Week11 – Short Paper Assignment – RUP

SWEN 603 9041

Debashis Jena

University of Maryland Global Campus

# Introduction

As Ahmed states, “When things do not work at extremes, then a middle ground is sought. This is how you can describe Rational Unified Process (RUP). RUP has a linear structure like waterfall models as well as iterative steps like those in agile method.” (Ahmed, 2012). RUP is actually a product of Rational Software. It was developed by Ivar Jacobson in the year of 1996, which was initially named as Objectory Process. In 1998, two new disciplines were added: business modeling and configuration management. Also, performance testing UI design, data engineering and RUP was updated to UML 1.1. In the year 1999, project management discipline was added alongwith some techniques for real time software development. Gradually many other techniques were added to RUP since 2000 with step by step guides to using Rational tools. The company was then acquired by IBM in 2003, and since then the process is named as IBM Rational Unified Process. IBM has the Rational Method Composer which allows customization of processes and RUP has been included in it. It follows six best practices which can be listed as below.

1. Develop iteratively, with risk as the primary iteration driver
2. Manage requirements
3. Employ a component-based architecture
4. Model software visually
5. Continuously verify quality
6. Control changes

As RUP was published it gradually got a lot of popularity. More about the RUP building blocks, project life-cycle phases and some of the method variances will be discussed in this short paper.

# Rational Unified Process Building Blocks

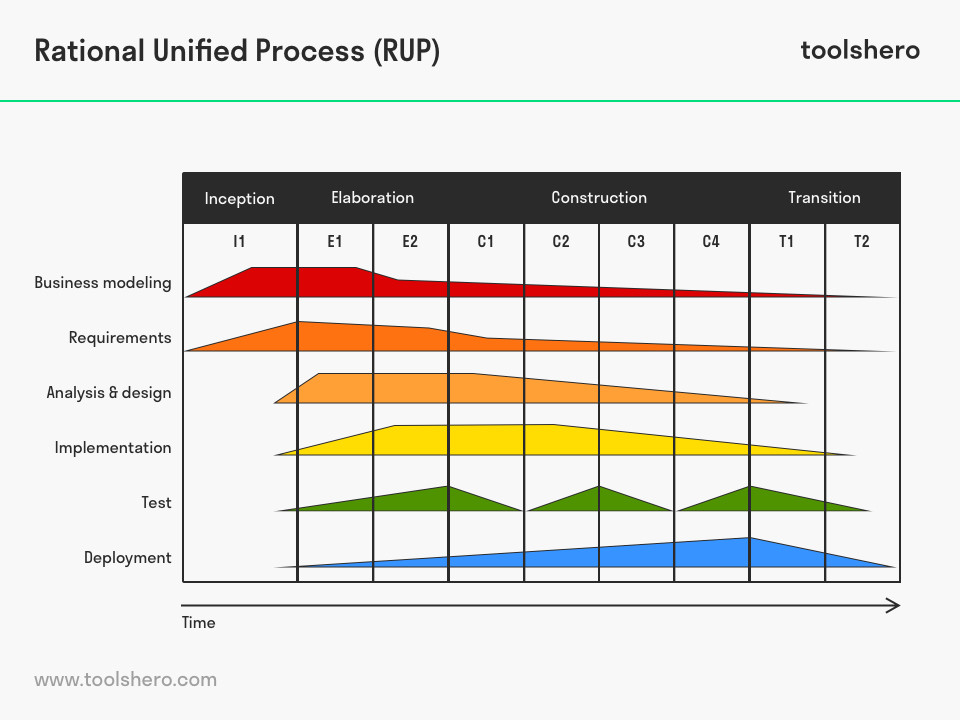
Rational unified process is a process that has the common building blocks as the traditional and agile methods. These building blocks describe “what” is to be produced, the required skills (what) and a step-by-step explanation (how) of the specific development goals. The content elements are as below.

* Role (who) – A set of related skills, competencies and responsibilities makes up a role. For example, a business analyst has the skills and knowledge of how the technical products can be interpreted to the users, which they apply when they come up with the requirements. Similarly an architect has the prior experience in designing many applications.
* Work product (what) – Each of the tasks must have a work product. It does not have to be a part of the application, but can be any part of the deliverables including documents or user manuals.
* Tasks (how) – As the literal meaning goes, task is a unit of work that is assigned to a role. As mentioned above each task must have a deliverable or a part of the deliverables. The tasks may be categorized into nine disciplines, six of which are engineering disciplines and three are supporting disciplines.
  + Engineering disciplines
    - Business Modeling
    - Requirements
    - Analysis and Design
    - Implementation
    - Test
    - Deployment
  + Other supporting disciplines
    - Configuration and Change Management
    - Project Management
    - Environment

# Project Life Cycle Phases

The projects that follow any of the variances of rational unified process go through four life cycle phases. These phases are similar to the same of traditional waterfall model. The entire process follows the process of iterative development. The project life-cycle phases can be described as below.

1. Inception phase – This is where the project proposal is created. This proposal typically includes the initial costing and budgets of the project. Also a project charter may be created describing the expected revenue, market recognition. A project plan may be created with the stakeholder identification, plan for procurement management and risk management.
2. Elaboration phase – This is where the detailed requirements are gathered for the proposed application. The outcome of this phase may include the following.
   1. A use-case model with actors involved and the description of the use cases.
   2. Software architecture
   3. Business cases with the list of risks
   4. A development plan
   5. Prototypes for the project or a proof of concept
   6. Optionally a preliminary user manual
3. Construction phase – This is the core phase of the application development, where the software system is built. Each of the software components is developed. There may be multiple iterations of the development processes. The number of iterations depend on the size and complexity of the project. For larger projects the system is divided into manageable segments and each of them are built to produce prototypes.



1. Transition phase – This is the phase the system is transitioned from one environment to another. Typically this phase involves many deployments and upgrades in the environments. The application is made available for users to use. This phase includes many end users training and creating user manuals.

# RUP Variances

There are many variances of rational unified process. Here is the list of them and a brief description of each of them.

1. Agile Unified Process (AUP) – It is a simplified version of RUP. It uses the agile techniques and principles such as simplicity, agility, tool independence, focus on high-value activities. AUP also follows the principles of Test-driven development (TDD), agile modeling, agile change management and database refactoring.
2. Essential Unified Process (EssUP) – This process uses practices such as iterative development and architecture-driven development. Team practices and process practices are part of EssUp as well.
3. Enterprise Unified Process (EUP) – This process was developed to address a few short comings of the overall RUP; such as the lack of system support and eventual retirement of a software system.
4. Open Unified Process (OpenUP) – In OpenUp, many of the optional parts of RUP have been excluded. However, it includes iterative development, use cases and scenarios driving development, risk management and architecture-centric approach.

**Reference**

Ahmed, Ashfaque. (2012). *Software project management: a process-driven approach*. [Books24x7 version] Available from *http://library.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=47182*.

No Author. (2016). Rational unified process. *Wikipedia, the free encyclopedia.* Retrieved from *https://en.wikipedia.org/wiki/Rational\_Unified\_Process*.