# RUP学习总结

Rational Unified Process（以下简称RUP） 是一套软件工程方法，主要由 Ivar Jacobson的 The Objectory Approch 和 The Rational Approch发展而来。同时，它又是文档化的软件工程产品，所有RUP的实施细节及方法导引均以Web文档的方式集成在一张光盘上，由Rational公司开发、维护并销售，当前版本是5.0。RUP又是一套软件工程方法的框架，各个组织可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 吸收了多种开发模型的优点，具有很好的可操作性和实用性。从它一推出市场，凭借Booch、Ivar Jacobson、以及Rumbagh 在业界的领导地位以及与统一建模语言（Unified Model Language , 以下简称UML）的良好集成、多种CASE工具的支持、不断的升级与维护，迅速得到业界广泛的认同，越来越多的组织以它作为软件开发模型框架。

## 1. 二维的软件开发模型

传统的软件开发模型瀑布式开发模型是一个单维的模型，开发工作划分为多个连续的阶段。在一个时间段内，只能作某一个阶段的工作比如，分析、设计或者实现。

在RUP中，软件开发生生命周期根据时间和RUP的核心工作流划分为二维空间。

时间维从组织管理的角度描述整个软件开发生命周期，是RUP的动态组成部分。它可进一步描述为周期（Cycle）、阶段（phase）、Iteration(迭代)。核心工作流从技术角度描述RUP的静态组成部分，它可进一步描述为行为（activities）、工作流（workflow）、产品（artifact）、角色（worker）。

不同的工作流在不同的时间段内工作量的不同。值得注意的是，几乎所有的工作流，在所有的时间段内均有工作量，只是工作程度不同而已。这与Waterfall process(瀑布式开发模型)有明显的不同。

## 2.静态结构：方法描述

软件开发过程描述了什么时候，什么人，做什么事，以及怎样实现某一特定的目标。RUP采用以下四个基本模型元素组织和构造系统开发过程。

角色 : the who

行为 : the how

产品 : the what

工作流 : the when

角色描述某个人或一个小组的行为与职责。一个开发人员可以同时是几个角色，一个角色也可以由多个开发人员共同承担。RUP预先定义了很多角色，例如：Architect、Use-Case Designer、Course Developer、Implementer …，并对每一个角色的工作和职责都作了详尽的说明。

行为是一个有明确目的的独立工作单元。产品是行为生成、创建或修改的一段信息。它是行为的输入同时又是它的输出结果。产品以多种形式存在，例如：模型（Model）、源代码、可执行文件、文档等。

模型是从某一个角度对系统的完全描述。RUP的很大一部分工作就是设计和维护一系列的模型，这其中有Use Case Model、Business Model、 Analysis Model、Design Model等。所有的这些模型都以UML描述，因此它们是标准的并为多种CASE工具支持。RUP并不鼓励写在字面上的文挡，产品应尽可能地在CASE工具中创建和修改并为版本管理工具跟踪和维护，它们在整个软件开发周期中动态地增加和修改。当然也可以根据需要为模型生成报告（Reports），但它们是静态的，是某一时刻模型的快照不需要维护和修改。

工作流描述了一个有意义的连续的行为序列，每个工作流产生一些有价值的产品，并显示了角色之间的关系。RUP主要提供两种组织工作流的方式：核心工作流（Core Workflow）和迭代工作流（Iteration Workflow）。

核心工作流从逻辑上把相关角色和行为划分为组，以描述RUP的逻辑组成部件。它们相当于模板一样，并不在开发过程中真正的执行。迭代工作流是RUP的一个具体的实现过程，它们对核心工作流进行裁剪，是核心工作流的具体实现。每类工作流都会同一个或多个模型打交道。

## 3.RUP有九个核心的工作流

以下简单描述这些工作流的目的：

商业建模（Business Modeling）：理解待开发系统的组织结构及其商业运作，确保所有参与人员对待开发系统有共同的认识。

需求分析（Requirements）：定义系统功能及用户界面，使客户知道系统的功能，开发人员知道系统的需求，为项目预算及计划提供基础。

分析与设计（Analysis and Design）：把需求分析的结果转化为实现规格。

实现（Implementation）：定义代码的组织结构、实现代码、单元测试、系统集成。

测试（Test）：校验各自子系统的交互与集成。确保所有的需求被正确实现并在系统发布前发现错误。

发布（Deployment）：打包、分发、安装软件，升级旧系统；培训用户及销售人员，并提供技术支持。制定并实施beta测试。

配置管理（Configuration and Change Management）：跟踪并维护系统所有产品s的完整性和一致性。

项目管理（Project Management）：为计划、执行和监控软件开发项目提供可行性的指导；为风险管理提供框架。

环境（Environment）：为组织提供过程管理和工具的支持。

由于版面所限，无法详细解释每一个工作流。前六个核心工作流的名字，很可能使人们同Waterfall Process的顺序工作阶段相混淆。但我们知道核心工作流并不是具体的实现，而核心工作流中的某些行为有可能在软件开发周期中，一遍又一遍地在迭代工作流中得以细化。

## 4.动态结构：迭代式开发

在时间维上，为了能够方便地管理软件开发过程，监控软件开发状态，RUP把软件开发周期划分为Cycles,每个Cycle生成一个产品的新的版本。每个Cycle都依次由四个连续的阶段(pahse)组成，每个阶段都应完成确定的任务。

起始阶段（Inception）：定义最终产品视图、商业模型并确定系统范围。

演化阶段（evaluation）：设计及确定系统的体系结构，制定工作计划及资源要求。

构造阶段（construction）：构造产品并继续演进需求、体系结构、计划直至产品提交。