



The Rap on RUP™ : An Introduction to the Rational Unified Process™

Jeff Jacobs

Jeffrey Jacobs & Associates

Belmont, CA

phone: 650.571.7092

email: jeff@jeffreyjacobs.com

<http://www.jeffreyjacobs.com>



Survey

- Does your organization have a well defined methodology/process?
- Does your organization use OOA/OOD?
- Does your organization use UML?



Agenda

- What is RUP?
- RUP Fundamentals
- Phases
- Product “features”
- Caveats
- Summary
- Questions



Process/Methodology Product Presentation

- Minimal UML bashing
- No rhyming
- No comparison with other methodologies
- RUP appears to be in flux since Rational's acquisition by IBM



Why RUP™ at SPIN?

- *The (RUP™) Knowledge base allows development teams to gain the full benefits of the industry standard UML*
- RUP™ covers all UML models
- RUP™ is **hot**; the latest silver bullet...



What is RUP™?

- A “software engineering process” (methodology)
- A knowledge base “process product”
 - CD to create web site
- UML model focused, not “paper documents” (but...)



What is RUP™?

- Configurable process/product
 - Recognizes and supports variety of different project types
 - Support for tailoring and configuring project web sites
- Project oriented

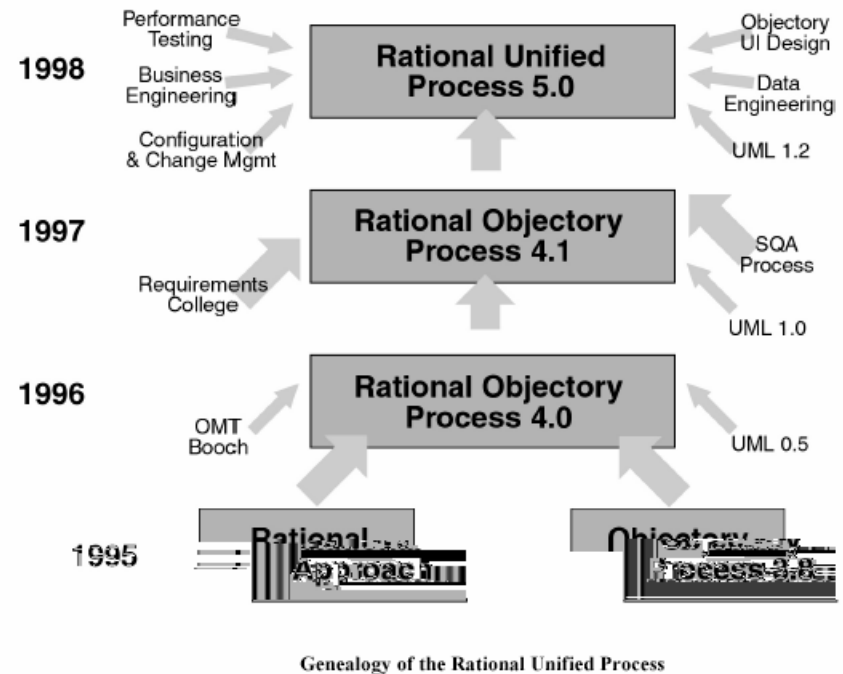


3 Flavors of RUP

- Generic - not dependant on specific technology
- Microsoft Web Solution Technology
 - Additional templates, guidelines etc
- IBM Websphere™ Technology

History

- Methodology by Merger & Acquisition
- Objectory Process created in '80s
- Rational Approach created in '80s
- Acquisition of RequisitePro
- IBM acquires Rational





Why Should You Care About RUP™?

- Your organization is at SEI/CMM Level 1
“Ad Hoc”
 - Provides an excellent path to CMM Levels 2 and 3
- You need to add OOA/OOD to your current process/methodology
- Management wants to know when you’re going to use the latest silver bullet



RUP™ Fundamentals

- RUP is object and process oriented
 - Data takes a back seat
- Architecture is “key to success”
 - Emphasizes need for prototyping of core functionality, not just UI
- Iterative development
- Use Cases are (were?) primary requirements specification technique



4 Phases

- Inception
- Elaboration
- Construction
- Transition



Inception

- Establish business case and business models
- Establishes initial “vision”, high level requirements via “business” use cases
- Create stakeholder “buy in”
- Evaluate risks and return



Elaboration

- Detailed requirements
- Architecture and prototype
- Design



Construction

- Coding and testing



Transition

- Putting the product in the user's hands
- Highly variable, depending on product
 - Data migration
 - Training
 - Parallel Operations
 - Beta testing
 - Etc.

Overview of RUP™ (Organization)

Rational Unified Process: Overview

Disciplines

	Inception	Elaboration	Construction	Transition
Business Modeling	High	Medium	Low	None
Requirements	High	Medium	Low	None
Analysis & Design	Low	High	Medium	Low
Implementation	None	Low	High	Medium
Test	None	Low	High	Medium
Deployment	None	None	High	Medium
Configuration & Change Mgmt	Low	Medium	High	Medium
Project Management	High	Medium	Low	None
Environment	Low	Medium	High	Medium

Iterations

Initial Elab #1 Elab #2 Const #1 Const #2 Const #N Tran #1 Tran #2

Click on an area of the screen for more information.

The Rational Unified Process® or RUP® product is a software engineering process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.

The preceding figure illustrates the overall architecture of the RUP, which has two dimensions:

The horizontal axis represents time and shows the lifecycle aspects of the process as it unfolds.



Best Practices

- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Model visually
- Continuously verify quality
- Control changes



Key Concepts of RUP™

- Organized by *discipline*
- *Workflow* - model of process for a discipline
- *Workflow Details* - 2nd level detail of workflow, detailing activities, roles and artifacts
- *Role* - who performs an activity
- *Activity* - defined piece of work that results in an artifact



More Key Concepts

- *Artifact* - a deliverable, may be document, model, code, etc
 - Templates and examples for many artifacts
- *Tool Mentor* - guide on using Rational Tools for RUP™



Analyst Roles

- Business-Model Reviewer
- Business Designer
- Business-Process Analyst
- System Analyst
- Requirements Specifier
- Test Analyst
- User-Interface Designer



Developer Roles

- Capsule Designer
- Code Reviewer
- Database Designer
- Implementer
- Integrator



More Roles

- Testers
- Managers
- Process Engineer
- Project Manager
- Change Control Manager
- Configuration Manager
- Deployment Manager
- Project Reviewer
- Test Manager



Disciplines

- A collection of related activities that are related to a major 'area of concern' within the overall project
- Disciplines span phases



RUP™'s Disciplines

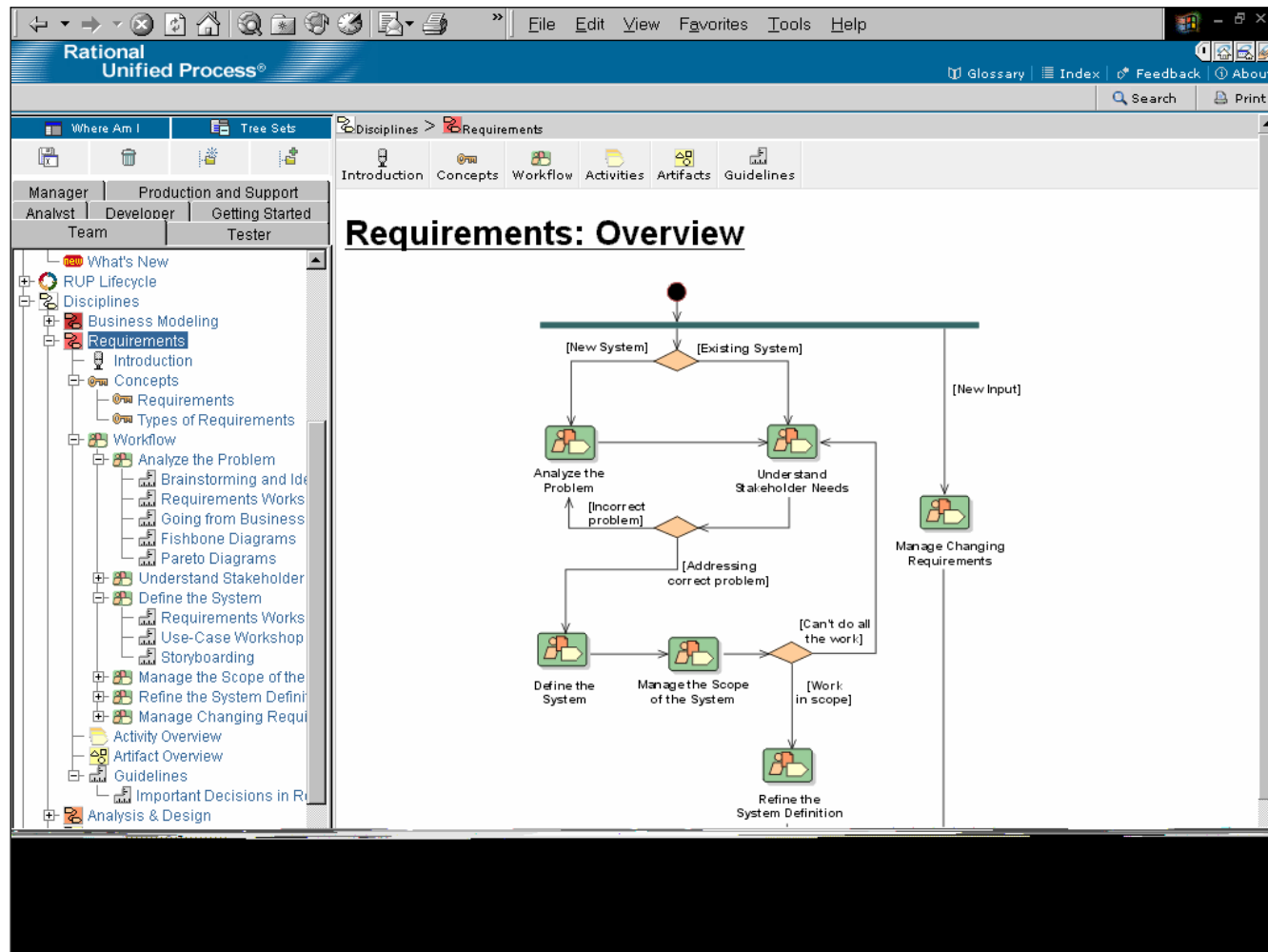
- Business Modeling
- Requirements
- Analysis and Design
(Analysis <>
Requirements, not
performed by “analyst”
role)
- Implementation
- Test
- Deployment
- Configuration and
Change Management
- Project Management
- Environment



Each Discipline is Composed of:

- Overview
- Introduction
- Concepts
- Workflow - the high level activity diagram (process flow)
- Workflow detail - second level process
- Activities - actions performed by roles
- Artifacts - deliverables
- Guidelines - tutorials, checklists, etc

Discipline Overview



Introduction

The screenshot displays the Rational Unified Process (RUP) software interface. The main window is titled 'Introduction to Requirements'. The left-hand navigation pane shows a tree structure with 'Requirements' expanded to 'Introduction'. The main content area is divided into sections: 'Introduction to Requirements', 'Purpose', and 'Relation to Other Disciplines'. The 'Purpose' section contains a bulleted list of goals:

- To establish and maintain agreement with the customers and other stakeholders on what should do.
- To provide system developers with a better understanding of the system requirements.
- To define the boundaries of (delimit) the system.
- To provide a basis for planning the technical contents of iterations.
- To provide a basis for estimating cost and time to develop the system.
- To define a user-interface for the system, focusing on the needs and goals of the users.

Below the list, the text explains that to achieve these goals, it is important to understand the definition and scope of the problem being solved. It mentions that the Business Rules, Business Use-Case Model, and Business Analysis Model developed during Business Modeling will serve as valuable input to the Requirements phase. It also states that Stakeholders are identified and Stakeholder Requests are elicited, gathered, and analyzed. A Vision document, a use-case model, use cases, and Supplementary Specification are described as ways to describe the system - what the system will do - in an effort that views all stakeholders, including potential users, as important sources of information (in addition to system requirements). Finally, it notes that Stakeholder Requests are both actively elicited and gathered from existing sources to get a clear picture of what different stakeholders of the project (customers, users, product champions) expect or desire.

Concepts

- Fundamentals of discipline/role

The screenshot displays the Rational Unified Process (RUP) web application interface. The browser window title is "Rational Unified Process". The main content area is titled "Requirements: Concepts" and contains a list of links: "Requirements" and "Types of Requirements". Below the links, there is a copyright notice: "Copyright © 1987 - 2003 Rational Software Corporation" and the text "Rational Unified Process 2003.06.00.05.06". The left sidebar shows a tree view of the RUP process, with "Requirements" expanded to show "Concepts", "Workflow", "Activity Overview", "Artifact Overview", and "Guidelines". The top navigation bar includes "Introduction", "Concepts", "Workflow", "Activities", "Artifacts", and "Guidelines". The bottom status bar shows "Applet RupPresenterApplet started" and "Internet".

Concept Example

Rational Unified Process®

Requirements > Concepts > Requirements

Concepts: Requirements

A **requirement** is defined as "a condition or capability to which a system must conform".

There are many different kinds of requirements. One way of categorizing them is described as the **FURPS+** model [GRA92], using the acronym FURPS to describe the major categories of requirements with subcategories as shown below.

- [Functionality](#)
- [Usability](#)
- [Reliability](#)
- [Performance](#)
- [Supportability](#)

The "+" in FURPS+ reminds you to include such requirements as:

- [design constraints](#)
- [implementation requirements](#)
- [interface requirements](#)
- [physical requirements](#).

(See also [IEEE Std 610.12.1990].)

Functional requirements specify actions that a system must be able to perform, without taking physical constraints into consideration. These are often best described in a [use-case model](#) and in [use cases](#). Functional requirements thus specify the input and output behavior of a system.

Requirements that are not functional, such as the ones listed below, are sometimes called **non-functional requirements**. Many requirements are non-functional, and describe only attributes

http://www.rational.com/eval/rup/process/workflow/requirem/co_req.htm#Interface Requirement

Jeffrey Jacobs & Associates, Silicon
Valley SPIN

December 3, 2003

30

Everything is Use-Case Driven

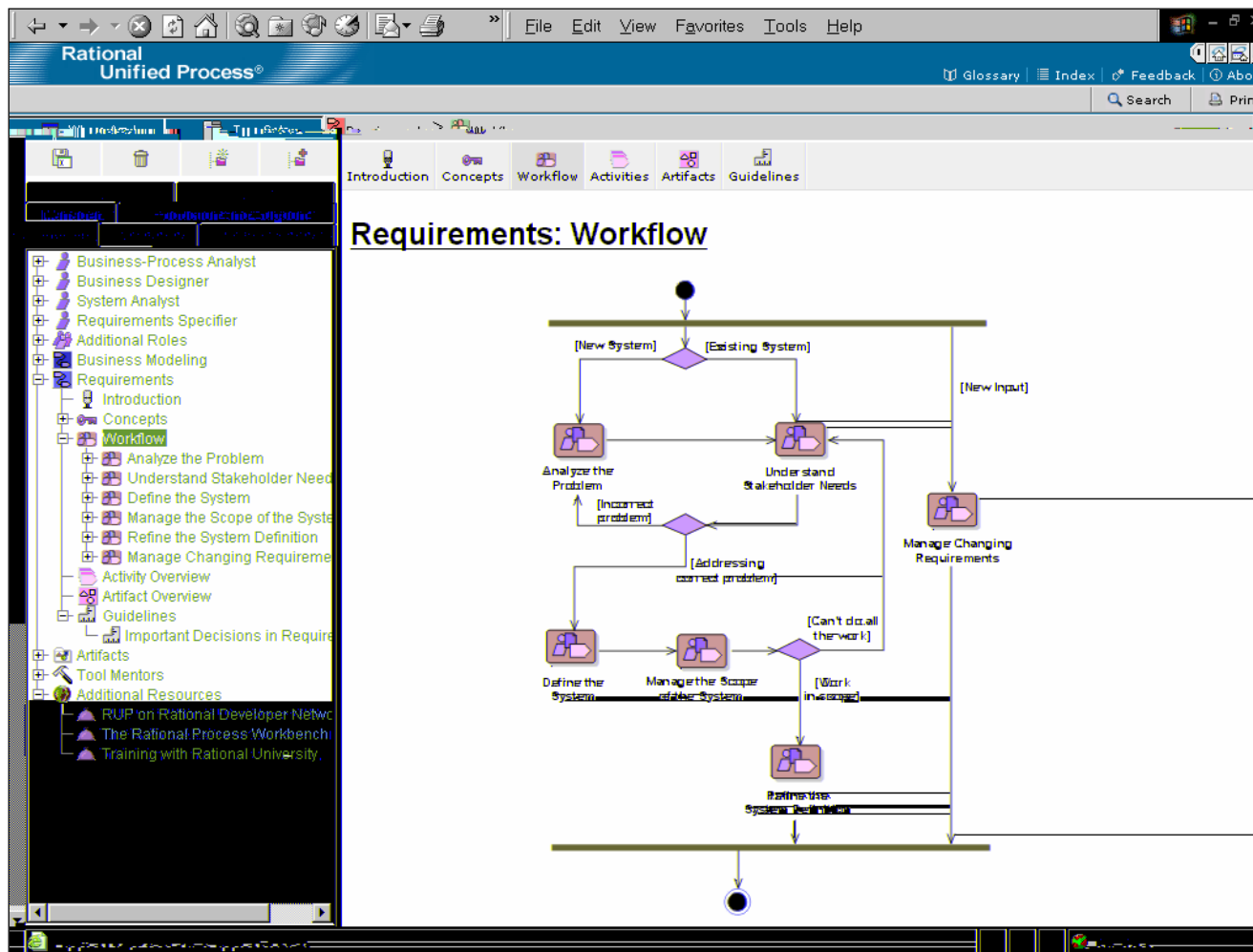
The screenshot displays the Rational Unified Process (RUP) software interface. The main content area is titled "Concepts: Use-Case View" and contains the following text:

To provide a basis for planning the technical contents of iterations, an architectural view called the **use-case view** is used in the Requirements discipline. There is only one use-case view of the system, which illustrates the use cases and scenarios that encompass architecturally significant behavior, classes, or technical risks. The use-case view is refined and considered initially in each iteration.

Below the text is a diagram illustrating the relationship between the Use-Case Model and the Use-Case View. The diagram shows a box labeled "The Use-Case Model" containing a "Top-Level Package" box. An arrow points from "The Use-Case View" to the "Top-Level Package" box, with the text "shows an architecturally significant subset of" next to it.

The interface also features a left-hand navigation pane with a tree view and a right-hand pane with a project structure tree.

Workflow



Workflow Detail

Rational Unified Process®

Requirements > Workflow > Analyze the Problem

Workflow Detail: Analyze the Problem

The purpose of this workflow detail is to gain agreement on the problem being solved. Analysis of the problem involves identify the stakeholders, define the boundary of, and identify the constraints imposed on the system.

Topics

- Description
- Related Information
- Timing
- Optionality
- How to Staff
- Work Guidelines

Description

The first step in any problem analysis is to make sure that all parties involved in the problem (stakeholders, customer, or opportunity) are well represented.

The diagram illustrates the workflow for 'Analyze the Problem'. It shows a central yellow box containing the activities 'Capture a Common Vocabulary' and 'Develop Requirements Management Plan'. This central box is connected to various artifacts and stakeholders. Stakeholders include Customer, End User, and Other. Artifacts include Iteration Plan, Vision, Glossary, Software Development Plan, Requirements Management Plan, Vision, Business Rule, Use Case Model, and Stakeholder. The central activities are supported by 'Find Actors and Use Cases' and 'Develop Vision'.

Activity

The screenshot displays the Rational Unified Process (RUP) software interface. The main window title is "Rational Unified Process®". The browser-like address bar shows "System Analyst" and "Capture a Common Vocabulary". The left sidebar contains a tree view of the RUP process, with "Capture a Common Vocabulary" selected under the "System Analyst" role. The main content area displays the details for this activity:

Activity: Capture a Common Vocabulary

Purpose

- To define a common vocabulary that can be used in all textual descriptions of the system, especially in use-case descriptions.

Role: [System Analyst](#)

Steps

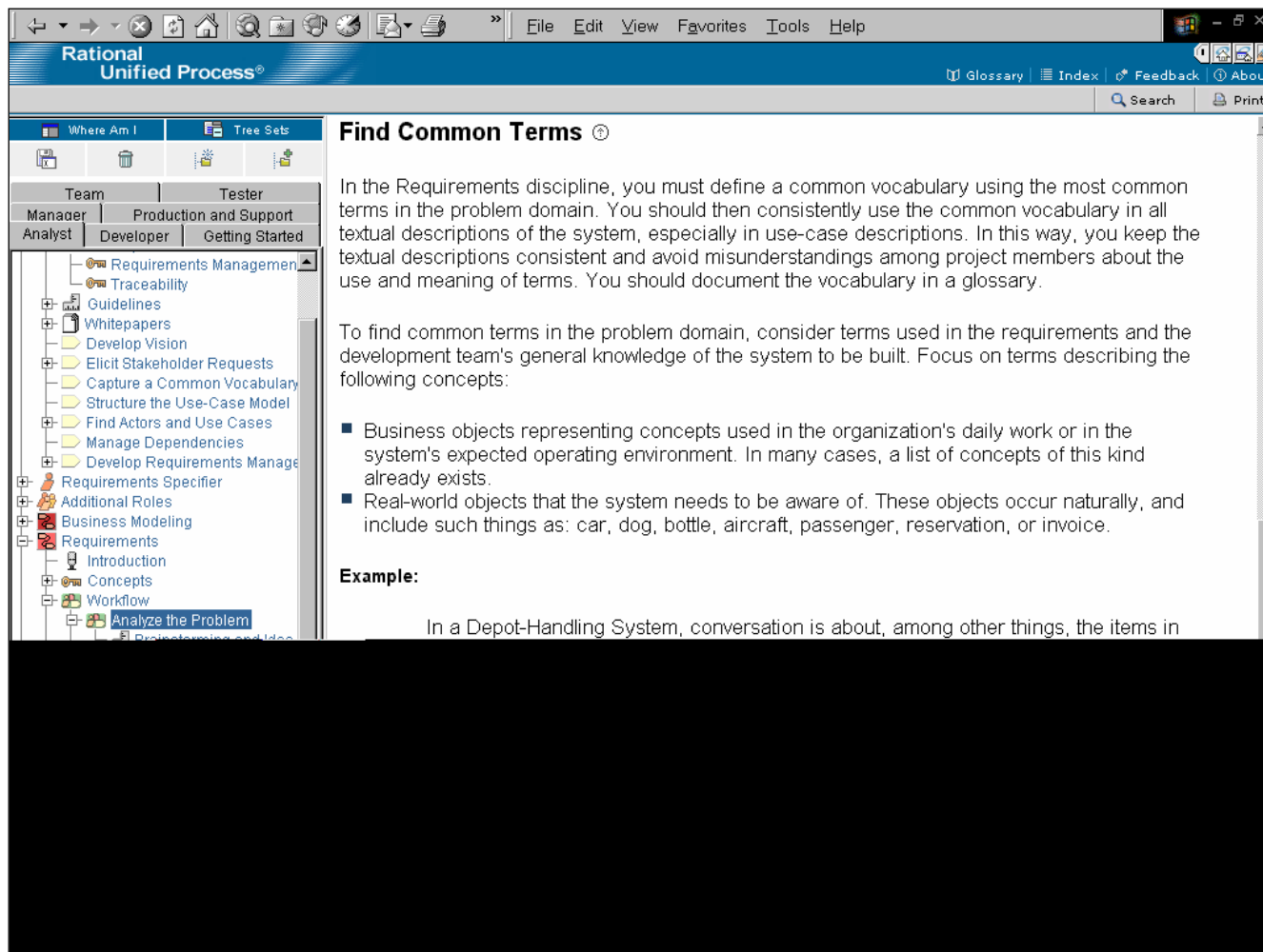
- [Find Common Terms](#)
- [Evaluate Your Results](#)

Input Artifacts:	Resulting Artifacts:
<ul style="list-style-type: none">Business Analysis ModelBusiness CaseBusiness RuleBusiness Use Case ModelStakeholder RequestsUse CaseUse-Case ModelVision	<ul style="list-style-type: none">Glossary

Tool Mentors:

- [Capturing a Common Vocabulary Using Rational RequisitePro](#)

Activity Step



The screenshot displays the Rational Unified Process (RUP) software interface. The main window is titled "Find Common Terms" and contains the following text:

Find Common Terms

In the Requirements discipline, you must define a common vocabulary using the most common terms in the problem domain. You should then consistently use the common vocabulary in all textual descriptions of the system, especially in use-case descriptions. In this way, you keep the textual descriptions consistent and avoid misunderstandings among project members about the use and meaning of terms. You should document the vocabulary in a glossary.

To find common terms in the problem domain, consider terms used in the requirements and the development team's general knowledge of the system to be built. Focus on terms describing the following concepts:

- Business objects representing concepts used in the organization's daily work or in the system's expected operating environment. In many cases, a list of concepts of this kind already exists.
- Real-world objects that the system needs to be aware of. These objects occur naturally, and include such things as: car, dog, bottle, aircraft, passenger, reservation, or invoice.

Example:


In a Depot-Handling System, conversation is about, among other things, the items in

The interface also shows a navigation pane on the left with a tree view containing the following items:

- Requirements Management
- Traceability
- Guidelines
- Whitepapers
- Develop Vision
- Elicit Stakeholder Requests
- Capture a Common Vocabulary
- Structure the Use-Case Model
- Find Actors and Use Cases
- Manage Dependencies
- Develop Requirements Management
- Requirements Specifier
- Additional Roles
- Business Modeling
- Requirements
- Introduction
- Concepts
- Workflow
- Analyze the Problem

Artifacts May Be Documents, Models, Code, Etc.

The screenshot displays the Rational Unified Process (RUP) interface. On the left is a tree view of project artifacts, including Storyboard, Software Requirements Specification, Use-Case Model, and Business Modeling Artifact Set. The main window shows the definition for a 'Glossary' artifact. The definition includes a description, role, optional occurrence, templates, examples, UML representation, and more information. It also lists input and output activities.

	The Glossary defines important terms used by the project.	
Role:	System Analyst	
Optionality/Occurrence:	Primary artifact used to capture information about the project's business domain. Inception and Elaboration phases.	
Templates and Reports:	<ul style="list-style-type: none"> ■ Template: Glossary 	
Examples:	<ul style="list-style-type: none"> ■ CREG Glossary - Elaboration Phase ■ CREG Glossary - Inception Phase ■ CSPS Glossary V1.0 ■ CSPS Glossary V2.0 	
UML Representation:	Not applicable.	
More Information:	<ul style="list-style-type: none"> ■ Checklist: Glossary 	
	<ul style="list-style-type: none"> ■ Purpose ■ Timing ■ Responsibility ■ Tailoring 	
Input to Activities:	<ul style="list-style-type: none"> ■ Architectural Analysis ■ Assess Viability of Architectural Proof-of-Concept ■ Detail a Use Case ■ Detail the Software Requirements ■ Find Actors and Use Cases 	Output from Activities: <ul style="list-style-type: none"> ■ Capture a Common Vocabulary



Document Templates

- Templates for document artifacts available in a variety of formats
 - Microsoft Word
 - HTML
 - Framemaker
 - Rational SODA
- Business Glossary Template

Guidelines

Rational Unified Process®

File Edit View Favorites Tools Help

Glossary Index Feedback About

Search Print

Where Am I Tree Sets

Team Tester

Manager Production and Support
Analyst Developer Getting Started

Business-Process Analyst
Business Designer
System Analyst
Concepts
Requirements Management
Traceability
Guidelines
Whitepapers
Develop Vision
Elicit Stakeholder Requests
Capture a Common Vocabulary
Structure the Use-Case Model
Find Actors and Use Cases
Manage Dependencies
Develop Requirements Management
Requirements Specifier
Additional Roles
Business Modeling
Requirements
Introduction
Concepts
Workflow
Activity Overview
Artifact Overview
Guidelines
Important Decisions in Requirements
Artifacts
Tool Mentors
Additional Resources
RUP on Rational Developer Network
The Rational Process Workbench
Training with Rational University

Guidelines: Important Decisions in Requirements

Topics

- Decide How to Perform the Workflow
- Decide How to Use Artifacts
- Decide Which Reports to Use
- Decide How to Maintain "Input Requirements"
- Decide How to Approve Use Cases

Decide How to Perform the Workflow

The following decisions should be made regarding the Requirements discipline's workflow:

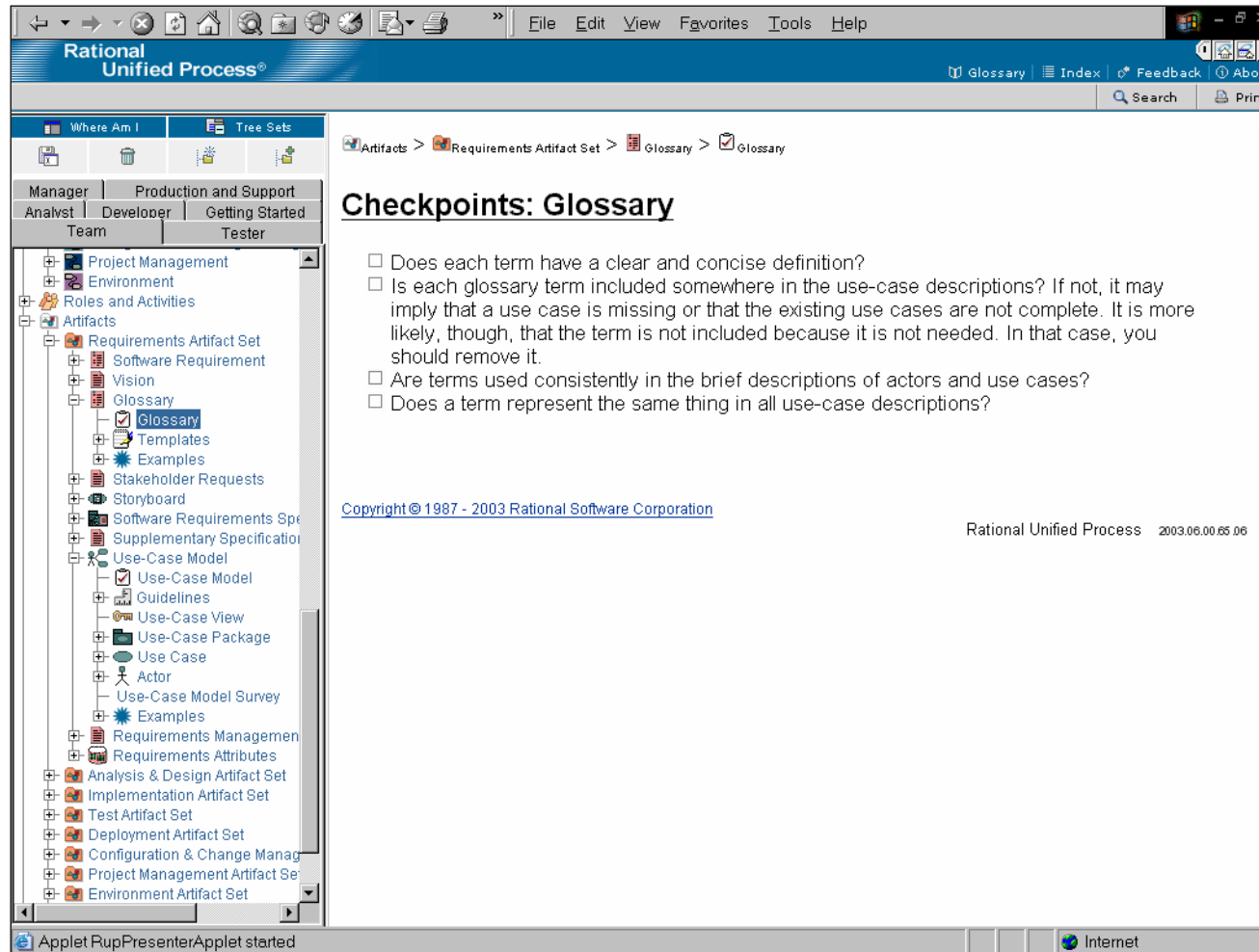
- Decide how to perform the workflow by looking at the [Requirements: Workflow](#). Study the diagram with its *guard conditions*. Decide which workflow details to perform and in which order.
- Decide what parts of the Requirements workflow details to perform. The table below shows some parts that can be introduced relatively independently from each other.
- Decide when, during the project lifecycle, to introduce each part of the workflow. As a general rule, the Requirements discipline should be introduced early in the project.

Part of workflow	Comments
Use-Cases	Some projects do not employ use-cases, which means that the project will not develop artifacts such as a Use-Case Model, Use-Case Package and Use Case. Instead use the Software Requirements Specification.
Workflow Detail: Manage Changing Requirements	This can be introduced after a few iterations in the project when there is a stable baseline.

AppletRupPresenterApplet started

Internet

Checkpoints for Quality Reviews



The screenshot shows the Rational Unified Process software interface. The left-hand pane displays a hierarchical tree structure of artifacts, with 'Glossary' selected under the 'Requirements Artifact Set'. The main window displays the following content:

Artifacts > Requirements Artifact Set > Glossary > Glossary

Checkpoints: Glossary

- Does each term have a clear and concise definition?
- Is each glossary term included somewhere in the use-case descriptions? If not, it may imply that a use case is missing or that the existing use cases are not complete. It is more likely, though, that the term is not included because it is not needed. In that case, you should remove it.
- Are terms used consistently in the brief descriptions of actors and use cases?
- Does a term represent the same thing in all use-case descriptions?

[Copyright © 1987 - 2003 Rational Software Corporation](#)

Rational Unified Process 2003.06.00.65.06

Applet RupPresenterApplet started



More Stuff

- Sample Projects
- Project Management Templates
- Tool Mentors



Sample Projects

- Examples of many artifacts for two projects
 - Course Registration System
 - Collegiate Sports Paging Systems

Glossary Example

The screenshot shows the Rational Unified Process (RUP) software interface. On the left is a tree view of project artifacts, including Business-Process Analyst, Business Designer, System Analyst, Requirements Specifier, Additional Roles, Business Modeling, Requirements, Artifacts, Requirements Artifact Set, Software Requirement, Concepts, Requirements, Types of Require, Vision, Stakeholder Request, Requirements Attribution, Requirements Management, Templates, Examples, CREG Vision - In, CSPS Vision V1.0, Glossary, CREG Glossary, CREG Glossary, CSPS Glossary V, CSPS Glossary V, Stakeholder Requests, Storyboard, and Software Requirements. The main window displays a 'Revision History' table and a 'Glossary' section.

Revision History

Date	Version	Description	Author
26/Dec/1998	1.0	Draft version	Bill Collings
19/Feb/1999	2.0	Expand glossary. Moved some of the terms to the Wylie College glossary.	Bill Collings

Glossary

1. Introduction

The glossary contains the working definitions for terms and classes in the Course Registration System. This glossary will be expanded throughout the life of the project. Any definitions not included in this document may be included in the Rational Rose Model. Generic terms used outside this project should be captured in the organizational Glossary.

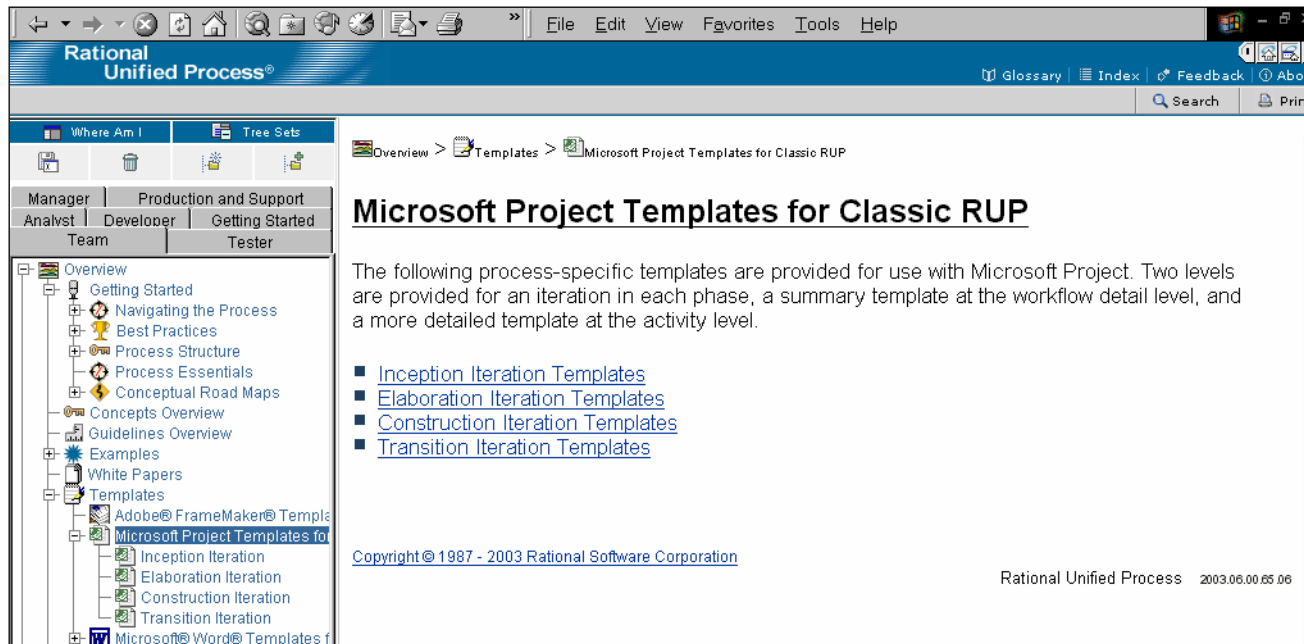
2. Definitions

Alternative course selection
A student might choose to register for one or more alternative courses, in case one or more of the primary selections are not available.

Billing System
Part of the university's Finance System used for processing billing information.

Prerequisite
The university requires for some courses that a student has passed one or more other courses to be

Project Management Templates



- Summary MS Project Example
- Detail MS Project Example

Tool Mentors - How to Use Rational Tools in RUP™

The screenshot displays the Rational Unified Process web application. The main content area is titled "Mentors: Adding Templates to Your Rational RequisitePro Project". Below the title, there is a "Purpose" section and a list of links for further information, including "Develop Requirements Management Plan", "Support Development", and "Prepare Templates for the Project". The left navigation pane shows a tree view of tool categories, with "Rational RequisitePro" expanded to show various tool-specific topics. The bottom of the screen shows the status bar with "Applet RupPresenterApplet started".

Jeffrey Jacobs & Associates, Silicon Valley SPIN

December 3, 2003

45



Caveats

- RUP™ is far from complete
 - Focused on software development
 - Series of books on *Unified Process ... Phase* by Scott Ambler and Larry Constantine provide “missing” coverage
- Project oriented
- Lack of comprehensive “book” makes learning difficult



More Caveats

- IBM influence is creating some confusion and inconsistencies
- Use Cases are insufficient for good requirements, IMHO
- Fails to adequately address “data intensive” applications
 - Only addresses database design, no place for “data requirements”
- Viewed as “silver bullet” by many



Summary

- Forms solid basis for improving software development process, particularly for ad-hoc, Level 1 organizations
- Provides basis for incorporating OOA/OOD/UML into current software development process
- Provides basis for development using IBM, Rational and Microsoft technologies
- 30 day on-line evaluation available, <http://www.rational.com>



Questions
