



The Menlo Institute LLC

The Rational Unified Process[®] (RUP)

A Well Documented, Complete yet Complex Methodology

**Thomas Meloche © 2002
Fellow, The Menlo Institute**

**The Menlo Institute LLC
410 N. Fourth Ave, 3rd Floor
Ann Arbor, MI 48104
www.menloinstitute.com
(734) 665-1847**

**The Rational Unified Process[®]
The Unified Modeling Language[®]
Rational[®] are registered by
Rational Software Corporation.**

Introduction

In order to mitigate problems described previously in proprietary software development processes, base your process framework on open, published and supported standards. Two software development processes stand out as strong candidates for consideration:

- Extreme Programming (XP)
- The Rational Unified Process® (Unified Process)

These processes are well documented and publicly available. In this paper, we will briefly describe The Rational Unified Process® and highlight why it is so important and how it addresses specific shortcomings of proprietary development processes.

The Rational Unified Process®

The Rational Unified Process® is based on the integrated work of three primary methodologists, Ivar Jacobson, Grady Booch and James Rumbaugh. These methodologists, aided by a large and extended methodologist community, were assembled by Rational Corporation to form a unified, cohesive and comprehensive methodology framework for the development of software systems. Their work, occurring over several years and based on existing, previously tested methodologies, have lead to significant standards in the development community, including the general acceptance of Use Cases and the Unified Modeling Language™ (UML).

The Unified Process has three distinguishing characteristics. These characteristics are:

- **Use-Case Driven** - The process employs Use Cases to drive the development process from inception to deployment.
- **Architecture-Centric** - The process seeks to understand the most significant static and dynamic aspects in terms of software architecture. The architecture is a function of the needs of the users and is captured in the core Use Cases.
- **Iterative and Incremental** - The process recognizes that it is practical to divide large projects into smaller projects or mini-projects. Each mini-project comprises an iteration that results in an increment. An iteration may encompass all of the workflows in the process. The iterations are planned using Use Cases.

Four Process Phases

The Unified Process consists of cycles that may repeat over the long-term life of a system. A cycle consists of four phases: Inception, Elaboration, Construction and Transition. Each cycle is concluded with a release, there are also releases within a cycle. Let's briefly review the four phases in a cycle:

- **Inception Phase** - During the inception phase the core idea is developed into a product vision. In this phase, we review and confirm our understanding of the core business drivers. We want to understand the business case for why the project should be attempted. The inception phase establishes the product feasibility and delimits the project scope.
- **Elaboration Phase** - During the elaboration phase the majority of the Use Cases are specified in detail and the system architecture is designed. This phase focuses on the "Do-Ability" of the project. We identify significant risks and prepare a schedule, staff and cost profile for the entire project.
- **Construction Phase** - During the construction phase the product is moved from the architectural baseline to a system complete enough to transition to the user community. The architectural baseline grows to become the completed system as the design is refined into code.
- **Transition Phase** - In the transition phase the goal is to ensure that the requirements have been met to the satisfaction of the stakeholders. This phase is often initiated with a beta release of the application. Other activities include site preparation, manual completion, and defect identification and correction. The transition phase ends with a postmortem devoted to learning and recording lessons for future cycles.

Core Workflows

The Unified Process identifies core workflows that occur during the software development process. These workflows include Business Modeling, Requirements, Analysis, Design, Implementation and Test. The workflows are not sequential and likely will be worked on during all of the four phases. The workflows are described separately in the process for clarity but they do in fact run concurrently, interacting and using each other's artifacts.

The Unified Process book and on-line documentation provide extensive information for implementing the process. It captures activities and artifacts for each workflow complete with examples. It also provides complete descriptions of workers and their roles, activities and artifacts during each of the phases. An excellent and easy to follow introduction to the process is Philipp Kruchten's book "The Rational Unified Process® , An Introduction."

Unified Modeling Language

The Unified Modeling Language™ (UML) was developed in conjunction with The Unified Process. Throughout the entire Unified Process lies the idea of creating models of the system being constructed. Models represent abstract views of the system from a particular point of view. These models are captured and communicated using UML.

UML is a powerful tool for some people and multiple books have been published on it including two by the process authors Booch, Rumbaugh and Jacobson:

- o The Unified Modeling Language™ User Guide
- o The Unified Modeling Language™ Reference Manual

These books may be used as the definitive references on UML. It is also recommended that you acquire the easier to read Martin Fowler book "UML Distilled."

A Large Process

The Unified Process and its accompanying text require significant study. They are, in many ways, an academic study of the topic. The texts, though complete, are very intimidating to most people. The best way to start is with the on-line documentation, along with formal training in the process. Find a mentor who can work directly with your team to introduce the workflows and activities into your organization.

It is important to recognize that the process is meant to be a living thing. It must be adjusted to your work environment and work habits. The trick is knowing when to adjust the process and when to adjust your habits.

The Unified Process provides a powerful framework for application development. It identifies necessary activities and helps you layout a formal plan for the software development process.

Standard Process Qualifications

All of the requirements necessary for a complete development process are fully captured in the Unified Process Workflows.

- **Open and public** - The Unified Process is openly published, distributed and supported. The Unified Process is documented, coherent and complete. In fact, the process is modeled and documented using the process itself. As a result, thousands of software developers have already been trained in the Unified Process. Even more software developers have

already been trained in key supporting technology such as UML.

- **Complimentary documentation** - A complete description of the Unified Process with sample deliverables is available on-line. Four texts by the primary creators of the Unified Process also exist:
 - The Rational Unified Process® - Philippe Kruchten
 - Unified Software Development Process - Ivar Jacobson, et al
 - The Unified Modeling Language™ Reference Manual - James Rumbaugh
 - The Unified Modeling Language™ User Guide - Grady Booch, et al

There are an additional 66 books readily available by a variety of authors on applying and using the Unified Process and UML. Also, there are hundreds of on-line white papers, articles and case studies.

- **Training readily available** - The on-line version of the Unified Process walks users through the process in a step-by-step tutorial manner. The Rational Corporation offers training on the Unified Process, tools and UML. The Menlo Institute offers training on the Unified Process, tools and UML and multiple other vendors offer training on the Unified Process, tools and UML.
- **Experienced process developers**
 - Grady Booch
 - James Rumbaugh
 - Ivar Jacobson
 - Philippe Kruchten
 - Extended cross-industry contributing team
- **Supporting tools**
 - Rational Rose™ for Business Modeling, Analysis and Design
 - Rational RequisitePro™ for Requirements Tracking
 - Rational Unified Process® for Process Training and Templates
 - Rational ClearQuest™ for Bug Tracking and Change Requests

- o Rational ClearCase™ for Configuration Management

Because the Unified Process has been widely and publicly disseminated there are multiple tool choices from other vendors, all designed to work specifically with the Unified Process.

Unified Process Conclusion

The Menlo Institute considers the Rational Unified Process® to be a well documented and complete methodology. We use it as an interesting source of ideas and tools and offer extensive training on its techniques and practices. If you decide to use The Unified Process, we can confidently support your implementation initiatives.

However, unless you have a *real* expert on-staff it is likely that you will not significantly increase your likelihood of success trying to adopt this process. The process is too complex, too difficult to learn, and too difficult to apply correctly. If you don't have an expert, an expert who has actually delivered similar projects using this process, then either hire or rent one and plan to engage the expert for at least one year.

The Unified Process does not capture the sociological aspects of software development and the details of how to truly develop incrementally. In order to complement your Unified Process initiative consider studying the core development practices of Extreme Programming (XP).

About the Menlo Institute

Founded in 2001 by Thomas Meloche, Richard Sheridan, James Goebel and Robert Simms, The Menlo Institute specializes in Software Development Process and Methodology with a special focus on an agile software development environment called The Software Factory.

The Software Factory

The Software Factory is a complete agile software development methodology created at Menlo. The Software Factory combines best practices from Alan Cooper's Interaction Design, Kent Beck's Extreme Programming, The Rational Unified Process and Six Sigma. The Menlo Institute has unique insight from real-world experience on how these practices can work together effectively.



Thomas Meloche
President
The Menlo Institute

30 people to over 240 people which generated revenue of over 100 million dollars. As a result, he is intimately familiar with the real problems that befall real software development projects. He also knows how to rectify them. During his tenure his delivery organization had a remarkable 100% successful delivery record.

Thomas has a B.C.E. in Computer Engineering from the University of Michigan, and especially enjoys providing training back to his alma mater. He has taught classes and provided mentoring for both the University of Michigan Law and Medical Schools.

Thomas Meloche is President of The Menlo Institute LLC and a founding member of Menlo Associates LLC. His passion is software development best practices and building successful software teams. Before founding The Menlo Institute, Thomas was the Director of Engineering at AppNet, Inc. For over seven years, he led a consulting staff that grew from

Menlo Institute Classes by Category		
Extreme Programming (XP)	Foundation Courses	Rational Unified Process (RUP)
Introduction to XP Coaching XP Teams Writing Story Cards XP BOOT Camp / Immersion	Secrets of Software Success Six Sigma Software Building a Software Factory Object Technology Overview	Managing with RUP-Lite Capturing Reqs With Use Cases Practical UML RUP BOOT Camp / Immersion
Requirements Gathering	Object Technology	Project Management
Capturing Reqs With Use Cases Writing Story Cards	Object Technology Overview Obj. Oriented Analysis & Design Advanced Obj. Oriented Design	Managing with RUP-Lite Software Project Mgmt Six Sigma Software