本栏目责任编辑:谢媛媛

基于 RUP 的软件开发过程研究

宋 昕

(浙江科技学院 信息与电子工程学院,浙江 杭州 310023)

摘要:随着对软件需求的不断增大,要求的不断提高,软件开发机构迫切地需要一种能够更有效地开发更高质量软件的方法。统一软件过程 RUP 是一种用例驱动,以架构为中心,采用迭代增量方式开发的软件工程过程。RUP 作为一种通用的软件过程框架,适用于大多数的软件项目,而信息系统的开发,也需要引入一种适当的开发过程作为指导,以提高质量、开发效率和复用性等。

关键词:RUP;软件过程;用例驱动;核心工作流

中图分类号:TP311　　**文献标识码**:A　　**文章编号**:1009-3044(2008)21-30459-03

Research of RUP-based Software Development Process

SONG Xin

(School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China)

Abstract: With the continuous growing demand and requirement for software, software developers need a more effective development method of higher-quality software development. Rational Unified Process is a software engineering Process which is use case driven, architecture-centric and using iterative and incremental developing method. RUP is a general framework of software Process. It fit for most software Projects, developing Information system, also need a development Process to advance the quality, efficiency and reusability.

key words: RUP; Software Process; Use case driven; Core Workflow

1 引言

行之有效的软件过程可以提高开发软件组织的生产效率、提高软件质量、降低成本并减少风险。传统的软件过程采用结构化技术,其缺陷对于开发中小规模、结构相对简单的软件而言似乎不明显,但对于规模庞大、结构复杂、软件需求模糊的软件开发项目则容易产生开发人员缺乏交流、缺少共享信息的问题。为弥补这方面的缺陷,适应软件开发的需要,涌现了许多行之有效的软件开发过程,其中应用比较广泛且具有代表性的软件过程主要有 RUP、XP、OOSP、DSDM、Catalysis 和 OPEN Process 等几种[1-2],本文将对具有较高知名度的 RUP 统一过程及其应用进行研究。

2 RUP 的开发要点

RUP(Rational Unified Process)是由 Rational 公司开发的一种软件工程过程,主要由 Ivar Jacobson 的 The Objectory Approach 和 The Rational Approach 发展而来,是文档化的软件工程产品。所有 RUP 的实施细节及方法引导均以 Web 文档的方式集成,由 Rational 公司开发、维护并销售,为各种软件开发组织提供了一种有效的分配、管理任务和职责的规范方法,保障开发组织能够在预定的进度和范围内开发出满足最终用户需要的高质量软件产品[2]。RUP 又是一个通用框架,各个组织可根据自身情况及项目规模等对 RUP 进行裁剪和改进,以制定出合乎需要的软件工程过程。RUP 的总体结构如图 1 所示。

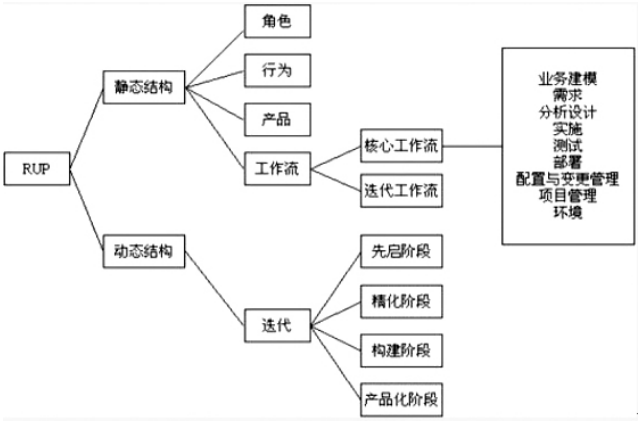

图 1　RUP 的总体结构

RUP 是一种具有明确定义和结构的软件工程过程。它采用用例驱动、以架构为中心、迭代增量的软件开发方法[3]。它明确规定了人员的职责、如何完成各项工作以及何时完成各项工作,并提供了软件开发生命周期的结构。自诞生时起,就引起了全球软件行业的关注,经过大量商业实践表明,RUP 是解决软件开发过程中根本问题的方法。

2.1 RUP 是用例驱动的

用例几乎普遍用来捕获系统的需求,但在 RUP 中,用例不只是捕获需求的工具,它们还能够驱动整个开发过程,是贯穿整个开发过程的线索。通过用例的驱动,我们可以比较清楚的看一个软件系统是如何实现其功能的。

用例是用户与系统的交互的动作集合,能够向用户提供有价值的结果。它获取的是功能需求,所有的用例合在一起,构成用例模型,它描述了系统的全部功能,代替了传统的系统功能说明。然而,用例不仅是一种确定系统需求的工具,它还能驱动系统分析、

收稿日期:2008-04-25

作者简介:宋昕(1982-),女,浙江金华人,助教,主要研究方向:数据库、软件工程。

软件设计开发

本栏目责任编辑:谢媛媛

设计、实现、测试的进行,即用例驱动整个的软件开发过程。基于用例模型,开发人员创建一系列的实现这些用例的分析、设计和实现模型,并审查每一个后续建立的模型与用例模型是否一致,测试人员测试系统以确定实现模型的构件正确实现了用例。因此,用例不仅启动了开发过程,而且使整个开发过程浑然一体。RUP 的用例驱动模型如图 2 所示。

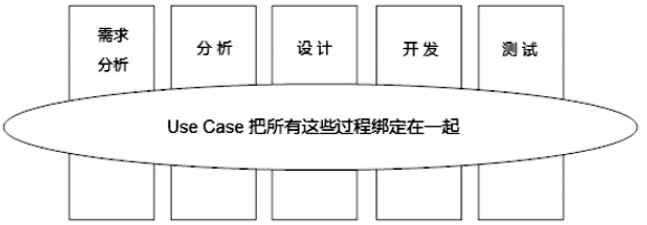

图 2　用例驱动模型

2.2 RUP 以构架为中心

软件系统是个单一的实体,从不同的视角展示它有助于更好地理解系统的设计。系统的不同视角的展示就是视图,所有的视图合在一起展示了架构。我们可以通过架构来方便地理解系统,可以通过架构来组织并行高效的开发,可以通过架构来更好地重用,可以通过架构来方便地进化系统。以架构为中心意味着,开发工作在早期阶段,就侧重于建立能够指导系统构造的架构模式,以保证不仅当前版本、而且产品的整个生命周期都有一个顺利的发展。

软件基本架构这个概念体现了系统中最重要的静态和动态特征。它刻画了系统的整体设计,去掉了细节部分,突出了系统的重要特征。架构时架构设计师在构件描述中详细说明的内容,架构可以控制系统的开发。软件架构侧重于系统的重要结构元素,如子系统、类、构件和节点,以及这些元素通过接口实现的协作。4+1 场景模型呈现了 RUP 架构,软件架构设计师和开发人员发现从不同视角展示该系统有助于更好地理解其设计。这里 4+1 指的是:逻辑视图、实现视图、过程视图、部署视图以及用例视图[4]。如图 3 所示:

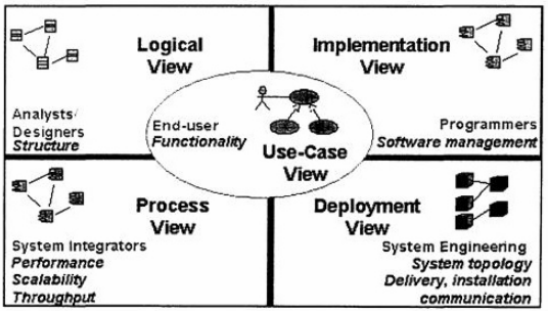

图 3　RUP 构架描述

2.3 RUP 是迭代增量开发

开发一个商业软件产品是一项艰巨的工作,可能持续几个月、一年甚至更长时间。因此,将一项工作分解成若干更小的部分或若干小项目是切合实际的。每个小项目是指能产生一个增量的一次迭代。迭代是指工作流中的步骤,而增量指的是产品的增加部分。在每次的迭代过程中开发人员标识并描述有关的用例,以选定的架构为向导来创建设计,用构件来实现设计,并验证这些构件是否满足用例。如果一次迭代达到了目标,开发工作便可以进入下一次迭代。如果一次迭代未能达到预期的目标,开发人员必须重新审查前面的方案,并试用一种新的方法。

图 4　RUP 的迭代开发过程

一个增量是一次迭代的内部版本与下一次迭代的内部版本直接的差别。迭代过程是以逐渐递增的方式构造出最终模型,每次迭代在经历需求、分析、设计、实现和测试时,会对每种模型增加一些内容。其中有些模型(如需求模型)在初期会得到更多的关注;而其他模型(如实现模型)则在构造期间成为关注的重点[5]。迭代和增量的开发意味着,有计划地完成每一个袖珍的项目,直至产品的最终完成。图 4 描述了软件的 RUP 迭代开发过程。

2.4 RUP 是可裁剪的

对于 RUP 的应用范围,一直以来都存在着一些争论和错误的观点。人们以为‘软件工程过程’一定要遵循大量的规定、指导和格式,并且所有标准充满了管理性的规定,通常只对大型团队开发大型项目时有指导意义,只有当系统足够复杂时才需要使用。事实上,RUP 是一个通用的软件过程,它可以通过裁剪,适用与大多数的软件项目。对于 RUP 是否可以变的敏捷,应用于中小规模项目中,许多专家也已经提出了他们的看法。著名学者 Michael Hirsch 在其《Making RUP Agile》一文中写道‘Can this be agile? Yes it can, if you know what to choose.’他通过在许多小项目上成功应用了 RUP 后,指出 RUP 完全可以适应小项目的需要,在小项目中成功应用 RUP 的一个关键是仔细选择合适的制品子集并保持这些制品非常简明、剔除不需要的形式主义制品。因而,在应用 RUP 时很关键的一点,就是要根据项目自身的特点,结合实际情况,对 RUP 过程进行裁剪,‘量体裁衣’,只有找到适合自己的方法,才能发挥其良好的效果。

3 RUP 开发过程

为了能够方便地管理软件开发过程,监控软件开发状态,RUP 把软件生命周期划分为若干次迭代,每次迭代生成一个产品的新版本并依次由四个连续的阶段组成,每次迭代都应完成确定的任务,正是这些迭代过程不断产生系统新的增量,使产品不断成熟,从低版本软件不断过渡到高版本软件。

软件生命周期被分解为周期,每一个周期工作在产品新的一代上,RUP 将周期又划分为初始(Inception)、细化(Elaboration)、构

造(Construction)和移交(Transition)四个连续的阶段。每个阶段都终结于一个良好定义的里程碑。其实,每个阶段就是两个主要里程碑之间的时间跨度[6]。在每个阶段结束时要依据里程碑目标进行工作评估,以确定是否实现了该阶段的目标以及是否可以进入下一个阶段。如图5所示为各阶段和里程碑的关系。

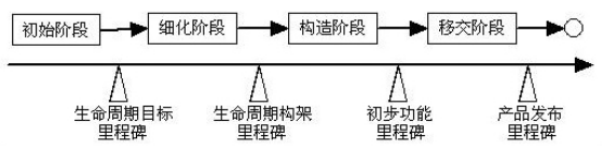

图5 阶段和里程碑

在初始阶段,项目组集中精力理解最初的需求,确定范围并组织项目。要理解最初的需求,可能要进行业务建模及其他基本的建模活动。细化阶段是建立系统架构的基线,以便为构造阶段的主要设计和实施工作提供一个稳定的基础的阶段。构造阶段是将主要精力集中在现象设计、实现以及测试来充实一个完整的系统的阶段。移交阶段是系统正式投入运行前的阶段,要达到的主要目标是确保软件完全满足用户需求。

工作流,是产生具有可观察结果的活动序列,每个工作流产生一些有价值的工件。RUP共有9个工作流,其中分为6个核心过程工作流:业务建模、需求、分析和设计、实现、测试、部署工作流;3个核心支持工作流:项目管理、配置和变更控制、环境工作流。这9个工作流并不是顺序执行的,而是在项目中轮流被使用,在每一次迭代中以不同的重点和强度重复。核心工作流的迭代开发循环模型如图6所示。

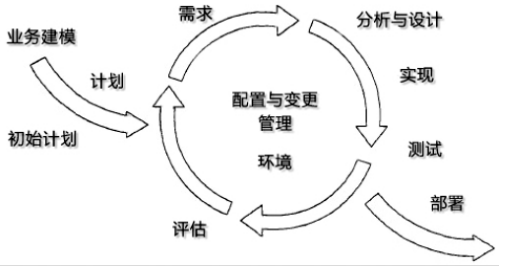

图6 核心工作流的迭代开发循环

4 结论

RUP是新一代软件工程开发方法,近年来在软件开发领域中得到了广泛关注。RUP采用面向对象技术,在迭代的开发过程、需求管理、分析与设计、可视化软件建模、验证软件质量及控制软件变更等方面为软件开发人员提供了相应的准则、模板和工具。在软件开发过程中应用RUP,可以规范管理,降低软件复杂性,减少软件开发风险,提高软件质量。

参考文献:

[1] Ian Sommerville.Software Engineering[M].北京:机械工业出版社,2006.
[2] Scott W. Amble.The Unified Process Elabration Phase Best Practices in Implementing the UP[M].北京:机械工业出版社,2005.
[3] 余八一.现代软件工程过程方法探析[J].科技资讯,2007,1(02):45.
[4] 张友生.基于RUP的软件过程及应用[J].计算机工程与应用,2003(30):104-107.
[5] 陆永忠,饶璟祥.小型软件项目RUP裁剪模型的研究[J].计算机工程与设计,2007,28(03):3027-3030.
[6] 吕西红,陈志刚.统一软件开发过程RUP中的关键技术研究[J].信息技术,2006(1):27-29.

(上接第454页)

作符重载等,使Java比c++更容易学习,其程序的可读性也更强。同时Java是一种更纯粹的面向对象程序设计语言。面向对象编程具有多方面的吸引力。对管理人员,它实现了更快和更廉价的开发与维护过程。对分析与设计人员,建模处理变得更加简单,能生成清晰、易于维护的设计方案。对程序员,对象模型显得如此高雅和浅显。此外,面向对象工具以及类库的巨大威力使编程成为一项更使人愉悦的任务。

总之,C++语言功能强大,操作系统和Office应用程序都是由C++程序编写的,可满足对功能的需求,Java语言是一门很优秀的语言,具有面向对象、与平台无关、安全、稳定和多线程等优良特性,是目前软件设计中极为健壮的编程语言。C++与Java已成为网络时代最重要的语言之一。

参考文献:

[1] 池静,邢秀娥.基于Java和C++语言的安全性讨论[J].河北建筑科技学院攀报,2005(4):86-88.
[2] 刘哲,史诗.Java和C++语言安全机制及网络安全规范的讨论[J].电化教育研究,2002(7):65-67.
[3] David L H,Mrmca S L.A practical flow-sensitive and context-sensitive C and C++ memory Leal(detector[J].ACM press.2003,38(5):168-181.
[4] Bryan Bates.C as a first language:comparisons with C++[J].The Consortium for Computing in small College,2004,19(3):89-95.