

敏捷软件开发的认识偏误与推广瓶颈浅析^{*}

匡松 周启海 陈森玲 黄涛

(西南财经大学经济信息工程学院 成都 610074)

摘要 本文概述了敏捷开发的五大架构主导性理念与四大核心价值观,揭示了敏捷软件开发方法的思想本质;分析了人们对敏捷软件开发的某些认识偏误与进一步推广应用的主要瓶颈,并提出了解决这两大问题的若干对策建议。

关键词 敏捷软件开发,认识偏误,推广瓶颈,对策建议

Tentative Analysis for the Recognizing Deviations and the Spreading Bottleneck of Agile Software Development

KUANG Song ZHOU Qi-Hai CHEN Sen-Lin HUANG Tao

(School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074)

Abstract This paper summaries the five main guiding principles and four core values of agile software development, reveals the thinking nature of the methods of agile software development, analyses the deviations of cognizing agile software development and the main spreading bottleneck of spreading Agile Software development, and puts forward some countermeasure proposals to overcome the two problems.

Keywords Agile software development, Recognizing deviations, Bottlenecks, Countermeasure proposals

1 引言

eXtreme Programming(简称 XP)、Adaptive Software Development(简称 ASD)、Crystal Methodologies、Feature Driven Development(简称 FDD)等轻载方法,是软件开发的新技术。这些轻载方法的一些创始人于 2001 年 2 月在美国犹他州成立了 Agile 联盟,并将轻载方法正式更名为 Agile 方法。这标志着敏捷软件开发思想的诞生,同时引起了软件业界的重视,现已经得到业界内国际范围的认可、实践和应用。2006 年 6 月,中国最大的 IT 技术服务平台 CSDN 与 Thought Works 公司借 Martin Fowler 先生再次访问中国的机会,共同举办首届“敏捷中国”开发者大会,并在上海举办“敏捷专家圆桌论坛”,向中国软件开发者传播业内领先的敏捷软件开发思想与实践,引发了敏捷软件开发的新一轮研究热潮。本文试简要剖析敏捷软件开发认识上的偏误与应用推广中的瓶颈,以进一步促进敏捷软件开发方法的普及、应用、推广。

2 敏捷开发概述

敏捷软件开发方法的先进性表现为其主导性理念、核心价值观、特色性优势,得益于 Kent Beck《解析极限编程——拥抱变化》中提出的著名敏捷宣言,敏捷思想得到具体的诠释。

敏捷软件开发理念的主导性理念可概括为:1)价值理念——描述“软件开发应该是什么样的”,而 XP 是以沟通、简单、反馈和勇气四个价值为基础;2)法则理念——用于判断一项实践是否能在 XP 的环境中成功使用的依据,而 XP 有五条法则(即快速反馈、假设简单性、递增更改、提倡更改、优质工作);3)活动理念——软件开发的基石以支持敏捷开发实践,而 XP 是以编码、测试、倾听和设计等活动为核心;4)实践理念——项目成员用之以成功执行敏捷活动的技术(Beck 介绍了落实进度、计划游戏、每周工作 40h 等 12 项实践);5)策略

理念——用于实现上述目的的经验等。

敏捷软件开发主导性理念要追求的敏捷过程核心价值观可简述为:1)个体和交互胜过过程和工具;2)可以工作的软件胜过面面俱到的文档;3)客户合作胜过合同谈判;4)响应变化胜过循环计划^[1]。

敏捷软件开发方法的特色性优势主要体现在:1)用户价值驱动,用户全程参与,交互主导开发,敏捷应对变化;2)短周期开发,小版本发布,尽早创价值——它的软件开发宗旨是“越早交付上线就能够越早地为企业提供价值,也就能越早地体现出该软件及其开发的工程价值与社会价值”。

敏捷软件开发方法的主导性理念、核心价值观、特色性优势实际上反映了敏捷软件开发方法的思想本质——必须确保软件开发者对其软件用户群的“主动服务、积极沟通,加强合作、良性互动,螺旋开发、递进更新,尽早创利、和谐双赢”。

3 敏捷软件开发的若干认识偏误

敏捷软件开发的新奇理念、独特方法、特殊实践,使人们目前对它尚存有若干认识上的偏误。

3.1 敏捷开发排斥计划设计

人们从工程学科中引入计划设计,以解决软件失序引起的软件质量问题。这种计划设计方式事先只仔细考虑重大问题,从高层抽象上设计软件而不涉及其代码编写,可抛开编程细节,考虑更广范围,从而避免了引发软件失序的诸多随机应变性决策。等到设计完成后,再交由单独的工作组去构造,而构造者只要遵照设计意图、遵从设计要求,就可以构造出良好系统。但是这种计划设计也存在一些缺陷(例如,它不可能把编程时需要处理的所有问题统统考虑到,因而不可避免在编程时对设计存有质疑;设计者与构建者,其技能上的差距、技能认识上的差异、工作上的分离、时间上的分置等,可能导致软件失序;更不待说存在“客户意愿多变,用户需求难定”等难

匡松 教授,硕士生导师,主要研究方向:计算机应用。周启海 教授,博(硕)士生导师,主要研究方向:财经计算、算法研究与实现、计算几何、同构化信息处理等;陈森玲 硕士研究生,主要研究方向:计算机应用;黄涛 讲师,主要研究方向:计算机应用。

题)^[2]。初次接触敏捷开发的人易认为“敏捷方法的优势在于应对变化,适应变化自然也就排斥计划”。其实,这纯属误解。事实上,敏捷开发并不排除计划,只不过其计划不是由架构师单独设计,而某些简单或简明(例如已是开发人员共识者)的计划设计也无需写成正式文档。实际上,在开发过程中每一小周期开发都要进行计划设计,而多个小周期设计开发的不断循环,就使所开发的敏捷软件系统呈现“从简单到复杂、从低级到高级,逐级渐使产品成形”的特色与优势。自然,演绎这个过程并非易事,故亟待改进或创新可支持敏捷开发(例如敏捷化测试、开发、重构和自动化)的相关新技术、新设备。

3.2 敏捷开发排斥文档、图表

受传统软件工程的影响,在软件开发中必须编制详细的文档说明、用于验证各个阶段或者将其标准化的图表(如UML),以利软件开发之间的交流与合作。不过,有些人认为文档、图表有帮助,而有些人并不完全认同,并认为它常有“一方所认定的文档、图表,另一方可能按自己想法去理解、实施”的潜在危险。此外,某些文档、图表往往和重量级过程相关。此时去制定这些并无多大帮助的文档、图表,不能不说是浪费时间。而敏捷开发认为客户需求是不断变化或者不能完全确定的,为此制定详细的需求文档是浪费精力的。敏捷开发者更愿意采用“一步步地迭代,加上客户一次次地持续参与,一级级地最终逼近客户目标”的理想模式。尽管如此,敏捷开发是否不需要文档、图表了呢?当然不是。编制文档、绘制图表的目的在于帮助理解设计,而改变设计未必意味着改变文档、图表。而人们通常易犯“贪大求全”毛病,试图使文档、图表更为全面(例如在文档、图表上标出了所有的类、属性和操作等),而过分详尽的图表往往难与代码保持同步。敏捷开发并不排除文档、图表在程序开发中的交流作用,但是奉行“刚好够用”原则;据此在实施敏捷开发时,追求文档、图表的“适可而止少而精,必要重要有实效”,主张:可以和应当使用文档或图表,但必须简单扼要、突出主题、更新同步(指能跟上代码更新速度),并应丢弃已确认无用的文档、图表,以确保交流方便、开发高效。因此,敏捷开发排斥文档、图表的说法并不成立。

3.3 敏捷开发实践不能适应

敏捷开发的实践要求比传统开发模式为高,故只熟悉传统开发模式的程序员要适应敏捷开发的实践方式,确有一定难度。

例如:敏捷开发中应用最为广泛的XP编程有很多部分属能动性实践(如口头交流、结对编程等)。敏捷方法提倡的首要交流方式,并不是代码或者图表,而是目前开发人员往往很难理解和做到的口头交流。口头交流中,客户的参与尤其重要。因为在敏捷开发过程中,客户角色和权力与传统软件工程有很大的不同。几乎可以说软件开发的过程控制全在客户手里,由客户决定开发的方向和进程。在开发过程中,客户可以选择那些将会带来业务价值的内容形成用户需求,从而决定“要做什么,先做什么,后做什么;指定测试环节,检验系统功能;判定所完成系统是否达到预期目的”;客户有权负责整理规划,了解什么可以实现何时实现及实现的成本,查看运行系统的情况,更改计划更换功能和改变工作的优先级等;客户不用等到系统做完便可以在每一个环节进行测试使用^[3]。客户与程序员间的交流,发生在客户定义发布和迭代规划期间,整个开发过程需要客户全天候与项目组在一起,以便及时解答问题。因为事实证明交流可使程序员避免猜测代码,从而使得编程工作进行得更为顺利。当客户无法到达现场时,应采取一些措施(如尽力找本地人代表客户),尽可能地让真

正的客户参加现场规划会议,访问客户,经常性向客户发布代码等,因为这是应付的必要代价。

又例如:敏捷开发中的结对编程也颇受争议。它要求所有的产品代码都是由结对的程序员使用同一台电脑共同完成的;结对的关系每天至少改变一次,以便于每个程序员在一天中可以在两个不同的结对中工作;在一次迭代期间,每个团队成员应该和所有其他的团队成员在一起工作过,并且他们应该参与了本次迭代中所涉及的每项工作。然而现行管理人员把编程人员看成稀缺资源,不愿意让“两个人干一项工作”。编程在传统上也被作为一种孤立的活动来进行教授和付诸实践,许多有经验的程序员也不愿意和其他人一起编程。事实上,老练的结对程序员形容结对工作“速度要快两倍以上”,而且定量的证据证明设计结果更好,生成的代码更简单、更易于扩展。因为结对编程的优势在于:许多错误在被输入时而不是在QA测试中或者在实地就被察觉,同期进行的“集思广益”和“配对接力”使得设计更好、代码长度更短、团队解决问题更快,参加人员在“视线范围”内学习到的东西更多,项目最终有多人理解了系统的每一部分,参加人员学会一起工作、更频繁地一起讨论,提供了很好的信息交流和团队动态,最后的缺陷量低得令人满意,而它极大促进了专业知识在团队中的传播^[3]。

事实上,尽快掌握敏捷开发实践所带来的好处,远远超过适应敏捷开发实践的困难。因此,人们绝不能误认为不能适应敏捷开发实践,因为只要“端正认识,调整心态,积极应对”,就一定不难逐步适应它。

3.4 测试先行无法理解

测试驱动开发(TDD, Test Driven Development),是敏捷开发中的一项核心实践和技术,也是一种新型软件设计方法论。其基本原理是在开发功能代码之前,先编写单元测试用例代码,测试代码可确定需要编写什么样的产品代码。很多开发人员在实施TDD时,往往会有这样的问题:他们或者认为自己不懂测试,无法根据需求抽象写出测试框架;或者认为自己是编程高手,自己的程序不会有问,不需要浪费时间去写测试代码;或者认为兼顾测试工作,会分散精力或加重负担;写出来的测试框架可能无法保持稳定和重用,难以控制程序缺陷等等。事实上,更多的程序员还不习惯没有概要设计和详细设计文档规范下的编程,认为“没有文档,自己和他人都很难理解和使用别人的代码”。他们更习惯以串行模式进行工作:做好一个模型以后,再做下一个模型;或者所在小组负责模型的设计或文档的编写,完成后才交由另外的小组完成后续开发工作^[4]。

但传统开发方法提供的阶段性详细文档往往会分散开发人员应付变化需求的更多精力,也难以保证文档与代码的同步。故TDD把测试代码框架和用例简化为编码的最好说明书,也兼作文档组成部分,而其测试更不需要开发人员学习新的脚本语言。TDD的重要目的不仅仅是测试软件,测试工作保证代码质量仅仅是其一部分,而且是在开发过程中帮助客户和程序员去除模棱两可的需求。TDD首先考虑使用需求(对象、功能、过程、接口等),并只编写测试用例框架,对功能的过程和接口进行设计,而其测试框架还可持续进行验证。这样,TDD建立的系统开发质量保证框架,为团队开发更带来高效ROI。因此,测试先行是不难理解的。

4 中国推行敏捷软件开发的瓶颈分析与对策建议

瓶颈1——软件开发模式的传统习惯。相关研究与统计
(下转第303页)

原型系统,着重讨论了各个部分功能模块的设计思想和主要目的。

参考文献

- 1 林晓东,刘心松. 文件系统中日志技术的研究. 计算机应用,1998, 118(1)
- 2 徐著. 基于数据挖掘技术的人侵监测模型:[博士学位论文]. 中科

院高能物理研究所,2001

- 3 赵小敏. 基于日志的计算机取证技术的研究及系统设计与实现 [D]:[硕士学位论文]. 浙江工业大学,2002
- 4 Han Jiawei, Kamber M. 数据挖掘—概念与技术. 范明,孟小峰译. 北京:机械工业出版社,2001
- 5 屈定春,林原. 一种新型的数据库应用—数据采掘. 计算机应用研究,1996(6)

(上接第295页)

资料表明:1)从收入构成上看,2005年中国软件产业的总收入达到了3900亿元(其中国内市场贡献率为90%),比上年增长将近40%(其中软件出口为5%,软件外包为2%);未来中国,整个软件业收入将以20%增长。2)从行业结构上看,2005年中国软件产业整体收入中,53%来自软件产品、34%来自系统集成、13%来自软件服务(在国外正好倒过来),中国软件产品增长率是35%、软件服务增长率是66%;未来中国,服务的比例会越来越高,最终会跟国际产业结构接轨。这说明“中国软件市场潜力巨大,敏捷软件开发需求迫切”,因为中国现行软件开发模式仍较偏重于传统软件开发模式(例如瀑布模型、结构化方法、原型法、面向对象方法等),不能适应潜力巨大的软件市场需要,从而亟需全国性倡导软件开发模式的创新,大力倡用包括敏捷方法在内的新型软件开发方法。

瓶颈2——软件交付速度的观念陈旧。有数据表明,从交付的功能和时间来看,中国前两年实施ERP的平均上线速度约为国外6倍。这首先要归因于国内企业利用自己后发优势(例如汲取了国外同行经验教训而少走弯路,可通过改变自身业务流程去适应所购打包软件来缩短定制化过程,很多行业用户无计算机系统的历史遗留大包袱等);其次也不排除其它非正常因素影响,例如行政干预、业务部门片面要求上线速度(比如硬性规定什么时候上线)等。但软件开发三要素——速度、成本和质量,相互影响、彼此制约、密不可分。这种片面追求速度,必然使开发成本增加,而软件系统的高效性、灵活性、扩展性、鲁棒性等下降。因此,应改变传统软件交付速度老观念,应在项目范围、进度上有适度弹性,要倡用敏捷软件开发方法,以兼顾速度、成本和质量。

瓶颈3——软件价值实现的理念落后。迄今为止,仍有不少人认为:1)软件的功能就等于价值;软件功能越多、系统越复杂、解决问题越多,价值就越大;但绝非功能越多越好,因为用户根本不会去用那些多余功能,而多余功能却会使维护成本激增、可扩展性降低。2)软件一旦上线,就实现了全部价值。但软件上线仅意味着当前生命周期的结束,也预示着下一生命周期的开始。一个优秀软件的总体价值,寓于它的多个生命周期中。据统计,国内企业IT投资,其80%用于新产品的开发,20%用在现有系统的扩展(在国外恰好相反)。因此,需转变传统软件价值实现的落后理念,要提倡软件价值的长效实现观,应倡用敏捷软件开发方法。

瓶颈4——现行从业人员素质培养的机制落后。软件产业是日新月异、蒸蒸日上的知识经济型高新技术行业。它的软件从业人员(例如在职程序员等)素质必须实行终生培养制,只有这样才能适应和满足软件产业生存和发展的需要,因为舍此将无以应对软件需求的高品质、软件技术的高难度、软件开发的高质量、软件发展的高速度。但中国现行软件从业

人员素质的传统培养机制,往往落后于软件产业发展的客观要求,使软件从业人员素质不能更好地满足软件开发、更新、维护的新需要。因此,中国需对现行从业人员素质培养机制的创新,通过各种手段与方式,积极建立现行从业人员素质培养的新式长效机制,以确保软件从业人员素质和软件科学技术进步能彼此适应、与时俱进。

瓶颈5——后备从业人员素质培养的模式脱节。高等院校、中专学校相关专业的学生们,是软件产业的从业人员后备大军。目前国内外对IT人才培养均较为重视,但从目前国内相关专业的软件专业技能教学的教学内容、教学方法、教学模式等来看,由于计算机教育学研究长期不足,致使大部分高等院校、中专学校相关专业现行教学,都处于“先从面向过程入门,再转向学习面向对象”的滞后脱节状态,未能推行和实现“使面向过程与面向对象实现协调与统一”的科学教学思想,更未能提倡和实现与软件开发技术进步保持同步并为之配套的教材体系和教学思想等。因此,应加强计算机教育学研究,促使国内外高等院校、中专学校相关专业加大教育改革和创新力度,争取更及时、更平稳、更有效地适应软件界的变化与发展。

瓶颈6——支持敏捷软件方法的开发工具、集成环境之匮乏。与支持传统软件方法的开发工具、集成环境相比,人们(包括软件产业的现行从业人员、后备从业人员)深感支持敏捷软件方法的开发工具、集成环境之严重匮乏。众所周知,缺乏开发工具和集成环境支持的软件开发方法,其应用往往困难,其推广常常不易。敏捷软件开发方法的应用与推广,离不开支持它的开发工具和集成环境,需要研究和推出更成熟、便捷、人性化、可视化的开发工具、集成环境,以及在此类开发工具、集成环境支持下进行敏捷软件开发的成功典型案例库。为此,教育界和产业界应紧密结合,实时互动地提供敏捷开发典型案例库,进而推广到更宽广领域,供人们认知和实践敏捷思想参考,促进教育界和产业界的高素质人才的供需平衡。

结论 敏捷软件开发是优势突出、大有可作为的软件开发新方法。但是还应主动消除人们暂时对敏捷开发的若干认识偏误,并及时采取对症下药、切实可行的有效措施,疏通各个推广瓶颈,中国的敏捷软件开发一定会迎来蓬勃发展的春天!

参考文献

- 1 Beck K. 解析极限编程——拥抱变化. 唐东铭译[M]. 北京:人民邮电出版社,2002
- 2 Succi G, Marchesi M. 极限编程研究. 张辉译[M]. 北京:人民邮电出版社,2002
- 3 Newkirk J, Martin R C, 王钧译. 极限编程实践. [M]. 北京:人民邮电出版社,2002
- 4 王晓毅. TDD:金字塔上神像的光芒. [J]. 程序员,2006(09)