Chapter VII

RUP and eXtreme Programming: Complementing Processes

Gary Pollice
Worcester Institute of Technology, USA

Abstract

The Rational Unified Process®, or RUP®, and eXtreme Programming (XP) are two popular software development methodologies or processes. Most people tend to think of them as opposing methods, where a project may adopt one or the other, but certainly not both. This essay describes how the two can be applied together to improve the software development practices of an organization.

Apples and Oranges

Trying to compare RUP to XP, or vice-versa, is truly like comparing apples to oranges. They were designed for completely different purposes and apply to different software development contexts and different levels of process. This
does not mean that you have to choose one or the other. XP and RUP are based upon similar principles and contain valuable techniques to guide you through software development projects. They can be applied in a complementary manner with a little bit of forethought and planning. The following sections are designed to help you understand both of these practices and how to apply them to your organization and project. These sections are:

- Introduction to RUP and XP
- Core Values and Principles
- Similarities and Differences
- Applying the Practices

**Introduction to RUP and XP**

**RUP**

RUP is not a single process. It is a process framework that describes, in detail, a set of activities, artifacts, and responsibilities that may apply to a software development organization. The framework is designed as a consistent description of an incremental, iterative software development lifecycle. RUP describes process along two dimensions, time and discipline, as shown in Figure 1, which is referred to as the RUP “hump” chart. The disciplines are applied at different levels, depending upon where the project is according to the time dimension. The time dimension describes a software project as it moves through four phases: Inception, Elaboration, Construction, and Transition. At the end of each phase, there is a milestone that determines if the phase is complete, and whether it is appropriate and advantageous to proceed to the next phase. Each milestone provides for a “go or no-go” decision by the stakeholders of the project. The milestones allow timely cancellation of projects that are unfeasible or too costly, usually due to changing requirements and business climate.

The Inception phase is the first of the RUP phases. During this phase you work to understand all of the stakeholders in the project, their needs and expectations, and describe the system’s requirements at a high level. You build a *shared vision* of the project’s results. At the end of the phase, you are in a
Related Content

Understanding Distributed Object Oriented Systems
Alex Podaras (2000). *Object Oriented Technologies: Opportunities and Challenges* (pp. 150-165).
[www.igi-global.com/chapter/understanding-distributed-object-oriented-systems/27352?camid=4v1a](www.igi-global.com/chapter/understanding-distributed-object-oriented-systems/27352?camid=4v1a)

From Bad Smells to Refactoring: Metrics Smoothing the Way
[www.igi-global.com/chapter/bad-smells-refactoring/27362?camid=4v1a](www.igi-global.com/chapter/bad-smells-refactoring/27362?camid=4v1a)

Reuse in Object Oriented Modeling: An Empirical Study of Experienced and Novice Analysts
[www.igi-global.com/chapter/reuse-object-oriented-modeling/27349?camid=4v1a](www.igi-global.com/chapter/reuse-object-oriented-modeling/27349?camid=4v1a)

Design of the Man-Machine Interface: Menus, Inputs, and Outputs
[www.igi-global.com/chapter/design-man-machine-interface/18741?camid=4v1a](www.igi-global.com/chapter/design-man-machine-interface/18741?camid=4v1a)