

Scrum 敏捷软件开发方法实践中的改进和应用

陈国栋, 罗省贤

(成都理工大学 信息技术与科学学院, 四川 成都 610059)

摘要:如今软件开发更加注重质量、服务和效率,传统的软件开发模式(如瀑布模式)已被先进的敏捷开发模式所替代。但是采用敏捷开发模式仍然有其一定的局限性,简单的照搬套用必然会适得其反。文中在遵循 Scrum 方法的宗旨和基本准则的前提下,对实践中的项目模块启动管理和内存文件管理实行如下应用:改进 Scrum 方法的每日例会方式、采用结对编程的模式、重视必要的开发文档、丰富 Scrum 方法的回顾会议、坚持使用测试驱动和设计持续集成。这样的灵活应用增进了团队的积极性,提高了项目的生产率,对同样采用 Scrum 方法作为开发模式的团队具有一定的指导意义。

关键词:Scrum;敏捷;持续集成;测试驱动

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2011)12-0097-03

Improvement and Application of Scrum Agile Software Development Method in Practice

CHEN Guo-dong, LUO Sheng-xian

(College of Information Technology and Science, Chengdu University of Technology, Chengdu 610059, China)

Abstract: At present, people pay more attention to quality, service and efficiency in software development. The traditional software development mode (such as the Waterfall Mode) has been substituted by the advanced agile development mode. Nevertheless, there is still absolutely limit when using the agile development mode. It will be counter-productive if apply this mode simply and mechanically. On the basis of following the tenet and the basic standard of Scrum method, some practical applications such as the start management and MFS (Memory File System) management are applied in this paper: improving the pattern of Scrum daily meeting, adopting pair work mode, valuing the necessary document, enriching the retrospective meeting of Scrum, insisting to use TDD mode and designing continuous integration. This flexible application not only advanced the productivity of items but also enhanced the positivity of the team. It has a certain guiding significance to those teams who also use Scrum method in development.

Key words: Scrum; agile; continuous integration; TDD

0 引言

随着软件行业的发展,基于详尽的需求分析、系统设计和把需求变更作为梦魇的瀑布模式已经逐渐被开发者们所舍弃了,而能够更快地面向市场、更好地响应客户需求变更的敏捷开发模式深得开发者们的青睐^[1]。正如敏捷宣言所述:个体和交流胜于过程和工具;可以工作的软件胜于综合的文档;客户协作胜于合同谈判;应对变化胜于遵循计划^[2]。敏捷开发模式更重视软件的生产率,且适用于解决需求模糊或快速变更的问题。现今已经有很多种敏捷方法被付诸到实践中了,广为人知的有极限编程(Extreme Programming,

XP)、Scrum、动态系统开发(Dynamic System Development Method, DSDM)、自适应软件开发(Adaptive Software Development, ASD)、水晶方法(Crystal Clear)、特征驱动开发(Feature-Driven Development, FDD)和测试驱动开发(Test-Driven Development, TDD)等^[3]。敏捷方法在实践中取得了巨大的成功。很多数据和实例告诉我们,敏捷是提高软件生产率不争的事实。然而过程本身不是敏捷的,但人可以敏捷,团队或者组织可以敏捷。如果只是做敏捷而不是敏捷地去做,就会导致敏捷方法应用的失败^[4]。在敏捷开发方法实践中需要去体验、分析以及进行必要的方法改进,以切实获得采用敏捷方法所带来的最大效益。

1 Scrum 简介

Scrum 是当今被广泛应用的敏捷开发模式之一。它有着自己鲜明的优势:客户成为开发团队的一部分;能够频繁提交可以工作的中间增量产品;可以不断更

收稿日期:2011-04-21;修回日期:2011-07-25

作者简介:陈国栋(1984-),男,硕士,CCF 会员,研究方向为高性能计算领域中的网络并行计算;罗省贤,教授,研究方向为高性能计算领域中的网络并行计算、网格计算、信号及信息处理领域中的数字信号处理方法研究及软件开发、信号及图像非线性处理等。

改项目的需求以适应用户需求的变化;不需制定详尽的不切实际的计划和编写冗长的文档使得团队更加灵活自如,自由地自我组织和自我管理使得团队积极主动、敏捷创新^[5]。Scrum 也有着区别于其他模式的特质。它拥有特定的成员角色:产品的负责人 PO (Product Owner), Scrum 主管 SM (Scrum Master) 和团队 (Team) (适宜的人数是 5 到 10 人)。它拥有独特的实践方式:冲刺 (Sprint), 计划会议 (Sprint Planning Meeting), 评审会议 (Sprint Review Meeting), 回顾会议 (Sprint Retrospective), 每日例会 (Daily Meeting), 产品订单梳理 (Product Backlog grooming)。它拥有着特殊的工件:产品订单 (Product Backlog), 燃尽图 (Burn Down Chart), 完成标准 (Definition of Done), 产品增量 (Product Increment)^[1]。

Scrum 开发过程是以 Sprint 为增量的迭代开发过程,一般一个 Sprint 适宜的迭代周期是 1 到 4 周。Sprint 开始时,团队已从按优先级顺序排列好的产品订单中选择适合本团队的项目,与 PO 澄清需求并生成同样按优先级顺序排列好的 Sprint 订单,并且保证在本次 Sprint 结束前做完。每个工作日团队都要收集汇报彼此的任务进程和余下的任务量。Sprint 结束时小组展示最终的项目成果,并收集来自团队之外的反馈,用以下一个 Sprint 的自我提升。图 1 描述了 Scrum 的开发流程。

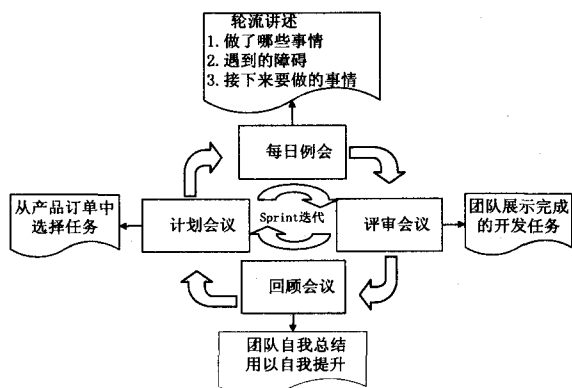


图 1 Scrum 的迭代过程

2 Scrum 方法在实践中的灵活运用

2.1 项目背景

文中的项目背景是某知名通信设备公司研发中心的开发和测试工作。研发团队具有多年 Scrum 开发经验,团队任务是研发和测试该公司某一产品线上的几个功能模块,其中包括基于底层 Linux 操作系统和顶层应用层的中间件模块:启动管理和内存文件管理。

2.2 Scrum 方法的改进

(1) 改进每日例会的模式。

按照 Scrum 倡导的每日例会形式,每个成员应轮

流回答三个经典问题,这使得我们小组的日会在很长一段时间内基本沦为了一场激烈的辩论会,而这些讨论往往是徒劳的。对此,决定对这种已经背离 Scrum 宗旨的日会进行改进。改进的主题就是转变角色,让项目任务成为会议的主角,因为 Sprint 订单原本就是对项目任务的细化。于是新的日会改变了模式:仍然由主管主持,所有成员群视白板,但变成了一种问答形式。如图 2 所示,主管会按优先级顺序遍历项目,针对细化了的每个项目的每个任务提问三个问题:今天花了多少时间、做到了什么程度(日会安排在下班前一小时)?遇到了什么问题?明天怎么解决、继续做多少?相关负责成员回答问题,其他成员洗耳恭听。如此下来效果非常明显,首先是会议时间平均不超过 10 分钟,节省的时间等效于一笔宝贵的财富。再者是大家对各项任务的进展一目了然,同时不同项目之间不同任务之间的成员的付出、进度、成果也一目了然。这样改进后完全达到了日会目的,对这种简单高效的会议方式,团队成员是乐此不疲。

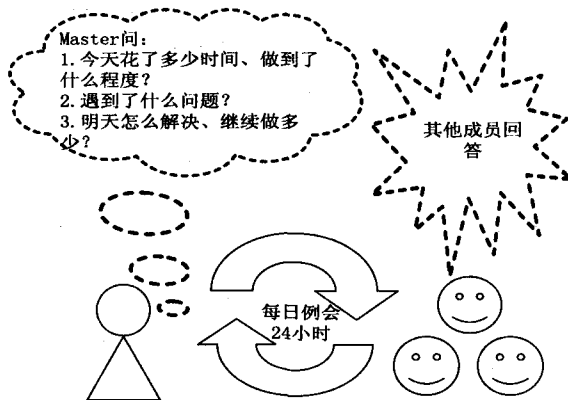


图 2 改进后的日会模式

(2) 采用事半功倍的结对编程方法。

Scrum 提倡团队配置精英成员,即每个团队成员都可独当一面、技术过硬、开发经验丰富^[6]。然而在软件行业起步晚、开发技术还相对落后的国内,鲜有具备十年研发经验的开发人员仍奋战在第一线的情形,一般开发人员都仅具有 2 到 5 年的研发经验。我们的团队就是一个典型的例子,8 人组成的团队平均开发经验不到 3 年,其中还有 2 个新手。因此,团队出现了发布的产品缺陷 (Bug) 多、生产率低、新手进展慢、一些模块过于依赖某一个人等许多问题。针对这样一个低效局面,尝试使用了结对编程方法,每个功能模块都至少由两人负责,避免了因某一成员缺席而导致一些模块的研发进展停滞的情况。结对编程的过程就是一个连续代码评审的过程,这无疑会提高代码的质量,并且结对编程可促进成员相互学习、彼此提升,特别是加速了新员工的成长,从而提高了整个团队的工作效率^[7]。

(3) 重视必要的开发生文档。

Scrum 轻文档重开发的敏捷理念并不是杜绝所有的文档书写。文档可以帮助新人快速熟悉小组项目,也有利于当前项目结束后的维护工作,或者是后续项目的跟进。将客户的体验形成几页纸的说明文档,可以使开发人员对客户体验有清晰的了解。这样在每次 Sprint 开始之前,有了定义好的产品订单,本次 Sprint 便可以持续、平滑地过渡到下一次 Sprint^[8]。

(4) 避免流于形式的回顾会议。

按照 Scrum 宗旨,回顾会议用以团队的自我提升,其重要性是显而易见的。对于软件开发来说,若不能够高效、进步和创新,则无疑是在走向灭亡。然而在实际应用中,回顾会议往往变得无所回顾,无可总结。渐渐地流于形式,甚至被一部分开发小组所抛弃,这样的例子屡见不鲜。我们团队也无例外地经历过这样的尴尬处境。对此采取了改进措施:回顾会议期间之所以无所回顾,究其原因每个成员对一个月内的众多细节根本无法一一回忆。所以在燃尽图旁新增三栏: Good、Bad 和 Ugly。团队成员在每个 Sprint 期间,可以就无论是技术、沟通还是管理方面的问题在这三栏中以即时贴的方式添加自己的意见,提倡的意见放在 Good 栏,改进意见放在 Ugly 栏,揭露弊病的意见放在 Bad 栏。即时贴所写的内容包括事件简述、日期、意见发表者。这样改进后,在回顾会议中就不会再无所回顾了。

(5) 坚持应用持续集成和测试驱动方法。

对于敏捷模式之一的 Scrum,持续集成(Continuous Integration)和测试驱动(TDD)同样是其两大基石^[9]。在大型复杂项目实施中,软件集成无疑是一个重要的问题。要降低项目的风险、确保产品的质量,尽早且频繁、持续地集成是一种事半功倍的方法^[10]。图 3 是持续集成的流程图。

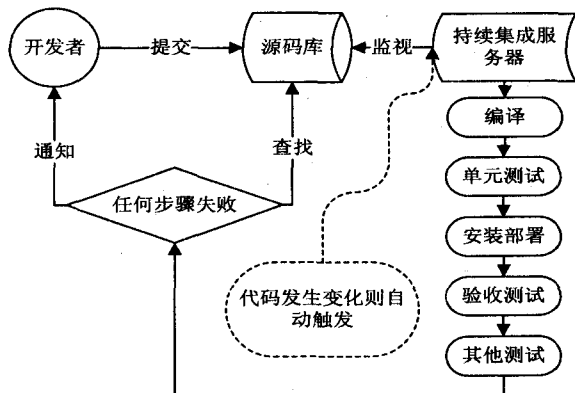


图 3 持续集成流程

使用测试驱动可以避免像传统方法那样对测试文档的依赖。通常用户需求的变更会使维护测试文档的成本与日俱增,这样的结果显然不够敏捷,也不符合 Scrum 方法的宗旨^[11]。如果使用测试代码来代替测试

文档,就会使问题变得简单多了。测试驱动通过对测试代码不断地重构来推动软件功能代码的编写和完善,不断增加软件新特性,直至实现全部软件功能,如图 4 所示^[12]。这种方法不仅保证了代码的正确性,也体现了灵活适应需求变更的敏捷性。

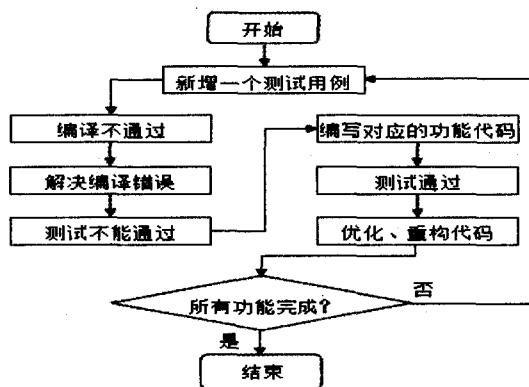


图 4 测试驱动流程

2.3 Scrum 方法的改进成效

经过改进 Scrum 之后,不但团队的生产效率得到了很大的提升,并且团队成员的积极性、工作效率、创新思维都得到了显著的提升。图 5 和图 6 是改进前后

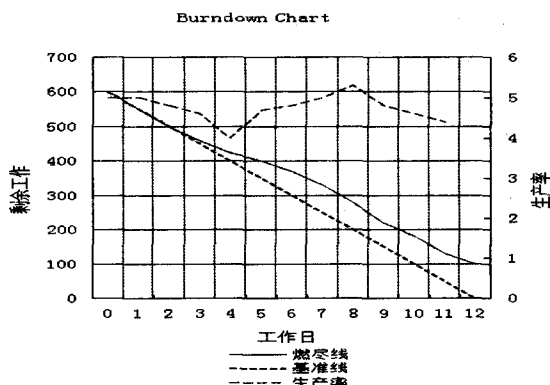


图 5 改进前的燃尽图

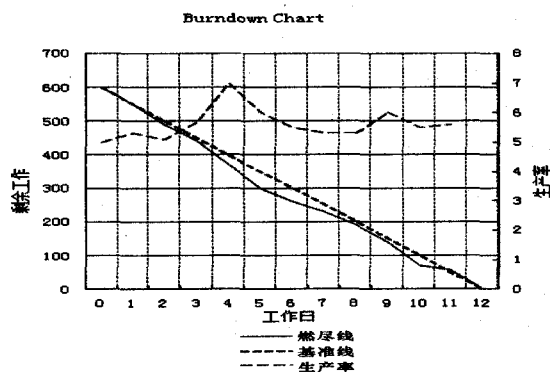


图 6 改进后的燃尽图

某两个 Sprint 的燃尽图。图表中的基准线表示在最理想的情况下每天应按时完成确定的等量任务,这时每天的生产率都是 5。燃尽线是实际工作的情况。从图

(下转第 104 页)

信息,提取数据并封装成 XML 格式的共享数据,发往给数据交换中心,交换中心将数据进行缓存并转发,路由模块调用 B 节点的 web services 接口传输数据。B 节点接收数据,根据其携带的 DataSwitchTrans_ID 值提取对应的映射表,根据表 3 中的数据匹配算法进行数据转换,利用算法输出的 SQL 语句进行数据库操作。交换过程结束后 tableB 的内容如下:

表 3 tableB(b)

UserID(int)	Name(String)
00001	Bjarne Stroustrup

5 结束语

文中详述了系统的结构以及适配器的实现技术,利用实例说明了数据交换的过程,系统能够为异构的数据交换提供有效的解决方案,数据交换的准确性与效率较高。下一步的工作是设计更好的数据流程,进一步提高系统的效率。

参考文献:

[1] 唐俊伟. 基于 Web 技术的数据交换平台的研究与设计[D]. 西安:西北工业大学,2006.

[2] 袁磊,李帅. 基于 WebServices 的异构系统信息交换的实现[J]. 计算机技术与发展,2008,18(12):8-10.

[3] 马云峰,王源. 基于 XML 和 WebService 的异构数据集成研究与实现[J]. 计算机技术与发展,2010,20(11):42-

(上接第 99 页)

中不难看出,改进前团队的生产率平均不超过 5,并且 Sprint 结束后还有部分任务完不成。改进后,不但按时完成任务而且生产率基本在 5 之上。

3 结束语

公司的文化、管理模式不同,开发者的经验不同等等都决定了 Scrum 方法不能被按部就班地套用。Scrum 宗旨是提倡灵活,因此在实际运用中的灵活改进并不悖于 Scrum 的原则。相反一味的借鉴模仿只会形似而神不似,导致适得其反的结果。“敏捷”本身就意味着适时地变化,应该在敏捷和不敏捷之间寻找一种合理的折衷方案。在实践中对敏捷模式进行适当的改进和灵活的运用,能够帮助我们更好地抓住敏捷开发方法的精髓,最终达到预期的目标和效果。

参考文献:

[1] Schwaber K, Beedle M. Agile Software Development With Scrum[M]. [s.l.]:Prentice Hall, 2001.

[2] Leffingwell D. 可伸缩敏捷开发:企业级最佳实践[M]. 李

46.

[4] 章明,许青松,沈锡军. 基于 XML 的数据交换共享平台模型[J]. 清华大学学报(自然科学版),2003,43(1):105-107.

[5] 李亚楠,刘连忠,贾毓星. 数据交换研究[J]. 计算机技术与发展,2008,18(2):5-8.

[6] 王艳敏,谢强,丁秋林. 基于本体和 WebServices 的数据交换平台[J]. 计算机技术与发展,2010,20(5):112-116.

[7] Philip G C. Software design guidelines for event-driven programming[J]. The Journal of Systems and Software,1998,41(2):79-91.

[8] 欧阳辉. 基于 LDAP 协议的异构数据源集成与信息共享关键技术研究[D]. 广州:华南师范大学,2007.

[9] Trinh Q, Barker K, Alhajj R. RDB2ONT: A Tool for Generating OWL Ontologies From Relational Database Systems[C]. [s.l.]:AICT/ICIW,2006.

[10] Xu Jiuyun, Li Weichong. Using Relational Database to Build OWL Ontology from XML[C]//International Conference on Computational Intelligence and Security Workshops. Washington, DC, USA:IEEE,2007.

[11] El-Khatib H T, Williams M L, MacKinnon L M. Using a distributed approach to retrieve and integrate information from heterogeneous distributed databases[J]. Computer Journal, 2002,45(4):381-394.

[12] 史立军,侯红,祁方民. 基于 Schema 的 XML 文档到关系数据库的转换[J]. 计算机技术与发展,2008,18(10):71-74.

冬冬,译. 北京:电子工业出版社,2009.

[3] 杨璇. 基于敏捷方法的科研管理系统的设计与实现[J]. 计算机技术与发展,2008,18(12):182-185.

[4] Pollice G. 敏捷起源以及敏捷创始人的传记[EB/OL]. 2009. <http://www.scrumcn.com/agiledev/html/?169.html>.

[5] Martin R G. 敏捷软件开发—原则、模式与实践[M]. 邓辉,译. 北京:清华大学出版社,2003.

[6] Kniberg H. 硝烟中的 Scrum 和 XP—我们如何实施 Scrum[M]. 李剑,译. 北京:清华大学出版社,2011.

[7] 钟杨,刘业政,马向辉. 小团队结对编程实践研究和重构[J]. 计算机技术与发展,2007,17(11):160-163.

[8] 张克强. 让管理层害怕的 8 个敏捷理解[EB/OL]. 2011. <http://www.scrumcn.com/scrumpc/html/?254.html>.

[9] 贝克. 测试驱动开发[M]. 孙方,译. 北京:人民邮电出版社,2008.

[10] 张扬,黄厚宽. 测试驱动开发及开发实践[J]. 计算机技术与发展,2006,16(5):75-77.

[11] 柳纯录. 系统集成项目管理工程师教程[M]. 北京:清华大学出版社,2009.

[12] 青瑞. 软件工程之全程建模实现[M]. 北京:机械工业出版社,2010.