RUP读书笔记

ＲＵＰ是一种迭代的、以架构为中心的、用例驱动的软件开发方法（software development approach)。

是一种well-defined and well-structureed software engineering process。它明确的定义了人员的职责、如何完成各项工作及何时完成各项工作。定义了软件开发生命周期的结构，定义了里程碑和决策的关系。

是一个process product,提供了可定制的软件工程的过程框架。可以定制适用于不同开发规模的软件过程。

ＲＵＰ基本原理：

1. 尽早且不断化解重大风险
2. 确保满足用户需求
3. 把注意力放到可执行软件上
4. 尽早在项目中适应变化
5. 在早期确定一个可执行架构
6. 使用构件构造系统
7. 建立高效的开发团队
8. 始终重视质量

需求变化和功能蔓延（feature creep)——由技术或用户驱使的额外功能——是项目中的主要困难源，推迟交付等。  
ＲＵＰ是基于软件过程工程原模型(SPEM)——基于ＵＭＬ的过程建模标准的。分为动态结构（水平），静态结构（垂直）。  
初始阶段  
　　目的：

* 了解项目的范围
* 建立业务模型
* 得到涉众的认可
* 里程碑：生命周期的目的里程碑（Lifecycle Objective Milestone,LCO)

细化阶段  
　　目的：

* 降低主要技术风险
* 建立架构基线
* 弄清构建系统要做哪些工作
* 里程碑：生命周期的架构里程碑（Lifecycle Architecture Milestone,LCA)

构造阶段  
　　目的：

* 开发出产品的第一个可执行版本
* 里程碑：初步可执行能力里程碑（Initial Operational Capability Milestone,IOC)

移交阶段  
　　目的：

* 完成产品的最终版本并交付给用户
* 里程碑：产品发布里程碑（Product Release Milestone,PR)

统一软件过程的特点：用例驱动、以架构为中心、迭代和增量  
1.用例驱动  
  用例体现的是用户(或系统)之间的交互的动作。  
  从外部来描述系统需要提供什么样的功能。  
  是软件开发过程的起点。  
2.以架构为中心  
  架构包含系统的静态和动态特征。  
  软件平台、可重用设施、实施、集成等要求。  
  刻画了系统的整体设计而忽略其中的各个细节。  
3.迭代和增量  
  项目细化。  
  迭代指工作流中的步骤，增量指软件产品中增加的部分。  
  
软件开发的四要素：  
人员：至关重要  
项目：创造产品  
产品：不仅仅是代码和文档  
过程：指导项目  
  
从人员的角度：用户-测试人员-设计人员-分析人员-项目经理-架构设计师  
从过程的角度：用例模型-〉分析模型-〉设计模型-〉实施模型-〉实现模型-〉测试模型  
将人员划分到不同的过程中(略)

RUP详解

一、RUP

Rational统一过程（Rational Unified Process,RUP），它是由Rational软件公司推出的一种完整而且完美的软件过程（关于什么是软件过程点击这里）。它可以作为软件开发的良好开端。

RUP最重要的它有三大特点：1）软件开发是一个迭代过程，2）软件开发是由Use Case驱动的，3）软件开发是以架构设计（Architectural Design）为中心的。

RUP总结了经过多年商业化验证的6条最有效的软件开发经验，这些经验被称为“最佳实践”

**1.最佳实践**

(1) 迭代式开发

在软件开发的早期阶段就想完全、准确的捕获用户的需求几乎是不可能的。实际上，我们经常遇到的问题是需求在整个软件开发工程中经常会改变。迭代式开发允许在每次迭代过程中需求可能有变化，通过不断细化来加深对问题的理解。迭代式开发不仅可以降低项目的风险，而且每个迭代过程都以可执行版本结束，可以鼓舞开发人员。

(2) 管理需求

确定系统的需求是一个连续的过程，开发人员在开发系统之前不可能完全详细的说明一个系统的真正需求。RUP描述了如何提取、组织系统的功能和约束条件并将其文档化，用例和脚本的使用已被证明是捕获功能性需求的有效方法。

(3) 体系结构

组件使重用成为可能，系统可以由组件组成。基于独立的、可替换的、模块化组件的体系结构有助于降低管理复杂性，提高重用率。RUP描述了如何设计一个有弹性的、能适应变化的、易于理解的、有助于重用的软件体系结构。

(4) 可视化建模

RUP往往和UML联系在一起，对软件系统建立可视化模型帮助人们提供管理软件复杂性的能力。RUP告诉我们如何可视化的对软件系统建模，获取有关体系结构于组件的结构和行为信息。

(5) 验证软件质量

在RUP中软件质量评估不再是事后进行或单独小组进行的分离活动，而是内建于过程中的所有活动，这样可以及早发现软件中的缺陷。

(6) 控制软件变更

迭代式开发中如果没有严格的控制和协调，整个软件开发过程很快就陷入混乱之中，RUP描述了如何控制、跟踪、监控、修改以确保成功的迭代开发。RUP通过软件开发过程中的制品，隔离来自其他工作空间的变更，以此为每个开发人员建立安全的工作空间。

2.关键名词解释

（1） 二维开发模型

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）。

（2） 核心概念

RUP中定义了一些核心概念，

角色：描述某个人或者一个小组的行为与职责。RUP预先定义了很多角色。

活动：是一个有明确目的的独立工作单元。

工件：是活动生成、创建或修改的一段信息。

（3） 裁剪

RUP是一个通用的过程模板，包含了很多开发指南、制品、开发过程所涉及到的角色说明，由于它非常庞大所以对具体的开发机构和项目，用RUP时还要做裁剪，也就是要对RUP进行配置。RUP就像一个元过程，通过对RUP进行裁剪可以得到很多不同的开发过程，这些软件开发过程可以看作RUP的具体实例。RUP裁剪可以分为以下几步：

1） 确定本项目需要哪些工作流。RUP的9个核心工作流并不总是需要的，可以取舍。

2） 确定每个工作流需要哪些制品。

3） 确定4个阶段之间如何演进。确定阶段间演进要以风险控制为原则，决定每个阶段要哪些工作流，每个工作流执行到什么程度，制品有哪些，每个制品完成到什么程度。

4） 确定每个阶段内的迭代计划。规划RUP的4个阶段中每次迭代开发的内容。

5） 规划工作流内部结构。工作流涉及角色、活动及制品，他的复杂程度与项目规模即角色多少有关。最后规划工作流的内部结构，通常用活动图的形式给出。

3.开发过程

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。在每个阶段的结尾执行一次评估以确定这个阶段的目标是否已经满足。如果评估结果令人满意的话，可以允许项目进入下一个阶段。

（1） 初始阶段

初始阶段的目标是为系统建立商业案例并确定项目的边界。为了达到该目的必须识别所有与系统交互的外部实体，在较高层次上定义交互的特性。本阶段具有非常重要的意义，在这个阶段中所关注的是整个项目进行中的业务和需求方面的主要风险。对于建立在原有系统基础上的开发项目来讲，初始阶段可能很短。初始阶段结束时是第一个重要的里程碑：生命周期目标（Lifecycle Objective）里程碑。生命周期目标里程碑评价项目基本的生存能力。

（2） 细化阶段

细化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。为了达到该目的，必须在理解整个系统的基础上，对体系结构作出决策，包括其范围、主要功能和诸如性能等非功能需求。同时为项目建立支持环境，包括创建开发案例，创建模板、准则并准备工具。细化阶段结束时第二个重要的里程碑：生命周期结构（Lifecycle Architecture）里程碑。生命周期结构里程碑为系统的结构建立了管理基准并使项目小组能够在构建阶段中进行衡量。此刻，要检验详细的系统目标和范围、结构的选择以及主要风险的解决方案。

（3） 构造阶段

在构建阶段，所有剩余的构件和应用程序功能被开发并集成为产品，所有的功能被详细测试。从某种意义上说，构建阶段是一个制造过程，其重点放在管理资源及控制运作以优化成本、进度和质量。构建阶段结束时是第三个重要的里程碑：初始功能（Initial Operational）里程碑。初始功能里程碑决定了产品是否可以在测试环境中进行部署。此刻，要确定软件、环境、用户是否可以开始系统的运作。此时的产品版本也常被称为“beta”版。

（4） 交付阶段

交付阶段的重点是确保软件对最终用户是可用的。交付阶段可以跨越几次迭代，包括为发布做准备的产品测试，基于用户反馈的少量的调整。在生命周期的这一点上，用户反馈应主要集中在产品调整，设置、安装和可用性问题，所有主要的结构问题应该已经在项目生命周期的早期阶段解决了。在交付阶段的终点是第四个里程碑：产品发布（Product Release）里程碑。此时，要确定目标是否实现，是否应该开始另一个开发周期。在一些情况下这个里程碑可能与下一个周期的初始阶段的结束重合。

4.核心工作流

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）。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。

● 商业建模

商业建模（Business Modeling）工作流描述了如何为新的目标组织开发一个构想，并基于这个构想在商业用例模型和商业对象模型中定义组织的过程，角色和责任。

● 需求

需求（Requirement）工作流的目标是描述系统应该做什么，并使开发人员和用户就这一描述达成共识。为了达到该目标，要对需要的功能和约束进行提取、组织、文档化；最重要的是理解系统所解决问题的定义和范围。

● 分析和设计

分析和设计（Analysis & Design）工作流将需求转化成未来系统的设计，为系统开发一个健壮的结构并调整设计使其与实现环境相匹配，优化其性能。分析设计的结果是一个设计模型和一个可选的分析模型。设计模型是源代码的抽象，由设计类和一些描述组成。设计类被组织成具有良好接口的设计包（Package）和设计子系统（Subsystem），而描述则体现了类的对象如何协同工作实现用例的功能。设计活动以体系结构设计为中心，体系结构由若干结构视图来表达，结构视图是整个设计的抽象和简化，该视图中省略了一些细节，使重要的特点体现得更加清晰。体系结构不仅仅是良好设计模型的承载媒介，而且在系统的开发中能提高被创建模型的质量。

● 实现

实现（Implementation）工作流的目的包括以层次化的子系统形式定义代码的组织结构；以组件的形式（源文件、二进制文件、可执行文件）实现类和对象；将开发出的组件作为单元进行测试以及集成由单个开发者（或小组）所产生的结果，使其成为可执行的系统。

● 测试

测试（Test）工作流要验证对象间的交互作用，验证软件中所有组件的正确集成，检验所有的需求已被正确的实现，识别并确认缺陷在软件部署之前被提出并处理。RUP提出了迭代的方法，意味着在整个项目中进行测试，从而尽可能早地发现缺陷，从根本上降低了修改缺陷的成本。测试类似于三维模型，分别从可靠性、功能性和系统性能来进行。

● 部署

部署（Deployment）工作流的目的是成功的生成版本并将软件分发给最终用户。部署工作流描述了那些与确保软件产品对最终用户具有可用性相关的活动，包括：软件打包、生成软件本身以外的产品、安装软件、为用户提供帮助。在有些情况下，还可能包括计划和进行beta测试版、移植现有的软件和数据以及正式验收。

● 配置和变更管理

配置和变更管理工作流描绘了如何在多个成员组成的项目中控制大量的产物。配置和变更管理工作流提供了准则来管理演化系统中的多个变体，跟踪软件创建过程中的版本。工作流描述了如何管理并行开发、分布式开发、如何自动化创建工程。同时也阐述了对产品修改原因、时间、人员保持审计记录。

● 项目管理

软件项目管理（Project Management）平衡各种可能产生冲突的目标，管理风险，克服各种约束并成功交付使用户满意的产品。其目标包括：为项目的管理提供框架，为计划、人员配备、执行和监控项目提供实用的准则，为管理风险提供框架等。

● 环境

环境（Environment）工作流的目的是向软件开发组织提供软件开发环境，包括过程和工具。环境工作流集中于配置项目过程中所需要的活动，同样也支持开发项目规范的活动，提供了逐步的指导手册并介绍了如何在组织中实现过程。

5.迭代开发模式

RUP中的每个阶段可以进一步分解为迭代。一个迭代是一个完整的开发循环，产生一个可执行的产品版本，是最终产品的一个子集，它增量式地发展，从一个迭代过程到另一个迭代过程到成为最终的系统。传统上的项目组织是顺序通过每个工作流，每个工作流只有一次，也就是我们熟悉的瀑布生命周期。这样做的结果是到实现末期产品完成并开始测试，在分析、设计和实现阶段所遗留的隐藏问题会大量出现，项目可能要停止并开始一个漫长的错误修正周期。

一种更灵活，风险更小的方法是多次通过不同的开发工作流，这样可以更好的理解需求，构造一个健壮的体系结构，并最终交付一系列逐步完成的版本。这叫做一个迭代生命周期。在工作流中的每一次顺序的通过称为一次迭代。软件生命周期是迭代的连续，通过它，软件是增量的开发。一次迭代包括了生成一个可执行版本的开发活动，还有使用这个版本所必需的其他辅助成分，如版本描述、用户文档等。因此一个开发迭代在某种意义上是在所有工作流中的一次完整的经过，这些工作流至少包括：需求工作流、分析和设计工作流、实现工作流、测试工作流。其本身就像一个小型的瀑布项目

RUP的迭代模型与传统的瀑布模型相比较，迭代过程具有以下优点：

降低了在一个增量上的开支风险。如果开发人员重复某个迭代，那么损失只是这一个开发有误的迭代的花费。

降低了产品无法按照既定进度进入市场的风险。通过在开发早期就确定风险，可以尽早来解决而不至于在开发后期匆匆忙忙。

加快了整个开发工作的进度。因为开发人员清楚问题的焦点所在，他们的工作会更有效率。

由于用户的需求并不能在一开始就作出完全的界定，它们通常是在后续阶段中不断细化的。因此，迭代过程这种模式使适应需求的变化会更容易些。