

RUP 与 XP 软件过程整合设计研究

Research of RUP and XP Software Process Integration Design

王永贵 王 璐 (辽宁工程技术大学 软件学院 辽宁 葫芦岛 125105)

摘要: 为了使中小型软件企业从天才式的自由开发和作坊式的制作模式解脱出来,步入团队化、规范化、标准化的新阶段,本文引入一套规范化的 RUP 与 XP 整合设计的软件过程框架。通过分析 RUP、XP 各自优缺点,对这两种过程模式进行相应的取舍、定制、整合设计后用于软件开发实践中。RUP 和 XP 整合方案完善了 XP 的过程模式,弥补了 RUP 在指导中小型项目开发的不足,有效地解决项目的紧急需求以及时间压力问题,且可以降低开发项目的风险,提高开发系统的效率和质量。

关键词: Rational 统一过程 极限编程 软件开发过程 整合设计 过程模式

1 引言

现代软件工程对软件开发团队提出了更高的要求,比如开发速度要求更快,同时要保证软件的质量绝对可靠、开发经费以及软件的扩展性问题。业界软件开发的过程一直在被传统的软件工程方法主导着。传统的软件工程方法按照瀑布模型或其变型从系统的问题定义、可行性分析、需求定义等一路走来,每一步的开始都要以上一步的完成作为前提。在业界对软件质量越来越关注的情况下,一些新的软件开发方法、开发思想如雨后春笋般的涌现了出来。其中影响比较大的是软件过程框架 RUP 和敏捷方法学。

航、分层树型浏览器、详细的索引、内置的搜索引擎、详细的位置图等。

RUP 是迭代与增量的二维生命周期结构,重复一系列组成系统生命周期的循环,如图 1 所示,横轴通过时间组织,体现开发过程的动态结构。纵轴通过内容组织,体现开发过程的静态结构。生命周期经历了 4 个阶段:先启阶段、精化阶段、构建阶段、产品化阶段。每个阶段又是一次或多次迭代完成的。一次典型的迭代都要经历九个核心工作流程:业务建模、需求、分析设计、实施、测试、部署、配置与变更管理、项目管理、环境。

2 Rational 统一过程技术(Rational Unified Process, 简称RUP)

RUP 是一种软件工程过程(software engineering process)。它提供了如何在开发组织中严格分配任务和职责的方法^[1]。IBM 软件公司开发并维护着这个产品,并与其它 Rational 软件开发工具集成。RUP 有自己的过程框架(process framework),各个组织可以根据自身的实际情况,以及项目规模对 RUP 进行裁剪和修改,以制定出合乎需要的软件工程过程。RUP 又是文档化的软件工程产品,所有 RUP 的实施细节方法引导可以用任意的 WEB 浏览器查看,并且可以很容易查找到信息,原因在于它有广泛的超链接、图形导

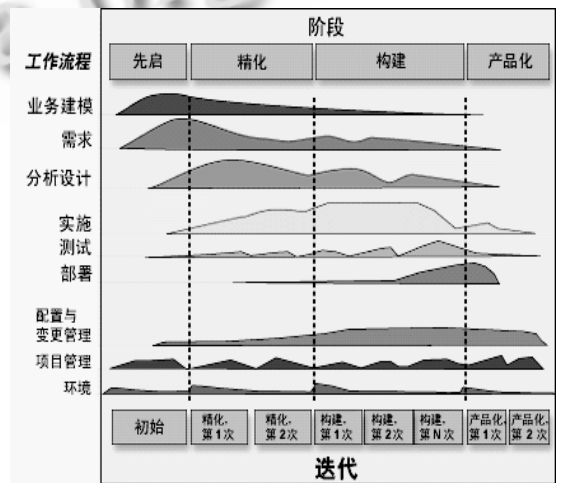


图 1 RUP 的二维结构过程

① 基金项目:辽宁省教育厅基金(05L169)

收稿时间:2008-12-16

3 极限编程技术(eXtreme Programming, 简称XP)

极限编程是由 KentBeck 在 1996 年提出的, 名称中的“Extreme”(极限)是指, 对比传统的项目开发方式, XP 强调把它列出的每个方法和思想做到极限、做到最好^[1]; 其它 XP 所不提倡的, 则一概忽略(如开发前期的整体设计等)。一个严格实施 XP 的项目, 其开发过程应该是平稳的、高效的和快速的, 能够做到一周 40 小时工作制而不拖延项目进度。XP 是一种近螺旋式的开发方法, 它将复杂的开发过程分解为一个个相对比较简单的小周期; 通过积极的交流、反馈以及其它一系列的方法, 开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等, 并根据实际情况及时地调整开发过程。

4 RUP与XP整合设计的研究

RUP 是一个以用例驱动、体系架构为核心、迭代递增式的、可裁剪的软件过程, 已经被广泛的应用于各类软件企业和软件开发的过程之中。XP 是一种软件过程思想, 旨在解决软件生命周期中存在的诸多问题, 例如解决如何应对需求变化、时间压力等问题。因此, 从软件实践的角度上来讲, RUP 与 XP 并不冲突, 甚至可以将 RUP 与 XP 进行整合设计应用于软件开发中以提高开发效率。

4.1 RUP 应用中存在的问题

RUP 提供的是理想开发环境下软件过程的一种完整且完美的模式, 然而商业环境中项目开发环境往往表现出人员、工具、资金等有限的资源与有限时间的约束问题^[2]。RUP 是过程组件、方法以及技术的框架, 可以对其进行适当的剪裁和扩充, 将其应用于任何特定的软件项目, 由用户自己限定 RUP 的使用范围。尽管 RUP 宣称是可配置的过程, 但是 RUP 并未给出具体的裁剪、扩充等配置实施的方法准则, 而这一点恰恰是解决在有限资源与时间约束下项目开发成败的关键。

RUP 从本质来说还是一个强调设计和规范的软件方法, 从这个角度来讲, 与传统的瀑布模型没有太大差别, 它的灵活性较之敏捷方法还是相对较弱的。RUP 中各个开发阶段定义的大量的制品并不是所有的都是交付用户的, 可交付的只是所有制品的子集。其中绝大部分的制品都是开发过程本身服务的。生产不必要

的制品只会造成工作量上的浪费。在一些中、小型软件项目, 特别是不可预测的软件项目开发中, 面临着各种紧急需求、面临着时间压力, 沿用 RUP 是很难应付自如。

4.2 XP 应用中存在的问题

XP 作为过程模式并不完整, 缺少制度化, 它不含有被 CMM 认为是良好的工程和管理实践制度化的关键设施和管理要件^[3]。XP 在人员、方法、产品等方面的论述远不及 RUP 全面详细, 例如 XP 未给出人员的职责分配方案、各方法赖以实现的工具环境、产品的类型划分及建模方法等。XP 核心原则是以人为本, 但是却淡化了过程管理的重要性。XP 需要领导人员拥有丰富的开发经验, 极端方法掌握不好, 很容易陷入混乱状况, 因此解决这个问题就需要过程来进行约束与引导。

XP 对于客户需求变更较为频繁的项目比较适合, 对于大型的、长期的项目, 使用 XP 的开发方式, 会带来一些弊端。由于开发中所产生的文档较少, 直接面对代码, 在开发结束后, 或者开发团队解散以后, 整个系统维护起来相当困难。

软件变更是需要一定的时间的, 在极限编程中, 欢迎客户对其需求进行变更, 可以体现对客户的人性化, 更好的把握客户需求, 然而变更后的开发会需要一些时间, 有可能会使项目的进度后延。

4.3 RUP 与 XP 的整合可行性分析

(1) 在“应用开发标准”实践方面, 软件开发不能局限于 RUP 所建议的指导原则和标准。为提高敏捷性, 可依据 XP 的“适应项目具体情况”的原则做适当的改动^[4]。

(2) 在设计方面, RUP 考虑产品的适应性、可扩展性与可重用性等高性能需求, 提倡以架构为中心的设计方法, 要求架构必须留有实现现在和未来需求的所有用例空间。XP 强调简单化设计, 只考虑当前定义的功能而不考虑以后需求的变化。这两种方法看似对立, 实则是对产品不同质量要求的不同应付策略。

(3) 在测试方面, RUP 和 XP 都强调质量问题。XP 强调“测试先行”。在编码开始之前先将测试写好, 而后再进行编码, 直至所有的测试都得以通过。RUP 对测试也非常重视, 只是 RUP 和 XP 两者对于测试在整个项目开发周期首先出现的位置设置不同而已。

(4) 在开发方式方面, 都强调迭代开发, XP 强调

的短迭代,小交付可以改进 RUP 只有部署时用户才能看到真正的系统问题。尽早地为客户发布有意义、有价值的系统可以为开发人员提供具体的反馈,告诉开发人员哪些需求符合客户要求,哪些不符合^[5]。

4.4 RUP 与 XP 整合设计方案

RUP 与 XP 的整合设计过程遵循 RUP 分阶段逐步演进的主导思想,仍然沿用 RUP 的四个阶段,初始阶段以需求分析为主,精化阶段以分析设计为主,构建阶段以实现为主,产品化阶段以产品移交为主。根据待开发项目的具体情况,对 RUP 进行适当的剪裁和改进。将 XP 思想中的现场客户、计划游戏、结对编程、测试驱动、短迭代小交付等引入到 RUP 的四个阶段开发之中,这样就可以得到新的开发过程框架。

(1) 初始阶段

定义项目计划、风险分析、项目前景(哪些在项目的范围之内,哪些在项目的范围之外)、确定参与者、涉众及收益、用例设计(勾画出最核心用例的应用场景,这些用例计划在精化阶段实现)、资源管理(计划中所需要的人员、工具以及其他类型的资源)、评价标准(对迭代的评估,确定是否我们可以进入下一个迭代计划)。

(2) 细化阶段

确定用户需要、产品特性并确认架构风险;细化需求,编写需求规格说明书,制定迭代计划,需求基线,完成重要用例的设计与实现,由此确定系统架构及第三方组件。已制定迭代计划,同时编写对应用例的测试用例。

(3) 构建阶段

按计划迭代开发,在每个迭代里采用小瀑布的方式,应用 XP 的部分最佳实践,每一个迭代为一个里程碑,提交给客户确认,由此得到需求变更,分析后调整迭代计划^[6]。将对用例的识别、描述和实现扩展到全部的用例。结对编程、小交付、现场客户、持续集成、代码重构、编码标准完成迭代过程。保持架构的完整性,在必要时要对其进行修改。监控前两个阶

段发现的关键的和重要的风险。

(4) 产品移交阶段

在产品化阶段多次向用户交付代码和支持材料,确保项目在产品化阶段顺利地顺利完成。提交客户测试,作小的修改,编写产品说明,用户培训,上线运行。

5 小结

整合设计的过程框架分别吸收借鉴了 RUP 与 XP 的优秀方法与思想。通过对 RUP 和 XP 的整合分析与设计可以看出,RUP 需要 XP 为其提高敏捷性,提高其对变化的快速反应能力,以适应市场的不断变化和客户需求的不变化;而 XP 可以利用 RUP 深刻的理论基础作为其发展的基石。总结出 RUP 与 XP 整合的过程模式是一个以用户为中心、架构为中心、用例驱动、测试驱动、风险驱动、即时沟通与反馈、易变更的有序软件开发过程。RUP 和 XP 的结合能够适应变化,使得开发团队更有效率的工作。

参考文献

- 1 Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process.北京:China Machine Press,2002.
- 2 金敏,周翔编.高级软件开发过程—Rational 统一过程、敏捷过程与微软过程.北京:清华大学出版社,2005.
- 3 张小红,韩小汀.基于 RUP 和敏捷软件过程的小型项目 IT 项目开发过程.郑州大学学报,2007,40(2):440-441
- 4 马敬远著.基于 Agile 的统一过程的研究与设计[硕士学位论文].北京:北京交通大学,2008.
- 5 Monnox A. Rapid J2EE Development.北京:China Machine Press,2005.
- 6 Pollice G, Augustine L, Lowe C, Madhur J. Software Development for Small Teams A RUP- Centric Approach,2004.