

# 软件过程模型应用策略

## Application Strategy of Software Process Model

王 军 ( 河南省信息中心 河南郑州 450003 )

**摘 要:** 该文提出了三种软件过程的应用策略。在比较分析几种不同软件过程模型的基础上,结合不同的项目实践,表明三种策略的应用环境,并且举例说明了方法在软件项目中的应用。实践证明,这些软件过程设计策略在项目实施的过程中起到了很好的指导作用。

**关键词:** 软件过程模型 模型比较 过程应用 RUP XP

本文第 1、2 节对软件过程进行了介绍,并对常用的几种软件过程进行比较分析,第 3 节在比较的基础上结合工程实践提出三种应用策略,最后对策略进行了总结。

随着软件开发技术和方法的发展,越来越多的大型软件系统被提出来,工程化的管理思想正在软件开发过程中实践,一系列软件过程模型也被提出来。针对不同的模型有其一定的适应环境和范围,本文从不同方面对目前常用的模型进行分析,在此基础上提出软件过程涉及的应用策略。

### 1 软件过程介绍

按 ISO/IEC 12207 标准,软件过程是指软件生命周期中的若干活动( activity )的集合<sup>[1]</sup>。活动,有时又称为工作流程( Workflow ),又可细分为子活动或者任务。使用软件过程目的是为了在一定的时间和经费预算内开发出高质量的产品。一个好的软件过程可以提高软件开发组织的生产效率、提高软件质量、降低成本并减少风险。

近年来,对软件过程的研究呈现出两个特点:

(1) 研究层次的变化,从高层抽象全面过渡到对过程细节的研究。早期,主要针对抽象的软件生命期模

型进行软件过程研究,过程活动的细节主要取决于个人的素质和经验。目前对软件过程的研究已全面过渡到过程细节的研究。

(2) 研究内容的扩展,研究范围已发展到对“过程建模”、“过程实施”、“过程度量”、“过程改进”等诸多相关内容的研究<sup>[2]</sup>。

本文对常用的软件过程模型进行比较分析,并对过程模型的应用策略进行了研究。

### 2 软件过程模型比较分析

随着软件需求不断增加,软件项目规模也越来越大,人们逐渐意识到要建立一种抽象模型来对软件开发过程进行描述。从 1970 年,由 Royce 提出了第一个软件过程模型——瀑布模型到上世纪 80 年代以来,又陆续出现了快速原型模型、渐增式模型、螺旋模型、可重用模型、面向对象的模型、形式化模型、领域模型、RUP 模型等多种新模型。其中每个过程模型都代表了一种将本质上无序的活动有序化的企图,每个模型都具有能够帮助实际软件项目的控制及协调的特性。表 1 分别从模型的优缺点、适应范围和包含的活动方面就一些常用的软件过程模型进行了比较分析。

表 1 软件过程模型比较分析

软件过程模型	优点	缺点	适用范围	包含的活动
瀑布模型	线性顺序模型 结构简单、清晰	不带反馈环的,需求难以十分完善	应用需求明确的情况	软件需求分析、设计、代码生成测试、维护
增量模型	每一个增量均发布一个可操作的产品	增量包之间存在相交的情况不好处理	不能在设定的期限内完成产品时,先推出核心产品	同瀑布模型
螺旋模型	有利于已有软件的重用有助于把软件质量作为软件开发的重要目标	风险评估比较困难	主要适用开发的大规模软件项目	制定计划、风险分析、实施工程、用户评估
原型模型	能够快速实现一个可实际运行的系统初步模型	不带反馈环的,基本上是顺序的第一个系统(原型)常常抛弃	有结构的系统或者需求不明确的系统	快速原型 + 瀑布模型
迭代模型	能够早得到用户的反馈,不断的测试和整合关注项目短期里程碑	迭代周期难以确定难以规划出每次迭代的内容和达到的目标	系统需求不稳定的情况	同瀑布模型
RAD 模型	通过大量使用可复用构件,采用基于构件的建造方法赢得了快速开发	时间约束需要“一个可伸缩的范围”	对模块化要求比较高的信息系统系统	业务建模、数据建模、过程建模 应用生成、测试及反复
XP	强调适应性而非预测性、强调以人为中心,而不以流程为中心,以及对变化的适应和对人性的关注	对于大型的系统应用困难	面向客户的开发模型,重点强调用户的满意程度	用户故事、体系结构、发布计划、交互、接受测试和小型发布
RUP	针对所有关键的开发活动为每个开发成员提供了必要的准则、模板和工具指导,并确保全体成员共享相同的知识基础	没有涵盖软件过程的全部内容,没有支持多项目的开发结构,这在一定程度上降低了在开发组织内大范围实现重用的可能性	多用于面向对象且基于网络的程序开发	通过时间组织,来描述它的周期、阶段、迭代和里程碑;以内容来组织来描述活动、产物、工作者和工作流
智能模型	利用专家系统来帮助软件开发人员的工作	需要系统开发人员与领域专家的密切合作	适应于特定领域软件和专家决策系统的开发	瀑布模型和专家系统结合
变换模型	解决了代码结构经多次修改而变坏的问题,减少了许多中间步骤	需要严格的数学理论和一整套开发环境的支持	基于形式化规格说明语言及程序变换的软件开发	需求定义、规格说明、软件原型 原型评审、程序转换
并发开发模型	在软件开发全过程活动的并行化,打破了传统软件开发的各阶段分割封闭的观念	强调开发人员团队协作,注重分析和设计等前段开发工作,从而避免了不必要的返工	对于客户/服务器结构更加有效,可以随时查阅到开发的状态	客户要求、管理决策,评审结果

### 3 软件过程模型的应用策略

为了解决产业环境中的实际问题,获得满足工程目标的软件,软件项目负责人及团队成员必须综合出一个开发策略,该策略覆盖过程、方法、和工具。对于一个特定的项目,可以通过剪裁过程定义所需的活动和任务,并可使活动并发执行。与软件有关的单位,根据需求和目标,可采用不同的过程、活动和任务。

#### 3.1 一元应用策略

该应用策略主要指在一项软件工程中活动中以一种主要软件过程模型为主进行项目的过程设计。该策略前提是软件工程项目的特点符合某一种过程模型的应用范围,并且该模型的应用能够满足工程目标的实现。下面以 RUP 模型在具体软件工程中的应用来说明。

3.1.1 RUP (Rational Unified Process, 统一软件过程) 是一个面向对象且基于网络的程序开发方法论。RUP 和类似的产品——例如面向对象的软件过程(OOSP),以及 OPEN Process 都是理解性的软件工程工具——把开发中面向过程的方面(例如定义的阶段、技术和实践)和其他开发的组件(例如文档、模型、手册以及代码等等)整合在一个统一的框架内。RUP 软件开发生命周期是一个二维的软件开发模型。横轴通过时间组织,是过程展开的生命周期特征,体现开发过程的动态结构,用来描述它的术语主要包括周期(Cycle)、阶段(Phase)、迭代(Iteration)和里程碑(Milestone);纵轴以内容来组织为自然的逻辑活动,体现开发过程的静态结构,用来描述它的术语主要包括活动(Activity)、产物(Artifact)、工作者(Worker)和工作流(Workflow)<sup>[5]</sup>。

3.1.2 RUP 中的软件生命周期在时间上被分解为四个顺序的阶段,分别是:初始阶段(Inception)、细化阶段(Elaboration)、构造阶段(Construction)和交付阶段(Transition)。每个阶段结束于一个主要的里程碑(Major Milestones);每个阶段本质上是两个里程碑之间的时间跨度<sup>[3]</sup>。RUP 中有 9 个核心工作流,分为 6 个核心过程工作流(Core Process Workflows)和 3 个核心支持工作流(Core Supporting Workflows)。6 个核心过程工作流分别是:商业建模、需求、分析和设计、实现、测试、部署;3 个核心支持工作流分别是:配置和变更管理、项目管理、环境。尽管 6 个核心过程工作流可能使人想起传统瀑布模型中的几个阶段,但应注意迭

代过程中的阶段是完全不同的,这些工作流在整个生命周期中一次又一次被访问。9 个核心工作流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。

3.1.3 RUP 是一个完整的开发过程方法,其核心思想是为例驱动,以构架为中心的迭代和增量开发[4,6],由于信息系统开发有自身的特点,比如人员计算机知识欠缺,业务流程不够完善,信息化建设风险大等。因此可以根据开发组织和项目的特点在 RUP 的模型基础上进行构建企业的软件过程模型。图 1 为具体信息系统在构造阶段的视图(限于篇幅限制,此处不再展开)。

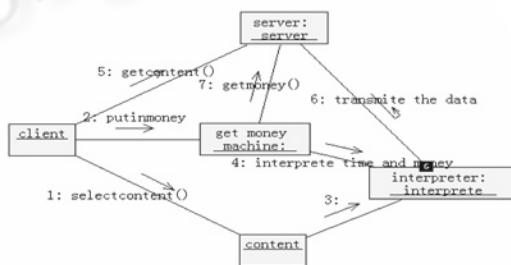


图 1 公共管理信息系统子系统协作图

#### 3.2 二元或多元组合应用策略

该策略指在具体的软件工程项目运用二种或者多种软件过程模型来构建软件过程。这种策略可以具体变换为两种或者多种软件过程模型的交叉构建;一种过程模型包含另外一种或者多种过程模型构建;两种或多种模型并行构建等。

3.2.1 用 XP 模型构造软件原型,用原型模型结合部分迭代模型来构件系统的过程。既能体现 XP 模型的速度也能够很好地控制项目的风险,同时 XP 模型还为原型模型中圆形的实现提供具体的操作方法。

3.2.2 用 RUP + XP 的模式来构建系统的过程。可以开始阶段使用 RUP 做业务建模、需求分析等,随后在开发阶段过程中用适当的 XP 方法帮助团队快速地前进,减少项目面对的实际风险。这样在撰写文档方面既不是 RUP 那样文档繁多,也不是 XP 的“代码即文档”,而是保留必要的文档:系统功能说明书、类划分、模块及接口设计,源代码文件说明及精确的代码注释,并随变化及时更新文档。这样既提高了开发人员的工作积极性同时也保证系统的可维护性。

以上两种方法只是笔者在参与的一些项目中采用的软件过程方法,这两个粒子的目的是说明二元或多元组合应用策略应用的灵活性,也就是根据具体的项目特点和多模型特点综合分析获取系统的过程设计。

### 3.3 创新应用策略

随着技术的发展和企业业务的复杂化程度逐步提高,更复杂的系统需求逐步显现出来。各个系统特点和模型特点的关联程度越来越低,并且不同系统的特征差异也越来越大,现有的一些过程模型可能不能够满足复杂系统的需求,创新的模型应用策略也就随之而出现。

该策略主要针对复杂的系统,结合系统的需求、时间、成本、质量等各方面的约束因素而设计的专用的软件过程。

## 4 总结

软件项目种类繁多,没有任何一种方法能够适用于所有的项目。本文通过对几种常用的软件过程模型的分析,提供了三种针对不同项目的应用策略,但并不是一把万能钥匙,准确地说应该是一种启发式的

思路、开放式的方法,具体在项目实施时也要结合行业特色、项目背景和技术平台等客观因素,因地制宜地加以变化和创造,再综合其它方法的优点以求获得最适合项目的方法。

### 参考文献

- 1 朱三元,钱乐秋,宿为民. 软件工程技术概论. 科学出版社, 2002
- 2 李松领,金蓓弘. 软件过程的比较框架研究. 计算机科学, 2004, 31(5).
- 3 克鲁奇特,麻志毅,等. RUP 导论. 3 版. 北京:机械工业出版社, 2004.
- 4 Ivar Jacobson, Grady Booch, and Jim Rumbaugh, Unified Software Development Process, Addison - Wesley, 1999.
- 5 李松领,金蓓弘. 软件过程的比较框架研究. 计算机科学, 2004, 31(5).
- 6 惠建华,赵春江,吴华瑞,王万森. 软件工程过程 RUP 研究. 微机发展, 2004, 14(4).