的一部分，并且应该共同协作以确保项目的成功。第二，规划游戏。在策划阶段，由客户或利益相关者定义新应用程序的规范（功能、价值、优先级），这将成为项目团队进行项目成本估算和管理的基础。第三，客户测试（现场客户）。要求客户在开发现场一起工作，明确用户需求，作出相应业务决策，并为每次迭代提供反馈。第四，小型发布。XP方法论秉承的是“持续集成，小步快走”的理念，也就是说，每一次发布的版本应该尽可能地小，让客户实时了解项目进展情况，从而提出更多的反馈意见，以便在下次迭代中计划进去，提高客户满意度。第五，系统隐喻。让项目参与人员必须对抽象概念理解一致。第六，持续集成。团队每天尽可能多次进行代码集成，以便及早发现和解决问题。第七，编码规范。建立有效的代码规范，可以加强团队内代码一致性、消除开发中存在的分歧点、提高软件稳定性，进而加快协作速度。第八，每周40h工作制。团队成员每周工作时间不得超过40h，保证开发时的效率。第九，代码集体所有制。在极限编程里没有严格文档管理，代码为开发团队共有，由团队所有人负责。第十，简单设计。从最简单的设计开始，系统通过迭代和增量不断完善。第十一，测试驱动开发。XP分为软件层面和系统层面的测试驱动开发。其中，软件层面是指颠覆传统软件开发方式（设计、编码实现、测试），即先写测试用例描述需求，然后利用软件使测试通过，最后重构代码改进设计。系统层面是指从用户故事中拆解测试用例，完成功能测试，向客户展现功能实现的情况，通过不断迭代，最终构建出满足客户需求的软件。第十二，重构。在不影响物体外部特性的前提下，重新优化其内部结构，此处的外部特性就是保证测试的通过。第十三，结对编程。2个程序员合用1台计算机编程、1个编码、1个检查，以提高软件编码的质量，并且2个人的角色可经常变换，保持开发者的工作热情。

5. 行为

第一，持续沟通。团队成员应频繁交流，确保信息流通。第二，客户参与。客户或客户代表应参与开发过程，提供反馈和需求。第三，代码审查。定期进行代码审查，以提高代码质量和团队协作。第四，自动化测试。使用自动化测试来确保代码质量，并快速发现问题。第五，维护设计。在开发过程中维护和更新设计，以适应需求变化。第六，适应性计划。根据项目进展和反馈调整计划。

6. 总结

极限编程是一种轻量级、高效化、低风险、科学性且充满乐趣的软件开发方式,旨在解决小团队面临的需求变化大且开发风险高的问题。但同时因其独特的开发机制,也造成本身的局限性,主要体现在以下几个方面。

第一,团队规模限制。XP不依赖正式文档,重视面对面交流,如果团队规模过大,口头交流就会变得极为困难,同时,由于XP实现代码集体所有,如果成员过多,软件将会变得难以控制。第二,文化认同度不高。结对编程比较受争议,不是所有人都认可这种方式会提高编程效率及质量,并且每周40h工作制也同样会让项目责任人无法接受,同时测试驱动开发更是颠覆传统的思维方式,被认为是增加了工作量和时间成本。

(二) Scrum 方法

1. 概述

Scrum方法是一种迭代式增量软件开发过程^[3]。它不是一套标准化流程,而是用于组织管理工作的框架,由价值观、原则、实践三部分组成,如图3所示。

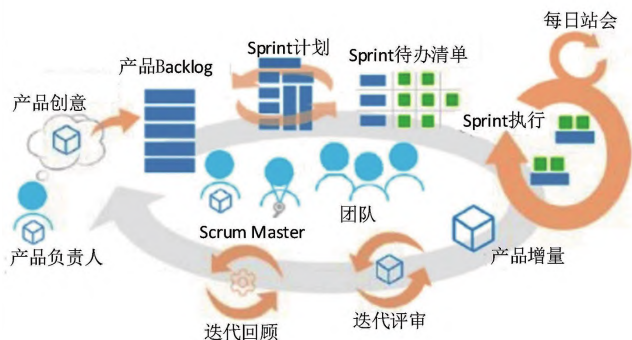


图3 Scrum 框架

2. 价值观

Scrum包含承诺、勇气、专注、开放和尊重五大价值观,为团队工作、行动和行为指引方向。

第一,承诺。每位团队成员都愿意对目标作出承诺,并且通过按时完成目标来维持团队的信用。第二,勇气。每位团队成员都需要有勇气作出承诺,履行承诺,敢于面对失败,解决问题。第三,专注。每位团队成员都需要把心思和能力用到各自承诺的工作中去。第四,开放。每位团队成员都需要共享项目中的一切信息。第五,尊重。团队成员互相尊重和信任。

3. 原则

Scrum奉行透明、检视和适应三大原则来支撑整个过程的实施,优化对未来的预测和控制风险。

第一,透明。Scrum过程中,要先为关键环节制定统一标准,然后保证关键环节对团队成员是显而易

见的,最后要确保所有关注这些环节的成员对观察到的事物达成统一的理解。第二,检视。Scrum过程中,为了尽早发现差异,使用者必须经常对工件和完成目标进行检视。第三,适应。在检视过程中,一旦发现执行过程偏离目标,就需要尽快调整过程内容,努力缩小偏差,保证质量。

4. 实践

Scrum实践定义了实现Scrum框架所需要的三种角色、三种工件和五种活动,具体如下。

三种角色:一是产品负责人,主要负责确定产品的功能和达到要求的标准、指定产品的发布日期和交付内容、最大化产品及开发团队工作的价值;二是敏捷教练,负责Scrum项目团队的运转和协调,是Scrum迭代过程的指导者,保证所有团队成员都能对Scrum有正确的理解,并在实践过程中帮助团队清除障碍,顺利交付目标;三是开发团队,即跨职能的团队,拥有创建产品增量所需的全部技能,主要负责软件产品在Scrum规定流程下的开发工作。

三种工件^[4]:一是产品代办列表(Product Backlog),它不仅是一份涵盖功能、技术、缺陷等需求的有序列表,还是产品需求变动的唯一来源,产品负责人会根据市场环境或业务需要管理产品代办列表中的内容和排序。二是冲刺任务列表(Sprint backlog),一组为当前冲刺选出的产品代办事项列表条目,外加交付产品增量和实现冲刺目标的具体计划。冲刺列表在冲刺规划会议中产生,一旦冲刺规划会议结束,产品负责人就不能再修改冲刺列表的故事清单。三是产品增量,其是截至当前所有冲刺完成的产品代办列表条目的总和。它们用来展示项目进展和价值增量。

五种活动^[5-6]:一是待办事项整理会议,澄清用户故事、理解商业价值、确立下个迭代需求范围及较细颗粒度任务拆分。二是冲刺计划会,在冲刺开始前,确定当前冲刺目标和任务。三是每日例会,对团队每天任务进展进行检视和沟通。四是冲刺评审会,对该冲刺阶段完成的需求进行评审,且开发团队向产品负责人及项目利益相关人员演示完成的产品增量。五是冲刺回顾会,在每个冲刺结束后,检视冲刺过程中存在的问题,制订改进措施,以作为下一个冲刺持续改善的依据。

5. 总结

Scrum工作方式决定了它能够让开发团队实现效率最大化,快速响应需求,并对不同的功能需求设定优先级,实现团队自主管理。但Scrum侧重管理框架,并未给出技术层面的要求,因此可能会产生软件产品质量不高、发现问题较晚等问题。

三、结语

侧重管理类的敏捷方法和侧重工程实践类的敏捷方法都有各自的优缺点,通过对敏捷方法适当的改进和融合,形成适合自己特有的开发方法,才能取得更好的实践效果。对于大型软件项目,敏捷方法不能按部就班的套用,应该在敏捷和不敏捷之间寻找一种折中方案,既能快速响应客户需求、提高开发效率和市场竞争力,又能够在产品生命周期内有效管控质量。

引用

[1] 朴勇.敏捷过程在软件工程课程中的教学实践

[J].计算机教育,2015(24):78-80.

[2] Kent B,Cynthia A.解析极限编程[M].雷剑文,李应樵,陈振冲,译.北京:机械工业出版社,2011.

[3] Mike C.Scrum敏捷软件开发[M].廖靖斌,吕梁岳,陈争云,译.北京:清华大学出版社,2010.

[4] 林阳涛.GJB5000B二级与敏捷实践的融合技术[J].电子技术与软件工程,2022(24):59-63.

[5] 丛超.Scrum敏捷方法在G公司软件项目管理中的应用研究[D].济南:山东大学,2020.

[6] 高玄凌.Scrum及XP组合应用策略研究[J].科技资讯,2017,15(32):13,15.

作者简介:郭靖,比亚迪汽车工业有限公司,硕士,工程师,研究方向为软件工程、功能安全等;孙艺通,比亚迪汽车工业有限公司,硕士研究生,工程师,研究方向为整车控制软件;黎中有,比亚迪汽车工业有限公司,本科,工程师,研究方向为整车控制软件;肖硕承,比亚迪汽车工业有限公司,本科,工程师,研究方向为软件工程。