

小团队结对编程实践研究和重构

钟 扬,刘业政,马向辉

(合肥工业大学,安徽 合肥 230009)

摘 要:结对编程是两个软件开发者在同一台电脑前一起工作的一种编程实践,是极限编程方法的基础。对于软件产品的开发,人的知识是项目成功的重要因素。在组织的软件过程改进中,合适地引入结对编程,有利于知识的共享并营造团队合作的氛围,进而改善产品质量和缩短完成时间。不同规模的组织采用的软件过程的不同,对结对编程的引入会有不同的影响,所以在实践中结对编程的重构是必需的。基于小型组织和敏捷过程的项目实践,给出了一个切合实际的结对编程重构方案,进而提高小团队的生产效率。

关键词:结对编程;极限编程;敏捷方法;软件过程;重构

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2007)11-0160-04

Empirical Study and Refactoring on Small Team Pair Programming

ZHONG Yang, LIU Ye-zheng, MA Xiang-hui

(Hefei University of Technology, Hefei 230009, China)

Abstract: Pair programming, where two software developers work together at a single computer, is the basis of extreme programming (XP). In development of software products, human's knowledge is the key factor in any successful project. In organization's software process improvement, the proper introduction of pair programming benefit to knowledge sharing and atmosphere constructing of team collaboration, and then improve the product quality and short the completion times. It's different influence on introduction of pair programming in different size of organization where different software process was used, so that some refactoring of pair programming must have in practice. For getting higher productivity, and based on project practice of a small organization and agile process, this paper proposes a practical pattern of pair programming.

Key words: pair programming; extreme programming; agile methods; software process; refactoring

0 引 言

在过去的几年里,相对于逐渐没落的传统软件方法,一大批以“敏捷”为旗帜的方法涌现出来。极限编程(Extreme Programming, XP)是其中广为流传的代表方法。此外,还有 Scrum, FDD, DSDM, Crystal, Adaptive Software Development 和 Lean Development 等方法^[1]。

结对编程是极限编程的一个重要原则^[2]。在传统开发过程中,每个开发人员负责系统的一部分开发任务,各自分工,互不干扰。而在结对编程的情景下,是两个开发人员结为一对,来共同完成同一个开发任务。

两人在同一台计算机面前进行编程活动,一个同伴使用鼠标和键盘来编码,另一个同伴观察代码并考虑设计问题。一方发现问题时,暂停编码工作,双方讨论解决。结对者的角色根据需要来不断调整和交换,包括与别的结对组交换成员。这种结对方式可以让每个成员对项目有一个整体的认识,并且有利于团队建立起良好的合作和学习氛围。极限编程的其他几项重要原则包括:频繁地小规模发布软件,简单设计,集体拥有代码和持续集成。

文中分析了结对编程在软件开发中的重要作用。并在具体项目的实践工作后,将结对编程如何应用到小团队(50人以下)中进行了一定的研究和重构。重构可以让人们理清结对编程的思路,而不是盲目地照搬照套。

1 结对编程的价值分析

由于软件的整体性和系统性非常强,一点微小的瑕疵就可能造成系统的瘫痪,甚至酿成巨大的灾难和

收稿日期:2007-02-14

基金项目:国家自然科学基金项目(70672097),国家自然科学基金重点项目(70631003)

作者简介:钟 扬(1981-),男,安徽合肥人,硕士研究生,研究方向为软件开发方法;刘业政,博士,教授,博士生导师,研究方向为数据挖掘及电子商务。

损失。结对编程通过两个人的智慧,为软件的高可靠性提供了有力的保障,也给团队和个人带来更多的好处^[3]。前期由于结对双方处于磨合期,结对编程会给开发带来一定的成本。但随着时间的推移,结对编程的价值会逐渐显现出来。下面从三个主要方面分析结对编程的价值所在。

1.1 结对编程可以提高工作的效率

软件的内在复杂性决定了软件开发必然是一个复杂的脑力劳动过程。当程序员独自编程时,往往急于完成任务,有了一点零星的想法就开始动工。当代码完成之后,由于前期设计的方案的不合理,导致后期代码维护的噩梦。而结对编程中任何方案都是经过两个人激烈的讨论之后产生的,这有力地保证了设计和编码的质量。

在结对编程中,一个人关注现有逻辑并编写代码,另一个人则从设计的角度思考下一步的工作。有了想法之后,互相讨论,再互换角色。在开发过程中,设计思考和编码实现不停地进行交换,保持了良好的开发节奏。

结对编程还可以使程序员之间互相督促,结对者会更加关注于任务的进展,并减少用于打电话、收发电子邮件和上网冲浪的时间。

1.2 结对编程可以促进知识的传播

软件公司在新员工加入时,一般都会有一段时间的培训,很多软件公司也会建立自己的知识库。但很多新员工在面临实际的项目时仍然会不知所措,培训的效果微乎其微。而结对编程则让新员工有机会与有经验的老员工一起共事,新员工学到的不仅是一些技术和技巧,更多的是他们思考问题、解决问题的方式。和不同的同事一起结对编程,促使你的能力得到快速的提高。

1.3 结对编程可以促进团队的建设

团队是有组织有计划的,可合理有效地利用各种资源,进行最佳的组合。结对编程并不是要一对固定的伙伴,它鼓励在团队中经常交换伙伴。这时,项目不再是一个人的事情,也不是两个人的事情,而是整个团队的事情。通过结对编程,大家可以在最短的时间内完成磨合。结对编程很好地促进了团队的沟通交流,经常一起合作结对编程的伙伴,彼此了解、熟悉,是工作和生活上的好友。在这样的团队里,大家很乐意互相协助,一起分享知识,分享快乐。

1.4 结对编程的实验和证据

在 1999 年,犹他州立大学(University of Utah)做了一项试验^[4]。两组学生,一组独自工作(一共 13 人),一组结对(一共 28 人,即 14 对)。他们需要完成

相同的任务(由助教预先设计并开发了测试案例)。

表 1^[4]是完成相同的三个程序,独自工作和结对工作使测试案例成功通过的百分比。可以看出结对组比个人组的通过率平均要高 18% 左右。

表 1 测试用例通过率(%)

	个人组	结对组
程序 1	73.4	86.4
程序 2	78.1	88.6
程序 3	70.4	87.1

图 1^[4]显示的是完成相同的程序两组所花费的时间比。虽然结对组在刚开始时的优势并不明显,但随后结对组的完成时间开始大幅度下降。

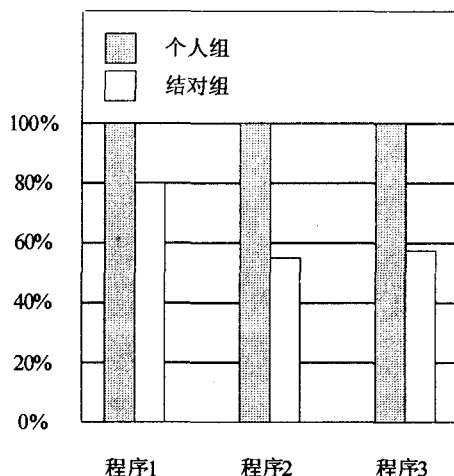


图 1 个人组和结对组完成时间的比较

在具体项目中,结对编程会带来比上面结果更高的价值。因为在实际开发中,如果错误越多,就要花费越多的时间去修复它。在上面的试验中,没有统计修复错误所花费的时间。

1.5 结对编程的误解和驳斥

在看过上述分析之后,多数管理者会认可结对编程带来的价值。然而有些管理者却选择了拒绝,以下是对结对编程的三个主要误解理由。

1) 结对编程造成人力浪费。一人编写代码,一人在旁观察,造成了人力资源的一种浪费。首先,开发人员的能力、经验、工作主动性,对软件开发的成本有非常大的影响。其次,前期的代码在后期发现问题并修正,是需要付出巨大的代价的。所以,两个人一起工作的结对编程使得工作的有效性大大提高,从这一点来说,结对编程反而充分利用了人力,减少了成本。

2) 公司人手不够。人员的流动造成了公司人手短缺,这一直是让很多软件公司头疼的一个问题。特别是老员工的离去,也就意味着公司多年的技术和业务积累的流失。而在结对编程的团队中,通过结对编程

和结对伙伴的交换,知识不再是掌握在一个人的手中,而是整个团队一起共享。通过结对编程使公司的人员、技术和业务处于非常稳定的状态。

3) 结对编程对开发者要求太高。是的,结对编程对开发者是有要求的。它要求开发者乐意和别人沟通、合作,要求开发者能够彼此尊重,愿意和别人分享自己的知识。这正是人们一直倡导的团队合作精神。

2 结对编程的实践和经验

笔者参与为某著名家电企业开发了一套系统,该系统是针对人力资源与考勤等方面的管理系统,以下简称 HR 系统。HR 系统的信息流如图 2 所示。

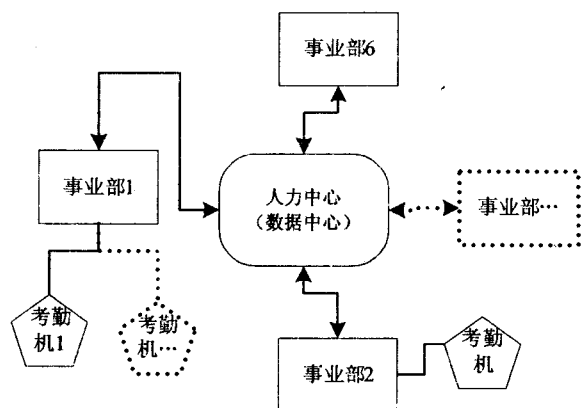


图 2 HR 系统信息流

各事业部的考勤系统是 HR 系统立项前上线的,且分别来自多个不同的供应商,存在着信息融合的问题。HR 系统为了解决这一问题,将数据中心置于人力中心处,并在各事业部安装了 HR 系统客户端。各事业部通过 HR 系统客户端将本部考勤数据导入数据中心,并将人员信息输入数据中心。

由于各事业部的考勤系统和人员管理的异构性,对 HR 系统的开发带来了极大的挑战。在 HR 系统的设计与开发中,笔者和另外一位程序员采取了与其他人不同的策略,即结对编程策略。通过一段时间的结对编程,发现结对的效率不比其他两个独立程序员低。同时,某方对业务流程和技术方案有疑惑时,双方能够及时沟通,找出较优的解决方案。最终发现,通过结对编程,HR 系统出错率明显降低,程序效率更高,业务流程更加合理。

最终通过结对编程的实施,开发小组成员克服种种困难,最终圆满地完成了 HR 系统的开发。目前系统平台运行良好,获得了客户的广泛认可。

通过项目的实践,发现结对编程的实施并不困难,首先需要改变的其实是人们的传统观念。为了获得更好的结对编程的效果,需要更多经验的积累,当然经验

也只会来源于实践。

下面是一些通过结对编程实践获得的经验。

1) 分享代码、经验和喜悦,分享你们在结对中能够分享的一切。结对编程强调合作和分享,代码是双方共同的劳动成果,与此同时,双方获得经验的提升和成功的喜悦。结对者必须要和对方分享这一切。

2) 不要指责对方。在结对中,尽量不要用“你的设计有问题”或“这个错误是你产生的”这样的词语。像“让我们来重新设计一下”和“这段程序终于通过了测试案例”才是应该使用的说法。

3) 公平结对。即使一方比另一方更有经验,也不要让另一方感觉自己不重要。在结对中,每一方都有平等的机会表达自己的观点,双方对代码的控制权是平等的。

4) 合理结对。比如在结对时,双方的有效沟通距离保持在 0.5~0.8m 之间。双方在结对时理智处理冲突,一切要通过协商达成一致。

5) 双方互相关心和体谅。结对编程不仅是为了更好的工作,也是为了更好的生活。结对者们不仅要成为工作上的好搭档,也要成为生活上的好朋友。这样,在编程工作中,双方才能更好的沟通和相处,才能达到结对编程的最大效果。

最后说一下结对编程的搭配问题。有人说,最好的搭配就是两个人能力相当。其实不然,结对编程应该是一种多样的变换组合。在结对编程的团队中,经验丰富的开发者有责任带领新人,传知授道解惑,同时可以享受传道的乐趣。新人,更应主动找有经验的伙伴结对编程,快速学习提高自己。结对编程的核心就是沟通,只要两个人能很好地进行沟通,那么他们就可以很好地搭配。

3 结对编程的两点重构

Martin Fowler 把软件的重构定义为:通过改变软件的内部结构,但不改变软件的可观察的行为的方式,使软件更易理解和维护的一种行为^[5]。

结对编程是从实践中发展出来的,它的理论细节跟当时的实践环境有关。把结对编程应用到具体实践中,需要根据具体的实践环境作相应的重构。同软件的重构类似,结对编程的重构是指通过改变和优化结对编程的实现细节,达到使之适应具体编程环境的目的,当然不变的是结对编程注重合作和沟通的核心和代码要有两人以上对其负责的精神。

3.1 重构一:结对编程工作区改进

根据图 1 的数据,个人生产率可简单计算为:

$$\text{个人组个人生产率: } \frac{1}{100\%} = 1$$

$$\text{结对组个人生产率: } \frac{\frac{1}{60\%}}{2} = 0.83$$

可以看出,结对组的生产率还有很大的改进空间,虽然随着结对编程的实践深入结对者的生产率会进一步提升,但仍然要寻求一定的改进。

任何事物都有两面性,结对编程设定两人共用一台电脑工作、编程,造成资源的使用冲突,这是发现的一个造成个人生产率下降的主要原因。图 3 显示了一个典型的编程工作区的改进和安排示意图。

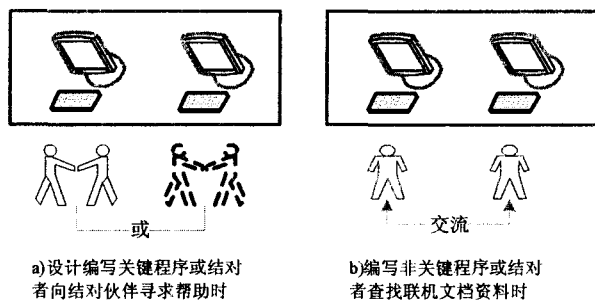


图 3 结对编程工作区改进示意图

3.2 重构二:结对编程扩展性改进

在某些情况下,尤其是在大型项目中,一个模块可能需要两人以上的开发成员来实现,而结对编程只强调了两个开发人员要结为一对。这时引入一个重构:将两人结为一对的强结对,根据情况转变为松结对,即 2~5 人组成一组,结对时根据情况寻找结对伙伴并不断交换。由松结对组成的团队称为松结对团队,如图 4 所示。一个基于结对编程组成的松结对团队必须具备以下两个关键特征:

1) 一个松结对团队中编写的代码对该松结对团队所有成员透明,即代码松结对团队集体所有。

2) 松结对团队是一个利益责任共同体,所有的责任和利益在整个松结对团队中人人平等。

4 结论和下一步工作

结对编程由于注重团队合作的建立,在近十年中逐渐获得了新兴的敏捷组织和团队的欢迎。结对编程不仅提高了软件的可靠性,同时也获得了较高的工作

效率,并且有利于学习型组织的建立。在软件开发中充分实践结对编程,并结合自身的条件作一定的重构,每个企业都可以获得理想的结对编程环境,进而提升自身在现代软件开发业中的竞争力。

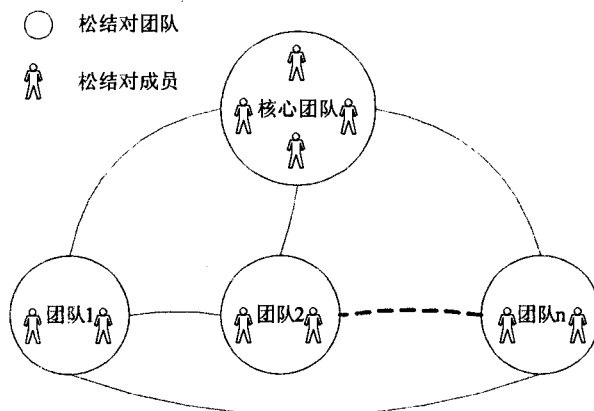


图 4 基于结对编程的松结对团队示意图

未来的主要研究工作是,如何让结对编程适应各种过程不同、团队大小不同、地域分布集中或分散的开发环境,并全面分析结对编程在各种环境中的作用和这些环境因素反过来对结对编程的影响。另一项重要任务是构建一定的工具和平台来支撑各种基于结对编程的软件开发工作。此外,可以在学校里开展学生结对编程的教学和科研活动,收集相关的数据,作进一步的研究工作。

参考文献:

- [1] 徐琛,杨宗源. 轻载软件开发方法[J]. 计算机工程, 2003,29(1):268-271.
- [2] Beck K. Extreme Programming Explained: Embrace Change [M]. 2nd ed. Boston: Addison - Wesley, 2004.
- [3] Cockburn A, Williams L. The Costs and Benefits of Pair Programming [C]//First International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2000). Italy: [s. n.], 2000.
- [4] Williams L, Kessler R, Cunningham W, et al. Strengthening the Case for Pair Programming[J]. IEEE Software, 2000, 17(4):19-25.
- [5] Fowler M. Refactoring: Improving the Design of Existing Code [M]. Boston: Addison - Wesley, 1999.

(上接第 159 页)

- Areas in Cryptography - SAC 2004, LNCS. [s. l.]: Springer - Verlag, 2004:130-143.
- [6] Heuberger C, Karti R, Proding H, et al. The Alternating Greedy Expansion and Applications to Left - to - Right Algorithms in Cryptography[J]. Theoretical Computer Science A,

2005, 341:55-72.

- [7] Muir J A, Stinson D R. New Minimal Weight Representations for Left - to - Right Window Methods [C]// In Proc. Cryptographers Track - RSA Conf (CT - RSA 2005), LNCS. [s. l.]: Springer - Verlag, 2005:366-383.