

一种基于极限编程的实验室项目软件方法 LabXP^{*})

邵 飞 葛季栋 胡 昊 鲍钦迪 吕 建

(南京大学软件新技术国家重点实验室 计算机软件研究所 南京 210093)

摘 要 本文基于一种敏捷的软件方法——极限编程,结合实验室项目的特点,探讨了在实验室项目中进行软件开发的方法,提出了一种实验室项目软件方法 LabXP(Lab eXtreme Programming)。鉴于实验室项目的特殊性,这种方法是

以极限编程(XP)为基础的敏捷软件方法(Agile Software Method),在保证软件质量的基础上实现了快速开发。

关键词 实验室项目,软件方法,XP

An eXtreme Programming Based Laboratory Project Software Methodology——LabXP

SHAO Fei GE Ji-Dong HU Hao BAO Qin-Di LU Jian

(State Key Lab. for Novel Software Technology, Institute of Computer Software, Nanjing University, Nanjing 210093)

Abstract This paper discusses the characteristics of a special kind of software project——laboratory software project and how to develop it successfully. Based on the agile software methodology-eXtreme Programming, we present a new software methodology-LabXP, which adopts some key practices of eXtreme Programming and adapts to the development of laboratory software project. By using this methodology, teams can ensure the quality of the project and enhance the development speed.

Keywords Laboratory project, Software methodology, eXtreme Programming

1 引言

根据软件的生产者和生产目的的不同,可以把软件生产过程分为由软件企业进行的商业软件生产过程和由实验室研究人员进行的实验室项目软件生产过程。在实验室环境下进行的软件生产是一种迥异于商业环境下软件生产的特殊软件生产过程,有一些它固有的特征。它的固有特征造成了实验室项目对项目的软件工程质量略有忽视。为实现实验室项目的软件质量保障,我们需要好的软件过程方法对实验室项目的软件生产过程进行管理。通用的软件过程方法有 CMM、极限编程(eXtreme Programming, 以下简称 XP)等, CMM 常适用于大型的商业软件项目,而 XP 则在轻量型的商业软件开发方面应用很广。由于实验室项目的特点,我们发现 XP 比较适用于实验室项目,但同时由于实验室项目和商业项目的不同,XP 中的一些实践并不完全适用于实验室项目。根据对实验室项目特征的分析,结合实际工作经验,我们在采纳 XP 的一些关键实践、变更使用 XP 的一些关键实践的基础上,适当加入了其它方法,提出了一种适用于实验室项目的软

件开发方法 LabXP,并使用它开发了一个系统。

本文首先简要介绍了极限编程方法,随后介绍了一种特殊的软件开发项目——实验室项目,着重分析了它的特征。在前两部分的基础上,我们得出了实验室项目软件方法 LabXP,详细分析了 LabXP 的实践和 LabXP 的工作过程。其后给出了 LabXP 在一个实验室项目中的应用,最后得出了结论。

2 XP 简介

自 20 世纪 60 年代以来,先后出现了多种软件开发过程模型^[1,2]。20 世纪 80 年代末期, Kent Beck 和 Ward Cunningham 在他们的工作中运用了一种独特的理念,后来在 90 年代末期这一理念被称为极限编程(XP)^[3]。

XP 是一种轻量、高效、低风险、柔性、可预测、科学而且充满乐趣的软件开发方法^[4]。它是一种高度动态的软件开发过程,通过短周期发布、重构和持续集成来适应不断变化的需求。Beck 等对 XP 的阐述把它分为四个准则(value)、五条原则(principle)、四种活动(activity)、五种策略(strategy)及十二

*)本文得到国家 863 项目支持,项目编号:2001AA113090。邵 飞 硕士研究生,主要研究领域: workflow 技术,软件过程管理。葛季栋 博士研究生,主要研究领域:软件过程, workflow 技术,移动 Agent 技术。胡 昊 博士研究生,讲师,主要研究领域:软件过程, workflow 技术,移动 Agent 技术。鲍钦迪 硕士研究生,主要研究领域:软件工程、软件过程技术。吕 建 博士,教授,博士生导师,主要研究领域为对象技术、分布技术、移动 Agent 技术。

参 考 文 献

- 1 刘元宁,苑森森,朱晓冬. 警用辨识系统现场指纹分类与点模式匹配算法的研究. 计算机研究与发展, 2003, 40(6): 813~819
- 2 Senior A. A Hidden Markov Model Fingerprint classifier. In: Proc. 31st Asilomar conf. signals, Systems, and computers, 1997, 306~310
- 3 胡红俊,赵丽如,贺贵明. 一种改进的指纹分类方法. 计算机工程, 2002, 28(4): 109~110
- 4 Kamijo M. Classifying Fingerprint Images Using Neural Network; Deriving the Classification State. In: Proc. Third Int'l conf. Neural Network, 1993, 1932~1937
- 5 Karu K, Jain A K. Fingerprint Classification. Pattern Recognition, 1996, 29(3): 389~404
- 6 李晓红,贺贵明,贾国颖. 一种基于大型指纹数据库的检索方法. 计算机工程与应用, 2002(19): 203~208
- 7 AnilJain, LinHong, Bolle R. On-Line Fingerprint verification. IEEE Trans on Pattern Analysis and Machine Intelligence, 1997, 19(4): 302~313
- 8 杨利敏,杨杰,李钢. 基于指纹分类的模式匹配. 电子学报, 2003, 31(7): 1030~1034

条实践(practice)等几个方面^[4-6]。其中,XP的准则和原则决定了XP的开发思想,活动则是XP的基本软件开发活动,实践是保证开发活动成功进行的有效技术,策略则是保证实践取得成功的经验和探索方法。

XP的十二条实践是XP项目获得成功的保证,在XP中具有非常重要的地位。XP的一项实践是项目成员成功地执行任何活动的一项技术。Beck介绍了十二项实践,它们分别是:计划游戏、短发行周期、隐喻、简单设计、测试、重构、结对编程、集体所有制、持续集成、每周工作40小时、现场客户、编码标准^[4]。

XP的这些实践相互作用、互相支撑,但是在不同的情况下,根据各个实践对环境的适用性,可以灵活地采用这些实践。在下文中,我们根据实验室项目的特征,直接采用了适用于实验室项目的七项实践,另外变更采用了较适用于实验室项目的四项实践。

XP的适用范围有一定限制,它主要针对的是团队的程序员不超过10个,并在同一个地方工作的面向对象项目,任何规模适中、需要快速而灵活地交付高质量软件的项目,都可以采用XP。Beck认为在如下环境非常适合使用XP:

- (1)规模较小的团队(团队的规模不超过10人)。
- (2)获得反馈的时间很短的环境。
- (3)容易进行测试的项目。
- (4)可以接受XP文化的团队和组织。
- (5)程序员在地理上在一起的团队。

3 一种特殊的软件开发项目—实验室项目

实验室项目是指在大学或研究所的实验室研制开发、以验证科学研究新理论或新技术开发为目的的研究项目。它一般具有如下特征:

(1)缺乏真正的客户方。因为实验室项目基本上是为了验证实验室研究人员提出的某个科学构想或新理论、新算法的正确性,这些理论、算法或新技术实际上就成为实验室项目的需求。因此作为该项目开发方的实验室研究人员往往同时扮演着该项目需求方的角色。

(2)需求变动较大。实验室项目的目的是为了验证一个新的理论或技术,具有较强的创新性和探索性,这些新理论或新技术就是实验室项目的需求。而对这些新理论的研究总在取得进展或发生变化,因此实验室项目的需求也在变动。

(3)常以开发出原型系统为目的。实验室项目的开发团队以实验室的研究人员为主,每个研究人员既有研究工作也有开发工作,研究人员往往在初期的研究工作中投入大量精力,但是忽略了构造一个完整产品的工作^[7]。因此他们进行项目开发时,只要原型满足了验证他们的研究构想的目的,就认为工作结束了。而这种原型系统,距可以实际交付使用的应用系统还有一定的距离。当原有的研究构想发生变化或出现了新的研究思想时,就会对这个原型系统进行改动,而这种改动往往变动比较大。随着研究的深入,项目就会越来越复杂,有时就会出现项目看起来无法更改的状况。

(4)文档简洁有效。由于实验室项目需求变化比较大,预先的完整的规范、分析或设计并不能符合项目需求不断变化的情况,成员倾向于对原型系统做创造性的设计,因此有必要将项目成员从繁多的文档中解脱出来,在项目文档管理方面,

选择简洁有效的方式,而这正是XP所提倡的理念。在XP中,文档几乎不存在,只存在一些开发人员记录计划要完成的项目特征的索引卡片^[8],或者说,代码就是文档^[9]。

(5)项目呈越来越复杂的趋势。随着科学研究的飞速发展,新的技术不断出现,新理论、新算法也越来越复杂,实验室项目的规模也呈现出越来越大的趋势。以我们实验室为例,在以往的项目中,代码量是相对比较少的,而现在在一个普通的863的科研项目就已经达到了六万多行的代码量。同时,实验室项目内容都很新,较少有类似项目的经验积累,而且倾向于使用新技术、试验新技术,项目开发也就愈发复杂。

在大学或研究所里进行的实验室项目,和在公司里开发的商业项目相比,具有很多不同的特点,它们之间的比较如表1所示。

表1 实验室项目和商业项目的比较

	实验室项目	商业项目
客户方	往往由研究人员充当,少有真正的客户方	实际客户
项目需求	研究人员自身提出,具有变动较大的特点	由客户提出,需求较清楚
项目生产目的	验证某种研究构想	生产出商业软件产品
交付产品	原型系统,离产品化还有一定距离	实际应用的商业产品
开发团队组成	实验室研究人员或研究生	专业程序员
项目文档	要求简洁有效	文档详细、全面
项目开发经验	每次都开发新的原型系统 项目多属于原创性项目 可复用资产相对较少	很多时候有类似项目的经验积累 可复用资料相对较多
技术应用	倾向于使用新技术、试验新技术	倾向于使用成熟技术

4 基于XP的实验室项目软件方法 LabXP

实验室项目的特点要求采用一种特殊的软件过程方法来实现软件质量保证。在实践中我们发现XP是一种较适用于实验室项目的软件过程方法,结合实验室项目的特点,我们对XP方法进行了改造使之能够完全适应于实验室项目,得到了一种基于XP的实验室项目方法 LabXP。

4.1 LabXP 简介

LabXP是根据实验室项目的特征,以XP的一些关键实践为基础构成的、适合由实验室开发人员组成的小团队进行软件开发的软件开发方法。这个软件方法应用XP的主要实践来进行软件开发,团队的人员一般不超过10人,团队成员之间紧密结合,每个人在团队中都承担着重要的角色,整个团队作为一个高效的开发整体完成开发任务。在LabXP中直接采用了七条XP实践,修改并采取了五条实践,对不适用实验室项目的一条实践不予采用。表2所示为XP中的实践在LabXP中的采用情况。

表2 LabXP中XP实践采用情况

直接采用	变更采用	不采用
简单设计、编码标准 结对编程、测试驱动 重构、小版本发布、持续集成	计划游戏、现场客户、隐喻 集体所有权	每周工作40小时

在LabXP中直接采用的七条XP的实践都是非常适用

于实验室项目的特征。它们的适用性如表 3 所示。

表 3 LabXP 中的实践与实验室项目的适用性

LabXP 中的实践	实验室项目特征
简单设计	需求变动大 文档管理讲究简洁有效
编码标准	开发团队由实验室研究人员或研究生组成, 团队成员风格各异,必须有一个编码标准
测试驱动、结对编程、 重构	需求变动大 开发新项目,背景经验较少,可复用资料少 倾向探索新技术,风险较大
小版本发布、持续集成	产生验证某种构想的原型系统 由于需求变动大,原型系统改动也较大

在 LabXP 中,我们之所以变更采用了 XP 中的四条实践,放弃了一条不适用的实践,是由实验室项目的特征决定的。

首先,由于很多实验室项目并没有真正的客户,在 LabXP 中,由一位对项目要实现的算法或新技术比较清楚的研究人员充当现场客户,我们称之为客户代表,和开发团队共同决定系统需求并独立完成验收测试的工作。

其次,在 LabXP 中,我们参考了 SCRUM 方法^[10],使用用例代替了 XP 计划游戏中的用户故事,我们认为用例比用户故事更能准确描述项目的需求。由于在 LabXP 的开发过程中可以容易地适应需求的变化,为了简单性,我们只使用简单的用例,不考虑用例的扩展和包含。

再次,对于规模比较小的实验室项目,我们采用隐喻来进行软件开发。而由于实验室项目的规模有越来越大的趋势,对于规模较大的项目,为避免单纯采用隐喻可能带来的混乱,采用面向模式的架构方法,这种架构方法也是适合变化的,在后期的迭代过程中逐步精化该架构。

最后,对于集体所有权,我们采取了“集体所有、修改告知”的策略,即所有开发人员对所有代码都熟悉,但开发人员要修改某一段代码时,需先告知该段代码的原编写人员,以期避免造成代码的混乱。LabXP 中变更采用的实践和 XP 中实践的对应情况如表 4 所示。

表 4 LabXP 中变更采用的实践与 XP 中实践的对应情况

XP	LabXP
现场客户	客户代表
计划游戏中的用户故事	简单用例
隐喻	隐喻、面向模式的架构
集体所有权	集体所有、修改告知

之所以不采用每周工作 40 小时这条实践,是因为相对于其它实践来说,每周工作 40 小时是不适合于实验室环境的。实验室的研究人员既要从事理论研究探索工作又要从事具体项目开发工作,每周工作 40 小时的工作时间是远远不够的,他们需要更多的工作时间来进行研究,且受到学校课程的限制,工作时间的安排具有一些不规则性。同时,实验室项目每次都开发新的原型系统,较少有同类项目经验的积累,可复用资产较少,并且项目成员倾向于采用新技术,所面临的开发难度和风险也比较大,很多时候为了达到预期的目标,必须延长工作时间。

4.2 LabXP 的工作过程

LabXP 是小的实验室团队进行软件开发的开发过程,

Brooks 曾经提到,独立工作的小的团队是更有效的^[11],当实验室项目比较大时,我们采取的策略是将大的团队分成可以独立工作的小的团队,各个小的团队采用 LabXP 进行软件开发。在各个团队中必须有一名研究人员对项目要验证的算法和新技术充分理解,由他来充当客户的角色,和团队中其它人共同进行项目开发。

所有的项目都有初始计划阶段,在此阶段,项目团队和客户共同决定软件要满足的需求(在 LabXP 中,我们用用例来表示项目需求,将每个功能需求写成一个用例并把它命名、写在一张卡片上),并决定软件要分成几个阶段来发布、每个发布版本满足哪些需求。团队对每个版本要满足的需求进行估算,和用户共同决定这个版本的发布计划。在每个发布计划中,团队首先将这些需求按照优先级从高到低的顺序做出待完成任务列表 backlog^[10]。在后序的工作中,将对 backlog 中的这些任务进行迭代开发,在所有的任务完成并通过验收测试后,发布新的版本。

在初始计划完成以后,首先统一编码标准,然后进行一系列的开发过程即迭代过程,增量地发布新的系统版本。每个迭代周期大约持续一到两周的时间,在最后一个迭代结束后,结束一个发布计划,发布一个新的小版本。

LabXP 的工作过程如图 1 所示。

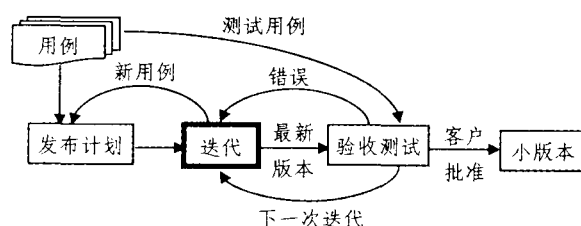


图 1 LabXP 的开发过程

4.2.1 如何进行计划与估算 通常,由项目开发团队和客户共同决定发布日期,在发布日期定下来后,项目团队和客户必须在一起商议,提供在产品的第一版发布中要包含的最有价值的特征。客户将这些特征进行优先级排序,项目开发团队对它们进行估算。客户和开发团队必须对这些特征达成一致意见,如果项目团队不能提供所要求的特征,双方必须共同商议,提供一个削减的功能集^[7]。

在协商完成后,按照特征的优先级,编制待完成任务列表,开发团队会根据他们所使用的开发环境和开发工具,估计出这些任务的风险。一般情况下,将风险大、优先级高的任务放在前面的迭代周期中,以便任务难度增大时有时间处理。将风险较小、优先级较低的任务放在后面的迭代周期中。

4.2.2 LabXP 的一次迭代过程 在每次迭代开始前,团队更新待完成任务列表 backlog 并重排任务优先级,选择这次迭代要完成的任务,开始本次迭代。每次迭代产生出一个可见的、稳定的、可交付的产品,这个产品是在上次迭代结果的基础上增加了新的功能。每次迭代是一个时间固定的过程,团队可以减少在这次迭代过程中要增加的功能,但不能推迟迭代结束时间。

团队在迭代过程中,每一至两天开一次短会,所有在项目团队的实验室人员都要参加,在短会上,每个成员都要说明自己已经做了什么,遇到了什么问题以及下一步要做的工作。在短会上,重点强调在开发过程中遇到的问题,让所有人员都了解存在的问题,问题的解决由熟悉该问题领域的人员在会

后商议解决。

在迭代开发过程中,只针对要完成的任务做简单设计,采取测试驱动的结对编程方式进行编码,在编码过程中,选取任何合适的时机进行重构。LabXP 的一次迭代过程如图 2 所示。

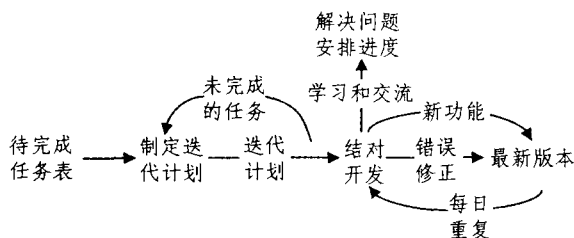


图2 LabXP 的一次迭代过程

4.2.3 测试和发布 每次迭代都会生产一个最新版的产品,由充当客户的研究人员进行验收测试,如果验收测试没有通过,则继续本次迭代,修正错误,直到通过验收测试。否则,从待完成任务列表中选择任务,开始下一次迭代。在最后一次迭代通过验收测试,征得客户同意后,发布本版本,完成一次发布计划。

5 LabXP 在一个具体实验室项目中的应用

5.1 项目概况

该项目是基于 CMM 软件质量保证平台中的软件估算子系统,该子系统的目的是研制出一个为软件企业提供估算流程、使用一些常用的估算算法为软件企业在软件生产中的各个对象、各个阶段进行估算,以达到帮助企业控制软件生产过程,提高软件生产质量和效率之目的。

该软件估算系统的开发周期只有一个月,并要求实现和基于 CMM 的软件质量平台无缝集成。该系统的开发团队由 5 人组成,其中一人充当客户代表,和团队共同决定该项目要满足的需求以及发布计划。

由于该系统开发时间短、任务量大并且开发团队小,我们决定采用 LabXP 进行开发。

5.2 使用 LabXP 进行项目开发的过程

首先客户代表和开发人员讨论项目需求,进行需求分析,开发团队确定该系统要满足的需求,编写用例并把每个用例写在一张卡片上。开发人员和客户代表共同确定了第一个小版本要实现的系统特征,并决定以一个月的时间实现第一版的发布,以每周为一个迭代周期。根据第一版要实现的特征,编制了待完成任务列表 backlog 并决定在每个迭代周期中实现哪些特征。

在第一个迭代周期中,我们从待完成任务列表中挑选了一些最有价值的用例作为第一个迭代周期要完成的特征,对它们进行了简单设计和实现。

在迭代过程中,我们使用统一的编程环境 Visual C++, 采取结对编程的方法。团队开发人员由三个二年级研究生和一个一年级研究生组成,分成两对,分别对一些特征进行实现。采用两个人讨论,其中一人编写代码另一人审查代码的方式。当结对的双方意见不一致时,我们采用 Miller^[12]所提供的方法来解决矛盾。结对使得双方把注意力都集中在项目上,并促进结对双方之间的交流,产生更具有创造性的思路。

在结对编程的开发过程中,我们每实现一个新特征都进行严格的测试。计划使用 CppUnit 作为单元测试工具,但由

于大家对 CppUnit 都不熟悉,加之时间紧,仍然采取了常规的测试方式。测试保证了功能的正确性,在进行测试代码编写的同时,也有利于我们确定程序代码的结构。在开发过程中,结对双方会根据测试的情况或项目新增加的特征重构他们所编写的代码,每次重构完毕,都进行测试,测试保证了我们重构代码的正确性,给了我们重构的勇气和信心。

在迭代过程中,我们每两天开一次短会,交流进度及遇到的问题并在会后解决遇到的问题。在迭代过程中,每增加一个新的重要特征,都由客户代表进行验收测试。

在第一次迭代中,我们编制了系统核心的功能并确定了它与环境的交互,在随后的迭代过程中,我们不断增加用例和功能,使系统趋向于更多功能和更复杂。在最后一周迭代结束后,我们顺利发布了系统的第一版。测试驱动的编程过程如图 3 所示。

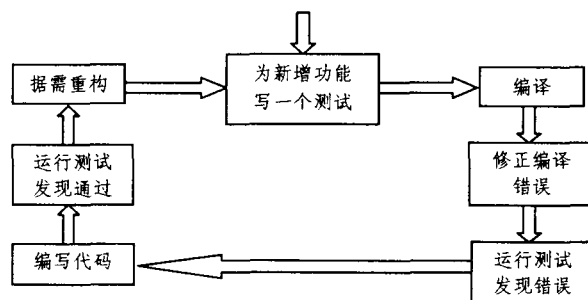


图3 测试驱动的编程过程

结论 本文对于适用于实验室项目的软件方法进行了初步探讨,提出了一种基于 XP 的实验室项目软件方法 LabXP。LabXP 是一种增量开发的敏捷软件过程,它以极限编程技术为基础,采用了 XP 的几个关键实践,为需求不确定、变化快、面向研究和技术探索的实验室项目开发提供了一种有效的软件开发过程。它有利于促进实验室项目的成功,并对软件工程方面的理论探索有较大的好处。对于如何有效应用 LabXP 进行大规模、大团队的实验室项目以及异地实验室合作的实验室项目进行软件开发,是我们今后进一步研究的方向。

参考文献

- 1 Sommerville I. 软件工程(第 6 版). 机械工业出版社, 中信出版社, 2003. 27~44
- 2 Pressman R S. 软件工程:实践者的研究方法(第 5 版). 机械工业出版社, 2002. 14~32
- 3 Cockburn A. 敏捷软件开发. 人民邮电出版社, 2003. 1~9
- 4 Beck K. 解析极限编程—拥抱变化. 人民邮电出版社, 2002. 29~125
- 5 Beck K, Fowler M. 规划极限编程. 人民邮电出版社, 2002. 19~126
- 6 Wake W C. 探索极限编程. 人民邮电出版社, 2002. 3~87
- 7 Wood W A, Kleb W L. Extreme Programming in a Research Environment. LNCS 2418, Berlin: Springer-Verlag, 2002. 89~99
- 8 Muller M M, Tichy W F. Case Study: Extreme Programming in a University Environment. www.ipd.uka.de/~muellerm/publications/iesec01.pdf
- 9 Grenning A. Launching Extreme Programing at a Process-Intensive Company. IEEE Software, 2001, 18(6): 27~33
- 10 Rising L, Janoff N S. The Scrum Software Development Process for Small Teams. IEEE Software, 2000, 17(4): 26~32
- 11 Brooks F P. 人月神话. 清华大学出版社, 2002
- 12 Miller R W. When Pairs Disagree, 1-2-3. LNCS 2418, Berlin: Springer-Verlag, 2002. 231~236