**关于软件过程管理基于agile的研究**

**论文作者：亢震**

**工程领域：软件工程**

**指导老师：邢鑫**

**摘要**

当前的众多信息系统开发都存在着业务流程复杂、需求不明、充满不确定因素和风险性高等特点,传统软件开发过程于此没有很好的解决措施,由此导致大量企业信息化项目的失败,阻碍了企业信息化进程。针对该问题, 敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运行使用的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。 通过良好的实践过程与过程模式来缓解信息系统开发的危机,提高软件产业的生产效率。

[敏捷建模](http://baike.baidu.com/view/1412845.htm)（Agile Modeling，AM）的价值观包括了EP（Extreme Programming：极限编程）的四个价值观：[沟通](http://baike.baidu.com/view/54445.htm" \t "_blank)、[简单](http://baike.baidu.com/view/66543.htm)、[反馈](http://baike.baidu.com/view/80120.htm)、[勇气](http://baike.baidu.com/view/53936.htm)，此外，还扩展了第五个价值观：谦逊。

敏捷开发是针对传统的[瀑布开发模式](http://baike.baidu.com/view/4169795.htm" \t "_blank)的弊端而产生的一种新的开发模式，目标是提高开发效率和响应能力。除了原则和实践，模式也是很重要的，多研究模式及其应用可以使你更深层次的理解敏捷开发。

本论文在深入研究软件过程的基础之上,决定采用敏捷过程的方式,建立敏捷化的过程来指导现代企业信息系统的开发。论文先分析软件过程的概念以及改进方法,对统一过程进行相应裁减,然后对裁减的过程加以敏捷化,建立敏捷过程框架。

**关键词：过程管理；敏捷开发**

**1软件过程概述**

软件系统的开发与建设是一个系统工程,将工程化的思想应用于软件开发过程,用工程管理的思想来定义软件过程,将有助于我们发现软件过程、软件工程中的薄弱环节,有助于我们通过过程改进的办法改进软件过程,提高软件开发的效率,丰富软件过程的思想,同时对工程管理学是一种扩展。本章将重点讲述软件过程的概念,模式,软件过程描述,以及软件过程改进的方法等等。为下面研究ARUP的提出打下理论基础

。

**1.1软件过程的概念**

软件过程包括了从软件项目需求定义开始到软件使用、升级直到被废弃的整个过程,跨越整个软件的生存期,是系统开发、运行和维护等全部活动及相关项的总和。过程就是活动的集合,活动是任务的集合,任务要起着把输入进行加工然后输出的作用。活动的执行可以是顺序的、迭代的、并行的、嵌套的或者是有条件地引发的。

**1.2软件过程模式的概念**

软件过程模式是从成功或失败的软件开发实践中总结出来的,是软件过程中生命周期、人员、活动、产品四大类要素相互关联的有机整体。

从软件过程模式的定义可以看出,软件过程模式定义了开发流程中“谁”、“实现什么”、“如何”和“何时做什么”,其中人员表示“谁”,产品表示“实现什么”,活动表示“如何”,生命周期表示“何时做什么”。因此,软件过程模式与软件生命周期模型的关系为:软件生命周期模型包含于软件过程模式之中。目前在软件界影响较大的软件过程模式包括Rational统一过程、敏捷过程等等。

软件过程模式是对软件过程的抽象描述与表示,它可以是形式化、半形式化的或非形式化的,可以通过人员所实施的活动来完成过程模式所描述的软件过程。软件过程模式的定义包含两个方面的涵义,第一,它是对软件过程的抽象。第二,它反映了人们看待过程的一个视角。

一般来说,一个软件过程模型中通过回答“做什么”、“何时做”、“谁来做”、“怎么做”等基本问题,帮助项目开发人员以及其他参与人员正确理解软件过程。

一个灵活的、强有力的过程模式将给软件开发提供有效的支持和指导,因为该项目为软件过程中的所有活动提供了统一的政策保证,为参与软件开发的所有成员提供帮助和指导,使得软件开发过程按照统一有序的步骤进行。

一个理想的过程模式应满足以下几个准则:

* 能够表示所有活动组织方式
* 能够表示各活动的工作方法和结果
* 能够表示各活动之间存在的相互之间的关系
* 能够体现过程的动态特性
* 易于被所有参与开发活动的人员理解
* 具有很强的灵活性,可以适应不同的软件项目
* 易于对开发过程进行控制和管理

**1.3软件过程描述**

软件过程描述是通过某种形式化的手段对软件开发过程加以系统严格地描

述,为软件开发人员提供一个标准的无歧义的软件开发规范,并以此为基础辅助

和指导开发人员的工作,同时对实际的软件开发过程进行监督和控制,从而保证

软件产品的质量和软件生产率。通过过程描述所表示的过程模式可以看作是软件

开发过程的脚本,它指导开发者如何严格按照工程化的方法一步一步地进行开发

工作,软件则是依据过程描述进行的一系列软件过程活动的产品。软件过程描述

与软件过程之间的关系类似于程序和进程之间的关系,软件过程是软件过程描述

的动态执行,而它如何执行则是由过程描述所决定的。软件有一个开发的过程,

软件过程也有一个开发的过程;软件开发产出软件产品,软件过程开发产出过程

产品,软件过程的实现就是将软件过程文档化,相当于软件开发的编码。

**1.4研究软件过程模式的意义**

1.4.1从软件过程的组成元素上讲

软件过程模式不仅关注软件过程中各生命周期阶段中的活动,更重要的是它

同时关注过程中人员与角色分配、过程中采用的方法及过程各个阶段的输入输出

产品,软件过程中这四大要素相辅相成、相互作用,从而构成一个有机的整体,

缺一不可。相对软件生命周期模型,软件过程模式更全面、深刻、细致地反映了

软件过程中各个层面和各个环节。作为对软件生命周期模型的补充和发展,软件过程模式的四要素及相互关系是项目计划、风险评估、人员管理、质量保证等项

目实践的重要依据,将它用于指导软件开发实践具有现实的可操作性。

1.4.2从过程模式的角度上讲

从软件过程模式的角度认识和分析软件过程能够快速准确地把握软件过程的

思想本质、原则规范、主要特点和实现策略等各个方面。

1.4.3从软件过程的实践角度上讲

软件过程模式的立场角度有助于软件开发人员清楚地比较出各种软件过程的

各自优缺点,进而对这些软件过程进行相应的取舍、结合、定制,然后用于自己

的软件开发实践中,以期以一种可预测的、循环的方式指导软件项目开发获得成

功。

**1.5软件过程改进**

为了实现能按时、按预算、按客户需求完成高质量软件的目标,人们致力于

软件开发新技术、新方法的探索。这些新技术、新方法尽管能为解决软件危机提

供一定帮助,但人们深深体会到对软件过程的管理和控制,对于解决软件危机问

题也是必不可少的,这就需进行过程改进。软件过程改进就是在软件过程工程中,

为了更有效地达到优化软件过程的目的所实施的改善或改变其软件过程的系列活动。过程改进的实施,就是在认知现有软件过程的基础上,利用过程运作所获得

的反馈信息,发现软件过程存在的问题和缺陷,提出改进的意见,进而实现软件

过程的改进和完善。

软件过程改进的战略要建立在软件过程改进环境下的一个整体框架,在框架

中要标识出软件过程改进中必须包括的关键的领域。软件过程改进框架包括以下

相互关联必不可少的四个方面内容:

.软件过程改进架构:为支持过程有两种类型的架构:一个是组织及管理方

面的架构,其包括角色与职责;另一个是技术方面的架构,包括技术工具与设备。

所有这些都为与过程相关的活动提供支持并且维持过程的连续改进活动。

.软件过程改进规划图:它指定一个所采用的改进过程模型,并规划出实现

高效软件过程的行动步骤和方法。改进规划图指明了实现软件过程所要经历的阶段及层次,以及为实现这些目标所必经的关键点。

.软件过程改进评估方法:它指出对当前软件过程、活动以及架构进行评估

所采用的方法及技术。软件过程评估是根据软件改进规划图而进行的。评估的结

果需进行分析,从而找出当前过程的长处与短处,以便为更好地进行改进提供建

议。通过改进才能使我们沿着过程改进规划图不断提高过程的成熟度。

.软件过程改进计划:它所包含的内容是为进行软件过程改进,根据评估所

发现的问题,有针对地制定出相应的改进方案。

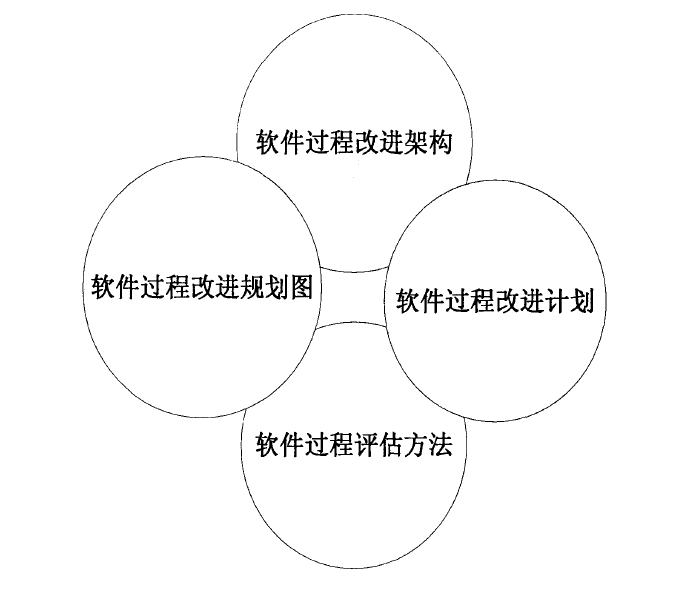


图1-1软件过程改进框架

Figure1-1 Software Proeess Improvement Frame

图1- 1中给出软件改进框架中四个组成部分的关系。框架中的四个相互关联的

部分是必不可少的,否则将会导致软件过程程序中的冲突。从相互关联的角度来

看,这是一种循环关系。一个完整的循环首先从对过程架构的评估开始,评审是

进行变革的催化剂,它与软件过程改进规划图相关联的。它的评审结果记录在软

件过程改进的计划中,以便有针对性地进行改进,从而达到过程改进规划图中更

高的软件过程成熟度级别。

**2敏捷方法论与敏捷过程模型设计**

**敏捷项目管理的概念及起源**

敏捷项目管理的概念来源于敏捷软件开发。随着敏捷软件开发的发展,极限项目管理(也称为极端项目管理Extreme Project Management或Radical Project Management)和敏捷项目管理(也称为灵活的项目管理Agile Project Management)的概念和方法被相继提出,并仍在不断发展。实际上,敏捷项目管理只是各种敏捷软件开发方法相应项目管理的统称,只针对于软件项目,并不是一种通用项目管理方法(也有人提出敏捷项目管理的通用概念,但未被广泛接受)。极限项目管理和适应性项目框架皆源于对Doug De-Carlo于2000年发布的弹性项目模式(Flexible Project Model)的改编。而弹性项目模式又来自于敏捷软件开发中的自适应软件开发方法学的启发。

现在二者已经发展成为一个通用的项目管理理论。极限项目管理适合于变化大、复杂程度高的项目。传统的项目管理则适合低变化、低不确定性的项目。而在二者之间是适应性项目框架。虽然所有的敏捷软件开发方法都被认为是属于极限项目管理的范畴,但从最近的敏捷软件开发的发展可以看出有些敏捷方法并不全属于极限项目管理的范畴。而且极限项目管理往往由于过于激进,显得不够实际,并不能被高级管理者特别是CIO所接受,且在大型项目中也无法得到有效论证。现在的敏捷项目管理研究大多有转向适应性项目框架的趋势。所以,虽然敏捷项目管理通常指的就是极限项目管理,但它被认为应是包括极限项目管理和适应性项目框架两部分的软件项目管理的统称,极限项目管理又是适应性项目框架的特例。

“敏捷”是企业界和学术界为适应日趋激烈的市场竞争和日益复杂的生存环境而提出的一种新概念,它要求企业通过提高敏捷性,在变化中不断调整,从而掌握主动并赢得市场。软件行业在不断的发展,同时市场的竞争也在不断的加剧,整个行业的生产效率有待于进一步的提高,整个软件产业有待于进一步升级,敏捷方法被无数的软件界人士尊为适应变化赢得市场的灵丹妙药,敏捷方法也不断的由理论转向实践。

自1996年以来,敏捷软件过程的研究和实践己取得了一些成果。Blackb盯n等专家分别讨论了基于时间的软件过程和并行软件工程;1998年Aoyama首次提出了敏捷软件过程的概念,他将之定义为基于时间的、并行的、分布的、演化的快速软件开发过程,并在大规模通信软件领域中进行了5年的成功实践。近两年,XP和SCRUM等轻载方法和全球软件开发的研究成为热点。至今,敏捷软件过程已成功用于一些软件公司。

但是,敏捷软件过程的研究和实践仍旧处于初期的探索阶段,系统的理论基础和工程方法还没有形成。大多数相关工作只局限于敏捷软件过程的一个分支。下面我将从软件过程的敏捷性需求、敏捷软件过程的特性出发提出敏捷软件过程的模型与结构,对敏捷软件方法进行全面详尽的研究。

**2.1软件过程的敏捷性要求**

如上所述,软件行业的竞争在不断的加剧,产业生产效率有待于进一步的提高,产业结构正在面临整体的升级。软件开发全球化、个性化、快速化、高质量的新特点要求软件过程具有敏捷性,要求软件企业和软件开发团队能够应对软件功能需求的变化和软件市场的变化,如上所述,软件过程的敏捷性要求主要表现为以下几个方面:

1.快速适应系统需求的变化

软件开发人员常常对需求的变化抱怨。但是,在当前业务环境中需求的变化是不争的事实。当项目开始时,软件需求可能是不明确的,甚至是不知道的。市场和业务的不断变化导致这些需求在开发过程中不断明确并完善。快速地满足客户不断变化的需求,成为软件开发人员乃至整个软件行业最大的挑战,也是软件产品成功的关键。

2.提高软件生产率

软件的发展速度远不如计算机硬件,其生产率长期以来没有跳跃性提升。随着需求日益增大,优质的软件供不应求的现象越来越突出。提高软件生产率,降低软件成本,建立高效的软件过程是软件业的宿愿。

3.突出企业自身特点,体现企业核心竞争力

在全球化经济时代,每个软件企业和其它行业的现代企业一样必须拥有核心竞争力,不求全能,但求有特色的、先进的局部优势。这就要求我们应该针对企业特点,发挥企业自身优势和核心能力,开发企业“自己”的软件过程。

4.支持动态联盟

当市场上新的机遇出现时,企业常常需要组织相关合作伙伴,以最快的速度将各自的核心能力达成最优组合,即虚拟组织,最终取得双赢。因此,动态联盟要求软件过程能和其它过程良好集成,是灵活应变的和可扩充的。

5.面向业务目标持续改进和重组

软件过程必须和企业的业务目标紧密结合,随着业务的重组和优化,快速灵活地变化,及时支持新业务环境下的软件开发模式。同时能依据CMM等过程标准,不断实践,持续改进,逐步建成成熟的软件过程。

**2.2敏捷软件过程的特性**

敏捷软件过程不仅仅简单地意味着软件的快速开发,它着重于对软件需求、过程和产品变化的灵活快速反应,是基于统一概念的一整套技术。和传统的软件过程存在很多的不同之处,敏捷软件过程是一种轻载的、基于时间的、JustEnough的、并行的和基于组件的软件开发过程。

**2.2.1轻载软件过程**

敏捷软件过程是一种“轻载”过程。它认为软件的需求是难以预期的,开发方法必须适应变化的需求,在快速的迭代中不断改进。小组成员并不完全按照完整的方法进行开发,而根据具体问题和情况,灵活地去除非增值活动,抛弃那些复杂的、实际上最后也是很难跟从的规则,仅仅执行一些必须的活动,使用必须的规则,编写必须的文档。整个过程中人的因素被放在第一。值得注意的是,采用轻载过程并不是回复到早期的无序编程状态,而是遵循软件工程的最佳实践原则,使用合适的规则,既保证软件开发质量,又提高效率。轻载过程对需求变化的快速适应性,使其特别适合互联网时代的开发要求。典型的轻载过程有XP和SCRUM,它们分别提出了不同的开发规则,如结对编程。

**2.2.2基于时间的过程**

有别于传统基于规模的软件过程,敏捷软件过程的结构和实施是面向时间的,是一种基于时间的软件开发。每一次迭代有固定的时间限制。一个复杂的项目可被分为多个迭代和多次发放。需求在迭代开始时被确定,直至下一次迭代开始前才能再次修改。

**2.2.3正好策略**

敏捷软件过程是主张结合企业业务的软件过程。这就是“JustEnough”(正好)策略。策略指出,在进行软件过程改进时,应着重领会CMM等过程模型的精神实质和基本原理,而不是简单地生搬硬套,必须根据自己的实际情况,建立适合自己的过程。世界上绝大部分软件企业均属小型企业,因此在实施CMM时,必须研究如何在小企业中正确贯彻CMM,注意照顾过程的多样性。最近,“JustEnough”被不少专家建议为企业过程的基本实施策略。无论是过程还是文档管理,“JustEnough”主张从实际出发,把过程管理与企业的业务目标紧密结合起来,同时探索可满足CMMKPAS的最小关键实践集合。

**2.2.4并行软件工程**

并行工程是1988年制造业提出的一种产品及相关过程的并行化设计方法。它的基本思想是把传统串行产品开发流程转变成集成的、并行的产品开发过程,使工程尽可能并行进行,从而达到加快工程进度、降低产品开发成本的目的。并行工程在制造业中已得到了广泛的应用并取得了良好的效果。敏捷软件过程将并行工程的思想和原则引入到软件领域中,提出了并行软件工程的概念,其核心表现在两个方面:第一方面,活动的并行化。传统的软件开发模式着重于串行,每个阶段仅当通过评审后才能进入下一阶段。并行软件工程站在软件开发全过程的高度,打破传统的各阶段分割封闭的观念,强调开发人员团队协同,注重分析和设计等前段开发工作,避免不必要的返工。当一个活动产生足够信息后,其后序活动就可以在其结束之前开始。例如,大部分设计完成后,在评审确认后就可以开始并行编码,在需求完成后,系统测试用例就可以开始设计。第二方面,全球软件开发。并行活动常常是分布的,随着经济的全球化,多个国家、多个民族、多个地区参与软件的共同开发已成为趋势,称为全球软件开发。全球软件开发利用全球广阔的人才`资源和时区差异,实现24小时不间断开发和大规模的并行软件工程。

**2.2.5基于构件的软件工程**

敏捷软件过程鼓励采用构件化开发方法。在基于构件的软件工程开发模式中,软件模块被抽象和封装为可复用的构件。整个系统就是一组相互连接的构件,构件间仅通过接口发生关系。软件开发将不再一切从头开始,开发的过程就是构件的组装过程,维护的过程就是构件升级、替换和扩充的过程。如何选择合适的构件和组建系统将是开发的重点,软件设计将更面向范式和框架。在整个软件中,除了少量构件要根据具体应用实地开发,大部分则选用商品化的构件。这些构件被良好地抽象和封装,并通过反复测试,具有较好的可靠性、可维护性、可扩展性和可重构性。基于构件的软件工程不仅提高了软件的质量,而且大幅度缩短了软件开发和维护的周期,为软件生产的工业化提供了可能性。

**2.3敏捷软件过程的过程模型设计**

敏捷软件过程和一般软件过程相比,具有十分突出的特点。这就要求它的过程模型除了刻画软件过程的基本元素外,还需要增加典型的对敏捷性的支持。这里,我从功能模型、合作模型、资源模型和产品模型4个侧面定义了敏捷软件过程及其过程模型。

**2.3.1敏捷软件过程的定义**

软件过程是指软件生存周期中用于开发和维护软件和相关产品所采用的一系列活动,这些活动的执行可以是有序的、重复的、并行的、嵌套的,也可以是有条件地引发。敏捷软件过程是一种特殊的软件过程,根据它的内在特征,可形式化定义为一个七元组:

P=(As,Cl,CZ,10,Rs,Ls,Ms),

即{活动,前置条件,后置条件,产品,资源,地点,度量指标}。

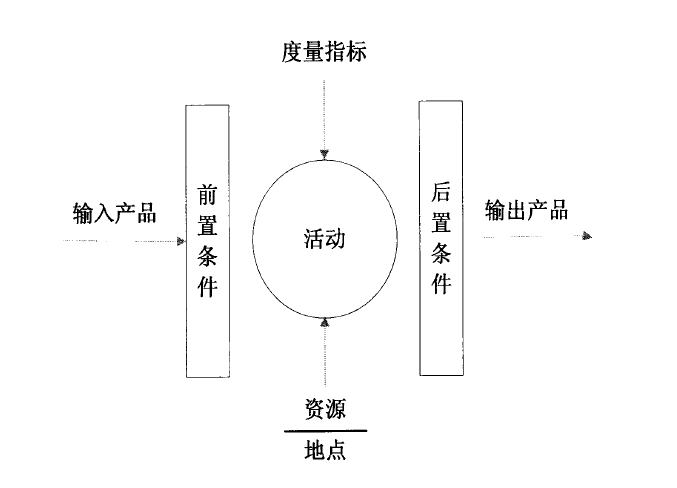


图2-1敏捷软件过程定义

Figure2-1 The Definition of Agile Software Process

活动AS是指过程中包含的所有开发和管理活动,构成了功能模型。活动在人或程序的操作下,将输入产品转换成输出产品。时间、优先级、风险等是活动的关键属性。在一个复杂的软件过程中,一个活动可以细分为多个子活动,从而形成一个树状的层次结构。在此,里程碑被视为一种特殊的活动,由计算机在指定的时间自动执行,检查相关产品是否全部递交。

产品10是指过程或活动中被输入、输出或引用的相应程序、文档和数据,它们是原子的,或者是包括其它产品的复合产品,这些构成了产品模型。约束C1,C2分别为活动的前置条件和后置条件,由活动之间和活动产品之间的约束关系组成,构成了合作模型。

地点U是过程执行的场所。在分布式软件开发中,不同过程或活动可能运行在不同的地点。这些地点分别向过程提供一定的资源Rs,包括人员、工具、机器、技术和场地等,构成了过程的资源模型。度量指标Ms是量化的过程目标。软件过程的敏捷性可归为4个度量指标—时间、生产率、健壮性、自适应和改善能力。软件企业的业绩(KPI)通过软件过程敏捷性的度量指标来体现,从而实现定量管理。敏捷软件过程的一个典型运行为:当一个过程的前置条件满足时,过程主体就可以启动这一过程,产生该过程的一个实例,开始执行过程逻辑,在指定的环境中利用支持技术将输入转化为有效的输出产品;当过程实例的后置条件满足后,用户就可以终止实例运行,提交产品。在整个运行过程中,度量指标随时辅助管理者进行定量分析、评估和决策,不断改进和调整过程,确保其正确高效地实施。

**2.3.2敏捷软件过程的约束条件**

软件过程模型描述不仅是一些过程元素的集合,更重要的是元素之间约束关系的描述。敏捷软件过程的约束关系应该支持用户灵活地运作过程,而不是强制用户按照某个顺序执行活动。因此,它定义了丰富的约束关系类型,从软件过程的数据流和控制流两种方式支持对敏捷软件过程约束关系的灵活描述,包括并发的控制、产品变更的处理和活动的反馈等。

针对特殊要求,还提供了预定义的规则允许用户刻画约束规范。

1.产品约束

软件过程的数据流是活动和产品间的输入/输出关系,构成了过程的产品约束。敏捷软件过程中活动缺省的关系是并发关系,如果两个并发的活动之间存在数据交换,则一个活动提交的产品不必在活动结束后,就可以被另一个活动使用。敏捷软件过程中的产品约束共分为3种,如表2-1所示。

2.时序约束

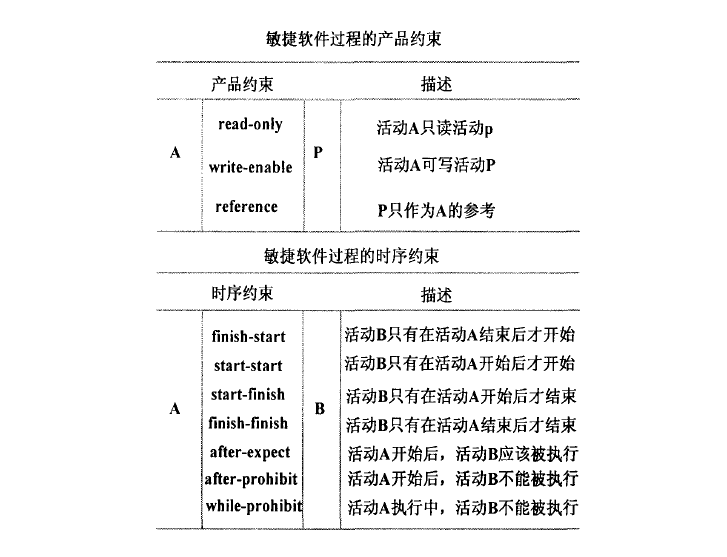
软件过程的控制流是活动和活动间的时序关系,构成了过程的时序约束。敏捷软件过程中的时序约束可以抽象为7种,如表2-1所示。

3.非并发约束

除了明确地指定活动之间的时序关系外,有时过程模型可能需要指定某些活动是非并发的,即同一时间内只能有其中一个活动在执行,但是并不指定这些活动的执行顺序,实际执行顺序由用户决定,称之为非并发约束,如图2-2所示。

表2-1敏捷软件过程约束1

Table2-1 Agile Software Proeess Restraintl



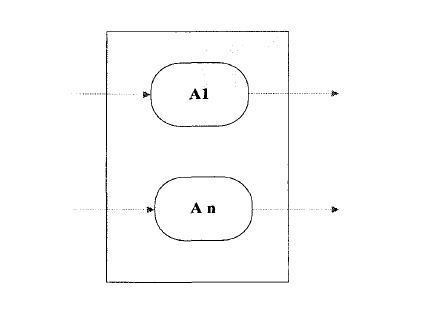


图2-2非并发约束

Figure2-2 Non-concurrent Restraint

**2.3.3敏捷软件过程模型的结构**

敏捷软件过程模型包括活动、地点、资源、产品、约束、度量和过程视图7个基本过程元素,从功能、合作、产品和资源4个侧面进行刻画。活动是过程的核心,又称为子过程,它以树状分层结构组织。一个活动可以细分为多个子活动,从而支持过程自顶而下逐步求精的建模。在每个抽象层次上,用户皆可定义多个不同的过程视图,从各个角度展示过程元素间的关系。我们采用UML定义了敏捷软件过程模型的结构,如图2-3所示:

构建敏捷软件过程是一项循序渐进的工作。过程模型已经为我们提供了分析和优化的基础,在此基础上,要面向企业的业务目标,深入理解软件过程,识别和研究过程模型存在的问题,提出更好的改进方法,逐渐使过程合理化和敏捷化。常用的过程分析方法分为静态和动态两种。静态分析是基于过程模型规范和定义的,动态分析则是通过过程的模拟了解过程的动态特性。无论是静态分析还是态分析,分析的目的都是针对模型进行验证和优化:建模过程是否合理?是否敏捷?是否是期望的过程?过程是否被正确建模?

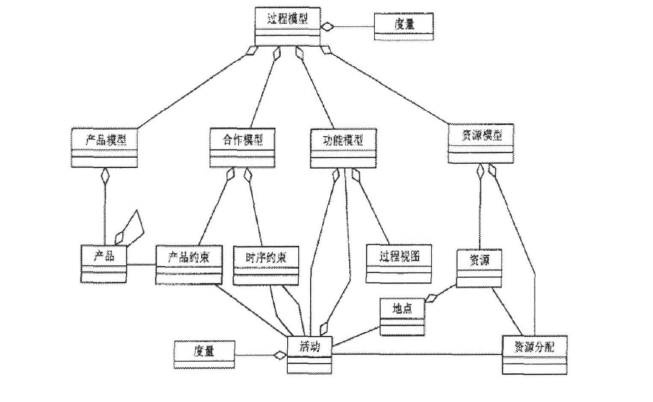


图2-3敏捷软件过程模型的结构

F1gure 2-3 The Structure of Agile Software Process Model

**2.4典型的敏捷方法**

敏捷方法整体上具有适应变化的特点,在其不断的发展过程中,又有很多独特的方法,他们在整体上都属于敏捷方法的范畴,但是又各有特点,下面将着重介绍一种知名的敏捷开发方法一XP方法:

XP方法是敏捷方法之中最引人注目的一个,最早由KeniBeck提出,它其实是一种演进式的原型化方法,具有沟通高效、设计简单、反馈迅速等特点,因而是一种轻载的敏捷的过程方法。

经过软件开发人员的不断探索,XP被认为有四个核心价值:

1.交流:XP强调项目组成员之间的充分的沟通与交流,塑造畅通的沟通渠道,使得项目组的任何人在有任何问题的时候能够及时的反映出来,降低系统风险,打造一个多渠道沟通的快速反应团队。

2.简化:在传统的软件开发中,往往简单的问题被复杂的流程给复杂化,使得问题的本质很难被人理解,让用户代表理解起来更加困难,这就给整个项目带来了很大的风险。简化包括简化流程,简化表述等等,使得项目组的任何人能够很清楚的看到问题的本质,看到问题的所在。

3.反馈:主要是考虑综合运用各种反馈方法,让问题及早的发现,及早解决,反馈包括短迭代,小版本,先测试后编程等等都属于反馈的例子。

4.胆识与勇气:敢于面对压力,放弃不能继续适用的代码,破旧出新;敢于面对现状,当发现系统目前存在的问题时,及时解决。在XP的不断的实践中,设计开发人员总结出十二条实践准则供业内人士参考,它们主要是:

1.团队协作:XP强调团队协作的重要性,这里所说的团队协作不仅来自于设计开发人员,而且也来自于用户。从某种意义上讲,来自于客户的协作比开发团队内的协作更加重要,实施起来更加困难。来自客户的合作更能够为整个项目的进展产生更加重要的价值。

2.计划策略:虽然我们上面强调了接受变化、适应变化的重要性,但是我们不能够满目的去接受变化,对于整个项目的进展以及项目组的管理,不能没有计划,计划在一定的限度内遵守价值原则进行必要的调整。这里所说的计划包括人员计划,交付计划,投资计划,迭代计划等等。计划是建立在对项目全局的角度上思考得出的。

3.系统比拟:既每个系统开发人员对自己负责的模块在系统中的定位有一个清楚的认识,每个系统人员对整个系统的概貌有个大致的认识,以便于开发人员在开展工作的过程中能够积极参与,协调各方。

4.小交付:这是XP的一个重要的实践准则,XP倡导项目组能够及早的开发出一个可以让用户看得到摸得着的软件系统,让用户去体验系统的功能,进而验证系统的功能与操作方式是不是于其预想的一致,进而提出改进意见。小交付能够有效的规避系统风险,发现问题及早解决,使得系统的用户满意度大幅提升。

5.测试先行:XP强调先测试后编码的观点,主张开发团队与测试团队协同工作,根据需求开发测试用例同时根据需求开发系统用例,单元测试的结果要保证完全通过,集成测试分两轮进行,开发人员首先进行集成测试,后由用户代表进行集成测试验证。

6.简化设计:XP强调所有的设计与开发遵循价值原则,所有的用例或者说功能模块按照优先级顺序排列,最重要的模块放在最前面考虑,将问题简单化,设计人员只考虑当前的对系统最重要的模块。这样就剔除了很多的冗余设计,节省了人力物力和时间。

7.结对编程:项目组可以安排能力互补的人员结成开发组,两位开发人员结对编程,提高代码质量,同时提高了开发效率。

8.设计改进:在不影响用户外部满意度的情况下,对系统内部的源码不断的进行调整,本着高内聚低祸合的原则,将系统源码梳理清楚,层次结构分明的源代码能够辅助上面所说的小交付,测试先行等原则。为整个项目的成功打下基础。

9.持续集成:既当有新的模块通过了开发人员的单元测试之后就应该集成到整个系统中来,进而不断的形成新的可以供用户验证的系统,持续的集成能够帮助系统设计开发及早的发现系统架构方面的问题,及早解决,并能够让客户及早的看到一个可运行的系统,有直观的感触,提出改进意见。

10.代码共享:XP强调代码共享机制,任何的开发者可以改进已经共享的代码,以提高代码质量。n.编码风格:xP强调编码的风格尽量统一,其他程序开发人员能够很容易的梳理清楚其来龙去脉,要求编码人员遵守统一的编码规范。

12.可持续性:XP强调开发人员高效的进行工作,整个开发团队步调一致。

**2.5敏捷方法的实践过程中存在的问题**

虽然敏捷过程给软件开发过程带来了新的思想,为解决软件开发过程中需求的多变性提供了解决方案,然而在其实施的过程中,依然存在问题,主要包括以下几个方面:

1.对管理过程要求“刚刚好”,而恰恰是这个使得软件开发人员在实施的过程中不知道如何把握。对于这个标准,需要我们做进一步的理论探讨,需要用形式化的制品去加以规范。

2.到目前为止,敏捷方法主要被小型或者中型的组织应用在小型或者中型的项目中,对于大型的组织和项目的限制还没有被打破,还没有优秀的敏捷过程模型被广大的软件开发人员所认可、所接受并运用于大型的项目之中,这是一个值得研究的方向。

3.敏捷方法非常强调过程中参与者的重要性,但是还没有考虑到文化的差异,地域的差异,性格的差异给人们的参与度造成的各种影响。以上的三个问题是敏捷方法在当前的实践中存在的突出问题,然而这些存在于敏捷思想本身的问题只有通过引入其他成熟的软件过程的思想才能解决,敏捷方法与RUP的有效结合是解决上述的问题一个很好的途径。