

# Rational Suite®

## Introduction

VERSION: 2002.05.00

PART NUMBER: 800-025078-000

WINDOWS/UNIX



**IMPORTANT NOTICE**

**COPYRIGHT**

Copyright ©1998-2001, Rational Software Corporation. All rights reserved.

Part Number: 800-025078-000

Version Number: 2002.05.00

**PERMITTED USAGE**

THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION WHICH IS THE PROPERTY OF RATIONAL SOFTWARE CORPORATION ("RATIONAL") AND IS FURNISHED FOR THE SOLE PURPOSE OF THE OPERATION AND THE MAINTENANCE OF PRODUCTS OF RATIONAL. NO PART OF THIS PUBLICATION IS TO BE USED FOR ANY OTHER PURPOSE, AND IS NOT TO BE REPRODUCED, COPIED, ADAPTED, DISCLOSED, DISTRIBUTED, TRANSMITTED, STORED IN A RETRIEVAL SYSTEM OR TRANSLATED INTO ANY HUMAN OR COMPUTER LANGUAGE, IN ANY FORM, BY ANY MEANS, IN WHOLE OR IN PART, WITHOUT THE PRIOR EXPRESS WRITTEN CONSENT OF RATIONAL.

**TRADEMARKS**

Rational, Rational Software Corporation, Rational the e-development company, ClearCase, ClearQuest, ClearQuest MultiSite, ProjectConsole, PureCoverage, Purify, Quantify, Rational, Rational Rose, Rational Suite, RequisitePro, RUP, SoDA, TestFactory, AnalystStudio, Rational Process Workbench, Rational Suite AnalystStudio, Rational Suite ContentStudio, Rational Suite Enterprise, Rational Unified Process, SiteLoad, TestStudio, among others, are either trademarks or registered trademarks of Rational Software Corporation in the United States and/or in other countries. All other names are used for identification purposes only, and are trademarks or registered trademarks of their respective companies.

Microsoft, the Microsoft logo, FrontPage, the Microsoft Internet Explorer logo, Visual C++, Visual Studio, Windows, the Windows logo, Windows NT, and the Windows Start logo are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries.

The Sun J2EE Patterns are used with permission from the book "Core J2EE Patterns" by Deepak Alur, John Crupi, and Danny Malks, published by Sun Microsystems Press/Prentice Hall. Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303. All rights reserved. SUN PROVIDES EACH J2EE PATTERN "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

FLEXIm and GLOBEtrotter are trademarks or registered trademarks of GLOBEtrotter Software, Inc. Licensee shall not incorporate any GLOBEtrotter software (FLEXIm libraries and utilities) into any product or application, the primary purpose of which is software license management.

Portions Copyright ©1992-2001, Summit Software Company. All rights reserved.

**PATENT**

U.S. Patent Nos. 5,193,180 and 5,335,344 and 5,535,329 and 5,835,701. Additional patents pending.

Purify is licensed under Sun Microsystems, Inc., U.S. Patent No. 5,404,499.

**GOVERNMENT RIGHTS LEGEND**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Rational Software Corporation license agreement and as provided in DFARS 277.7202-1(a) and 277.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 227-14, as applicable.

**WARRANTY DISCLAIMER**

This document and its associated software may be used as stated in the underlying license agreement. Rational Software Corporation expressly disclaims all other warranties, express or implied, with respect to the media and software product and its documentation, including without limitation, the warranties of merchantability or fitness for a particular purpose or arising from a course of dealing, usage, or trade practice.

# Contents

<b>Preface</b> .....	<b>ix</b>
Audience .....	ix
Other Resources .....	ix
Rational Suite Documentation Roadmap - Windows .....	x
Rational Suite Documentation Roadmap - UNIX .....	xi
Contacting Rational Technical Support .....	xii
<b>1 Welcome to Rational Suite</b> .....	<b>13</b>
Principles of Software Development .....	13
Rational Suite Can Help .....	16
What's in Rational Suite? .....	16
Optimize Your Team's Productivity .....	16
Tools That Unify Your Team .....	17
Rational Suite Team Unifying Platform .....	17
Accelerate Project Delivery with Professional Services .....	19
Simplify the Solution for Your Team's Development Needs .....	19
Rational Suite: Summary .....	20
Incorporate Best Practices into Your Team's Process .....	21
The Rational Unified Process at a Glance .....	21
Adopting the Rational Unified Process .....	22
The Rational Unified Process and Rational Suite .....	23
What's Next? .....	23
<b>2 Defining the Right System: The Analyst</b> .....	<b>25</b>
Understand the Problem Before You Invest .....	25
Iterative Development and the Challenge of Managing Requirements .....	26
Capture and Manage Requirements with Rational Suite AnalystStudio .....	27
Defining Requirements for Your Team .....	27
Managing Change for Your Project .....	28
Helping Your Team Communicate Visually .....	28
Measuring Progress and Providing Project Reports .....	29
Capitalizing on the Power of Use Cases to Test All Dimensions of Quality .....	29
Summary .....	30

<b>3</b>	<b>Managing Complexity: The Developer</b>	<b>31</b>
	Managing a Complex Process	31
	Analysis and Design: The Next Steps in Iterative Development	31
	Creating Resilient Component-Based Architectures	
	with Rational Suite DevelopmentStudio	33
	Communicating Visually with Models of Your System	33
	Accelerating Code Implementation	33
	Keeping Code and Models Consistent	33
	Evaluating Changing Requirements	34
	Validating Changes to the System	34
	Managing Changes to the System	34
	Keeping the Team Up to Date	35
	Testing Code Early and Often	35
	Tracking Test Results	36
	Creating Component-Based Executable Architectures with	
	Rational Suite DevelopmentStudio—RealTime Edition	38
	Building Complex, Real-Time Systems	38
	Working the Way Real-Time Systems Operate	38
	Summary	39
<b>4</b>	<b>Deciding to Release: The Tester</b>	<b>41</b>
	Making the Crucial Decision to Release	41
	Subsystem and System Tests: The Last Step in Iterative Development	41
	Verifying Software Quality with Rational Suite TestStudio	43
	Unifying the Team by Keeping Members Informed	43
	Making a Plan and Measuring Progress	44
	Does Your Application Meet Requirements?	44
	Is Your Application Reliable?	44
	Does Your Application Have Memory Leaks?	45
	Is Your Application Fast Enough?	45
	Does Your System Perform Under Production Load?	45
	Have You Tested Enough?	45
	Optimizing Defect Tracking	45
	Summary	46

<b>5</b>	<b>Managing Change and Risk:</b>	
	<b>The Project Leader and Web Teams</b>	<b>47</b>
	How Well Can You Plan Ahead? . . . . .	47
	Rational Suite: The Complete Solution for Iterative Development . . . . .	49
	Implementing Best Practices . . . . .	49
	Developing Complex Products Using Unified Change Management . . . . .	50
	Change Request Management . . . . .	51
	Configuration Management . . . . .	51
	Managing Project Status Using Unified Software Project Management . . . . .	52
	Managing Change and Risk in Web Development . . . . .	52
	Using Templates to Accelerate Web Site Changes . . . . .	53
	Deploying Quickly and Confidently . . . . .	53
	Summary . . . . .	54
<b>6</b>	<b>Next Steps</b>	<b>55</b>
	Adopting Practices and Tools . . . . .	56
	Adopting All of Rational Suite . . . . .	56
	Adopting Rational Suite Gradually . . . . .	56
	Exploring Rational Suite Tools and the Rational Unified Process . . . . .	57
	Contacting Rational Professional Services . . . . .	57
	Assessment Services . . . . .	57
	Rational University . . . . .	58
	Rational Customer Support . . . . .	58
	Online Technical Resources . . . . .	58
	Additional Resources . . . . .	58
	<b>Glossary</b>	<b>59</b>
	<b>Index</b>	<b>65</b>





# Preface

This manual provides an introduction to Rational Suite for Windows and UNIX users. Rational Suite delivers a comprehensive set of integrated tools that embody software engineering best practices and span the entire software development lifecycle. The unparalleled level of integration with Rational Suite improves communication both within and across teams, reducing development time and improving software quality.

## Audience

---

This manual is intended for all members of a software development team, including managers, project leaders, analysts, developers, and testers.

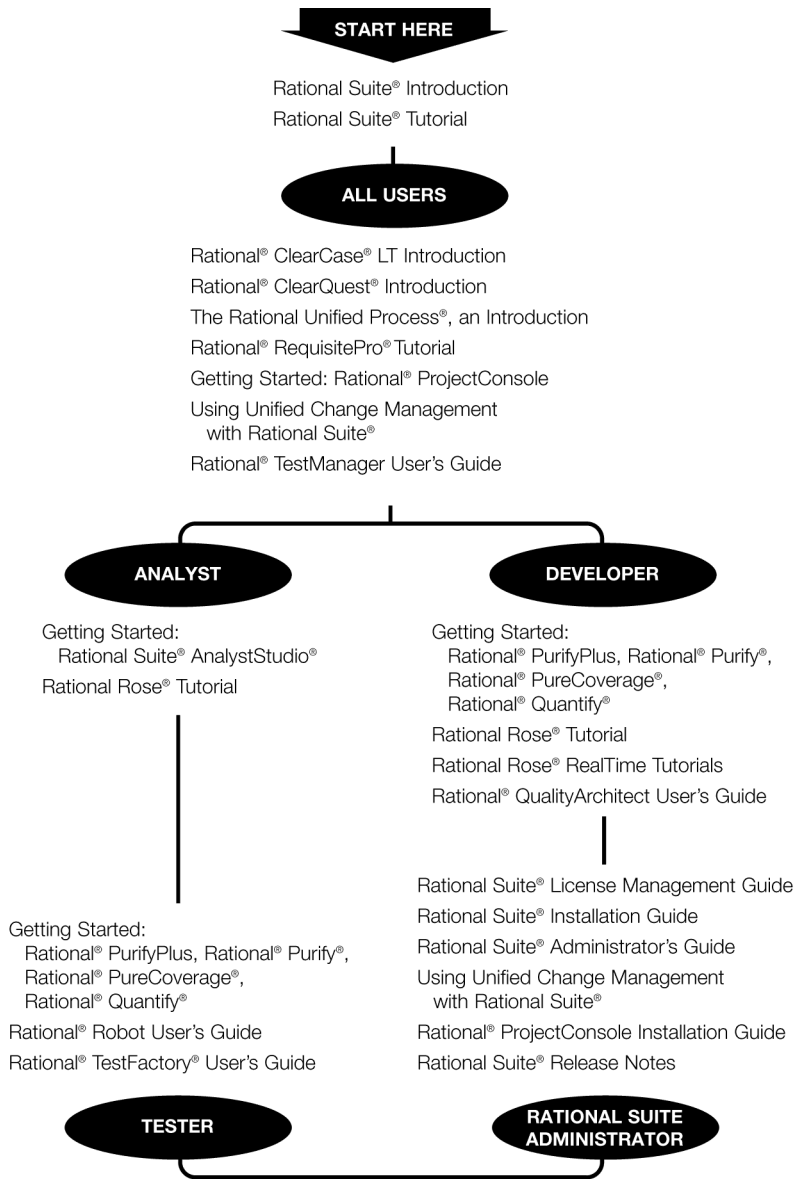
## Other Resources

---

- All manuals are available online, either in HTML or PDF format. The online manuals are on the Rational Solutions for Windows Online Documentation CD.
- To send feedback about documentation for Rational products, please send e-mail to [techpubs@rational.com](mailto:techpubs@rational.com).
- For more information about Rational Software technical publications, see: <http://www.rational.com/documentation>.
- For more information on training opportunities, see the Rational University Web site: <http://www.rational.com/university>.
- For articles, discussion forums, and Web-based training courses on developing software with Rational Suite products, join the Rational Developer Network. Click **Start > Programs > Rational Suite > Logon to the Rational Developer Network**.

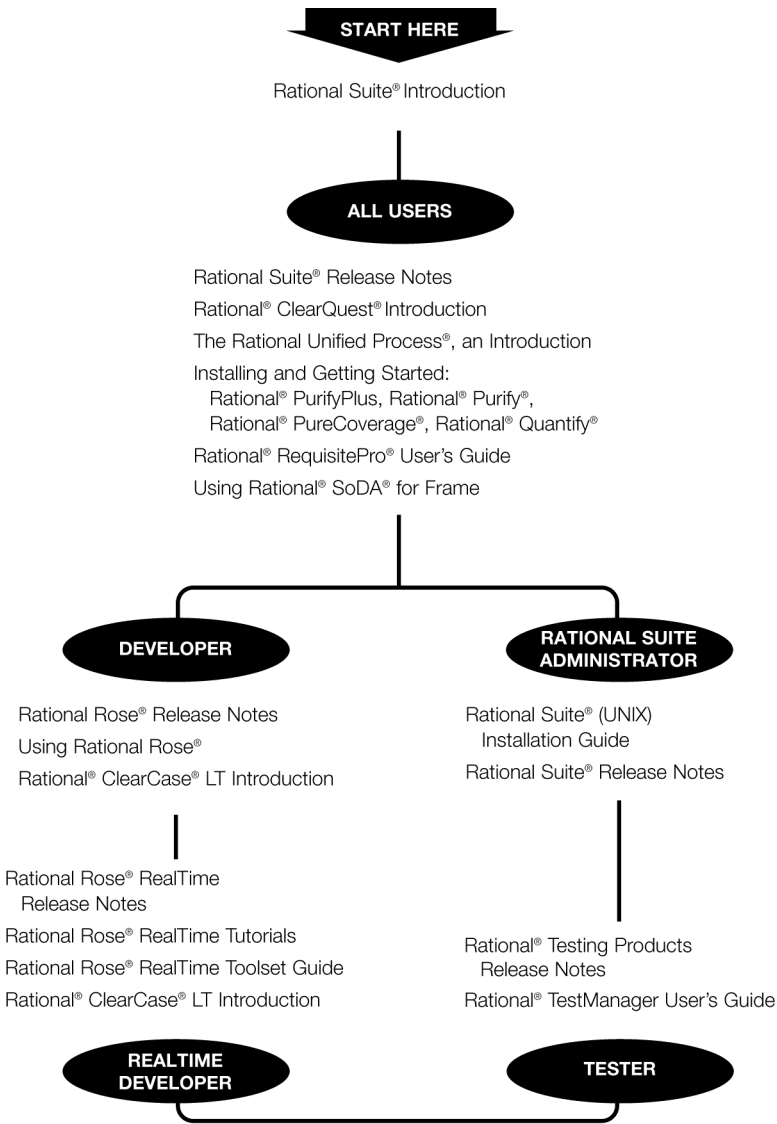
# Rational Suite Documentation Roadmap - Windows

---



# Rational Suite Documentation Roadmap - UNIX

---



## Contacting Rational Technical Support

---

If you have questions about installing, using, or maintaining this product, contact Rational Technical Support as follows:

Your Location	Telephone	Facsimile	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4546-201 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

**Note:** When you contact Rational Technical Support, please be prepared to supply the following information:

- Your name, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your case ID number (if you are following up on a previously reported problem)

Think about your last software project. Was it delivered on time? Was it released within its budget? Was communication among team members clear and timely? Did your team maintain consistency throughout the project as it defined requirements, developed designs, and wrote code? Was your development process repeatable? Did your software meet requirements, satisfy users, and perform reliably?

Many project teams experience problems in these areas. In fact, many software projects finish late (or not at all), and the results often don't match the requirements. Teams often uncover serious design flaws late in the development process. Defects are often found after the software ships, instead of during development.

How can you make your next project more successful?

## **Principles of Software Development**

---

Rational Software Corporation, the software development company, helps organizations overcome many software development issues by accelerating time to market while improving quality (see Figure 1). The Rational solution helps organizations develop software through a combination of:

- Software engineering best practices
- Integrated tools that automate these best practices
- Professional services that accelerate the adoption and implementation of these best practices and tools.

**Figure 1 Using Best Practices and Tools to Develop Better Software**



Rational helps you increase your productivity and effectiveness by focusing on the following software development best practices, as shown in Figure 1.

**Develop software iteratively.** Iterative development means analyzing, designing, and implementing incremental subsets of the system over the project lifecycle. The project team plans, develops, and tests one subset of system functionality per iteration. The team develops the next increment, integrates it with the first iteration, and so on. Each iteration results in either an internal or external release and moves you closer to the goal of delivering a product that meets its requirements.

Developing iteratively helps make your project more predictable, lets you collect feedback early, helps you identify and eliminate risks early in the project, and lets you test continuously throughout the project lifecycle.

**Manage requirements.** A requirement is one criterion for your project's success. Your project requirements answer questions such as "What do customers want?" and "What new features must we absolutely ship in the next version?" Most software development teams work with requirements. On smaller, less formal projects, requirements might be kept in text files or e-mail messages. Other projects may use more formal ways of recording and maintaining requirements.

By managing requirements, you can understand how changing requirements affect your project. You can effectively communicate requirements to all team members and to stakeholders. Effective requirements management helps your organization ensure that its products meet its stated goals.

**Use component-based architectures.** Software architecture is the fundamental framework on which you construct a software project. When you define an architecture, you design the structural elements of a system and their behavior, and you decide how these elements fit into progressively larger subsystems.

A component is a nontrivial, independent, and replaceable part of a system that combines data and functions to fulfill a clear purpose. You can build components from scratch, reuse components you previously built, or even purchase components from other companies. Designing a component-based architecture makes your product more reliable, maintainable, and extensible.

**Model software visually.** Visual modeling helps you manage software design complexity. At its simplest level, visual modeling means creating a graphical blueprint of your system's architecture. From this visual, you can quickly detect problems such as inconsistencies and lack of modularity.

Visual models also improve communication across your entire team. They help you detect inconsistencies between requirements, designs, and implementations. They also help you evaluate your system's architecture, ensuring sound design. Rational recommends using the *Unified Modeling Language (UML)*, the industry-standard language for visualizing and documenting software systems, to create visual models. UML provides a powerful and unambiguous communication mechanism for your whole team.

**Continuously verify quality.** Verifying software quality means testing what has been built against defined requirements. Testing includes verifying that the system delivers required functionality and verifying reliability and its ability to perform under load.

An important benefit of iterative development is that you can begin testing early in the development process. Testing every iteration lets you discover problems early and to expose inconsistencies between requirements, designs, and implementations.

**Manage change.** It is important to manage change in a trackable, repeatable, and predictable way. Change management includes facilitating parallel development, tracking and handling enhancement and change requests, defining repeatable development processes, and reliably reproducing software *builds*.

Managing changes to your project facilitates clear communication. It helps you propagate change throughout your organization, define and repeat development processes, and control risk.

## Rational Suite Can Help

---

To put these software development principles to work, Rational Software offers Rational Suite, a family of market-leading software development tools supported by the Rational Unified Process. These tools help you throughout the project lifecycle.

Rational Suite packages the tools and the process into several editions, each of which is customized for specific practitioners on your development team, including analysts, developers, and testers. Alone, these tools have helped organizations around the world successfully create software. Integrated into Rational Suite, they:

- **Unify your team** by enhancing communication and providing common tools.
- **Optimize individual productivity** with market-leading development tools packaged in Suite editions that are customized for the major roles on your team.
- **Simplify adoption** by providing a comprehensive set of integrated tools that deliver simplified installation, licensing, and user support plans.

## What's in Rational Suite?

---

Rational Suite editions are sets of tools customized for every member of your team. The following sections describe each Suite edition and the tools they contain.

### Optimize Your Team's Productivity

Rational Suite editions are optimized to enhance the productivity of each member of your team, and of your team as a whole.

**Rational Suite Team Unifying Platform** is a Suite that is included in every Suite edition. The Team Unifying Platform is designed for project members who need access to common project artifacts, but do not need any of the optimized, role-specific tools found in the other Suite editions. Project and program managers, project administrators, and development managers use the tools in this Suite edition. Although Rational Suite Team Unifying Platform is available for Windows only, most of the tools included in the Team Unifying Platform are also included in Rational Suite editions for UNIX.

**Rational Suite AnalystStudio** is customized for team members who gather and manage project requirements. Rational Suite AnalystStudio is available for Windows only.



**Rational Suite DevelopmentStudio** is customized for software architects, designers, and developers. These team members use DevelopmentStudio to design, evaluate, implement, and test software architecture and applications. Rational Suite DevelopmentStudio is available for Windows and UNIX.

**Rational Suite DevelopmentStudio—RealTime** is customized for software architects, designers, and developers of real-time embedded software. This Suite provides an integrated set of development and testing tools to optimize the development of complex software for devices such as cell phones and pagers, and infrastructure software for the routers and hubs that connect and power the Internet. Rational Suite DevelopmentStudio—RealTime is available for Windows and UNIX.

**Rational Suite TestStudio** is designed for team members who are responsible for software quality assurance, functional testing, performance testing, and load testing. Rational Suite TestStudio is available for Windows only.

**Rational Suite ContentStudio** is optimized for Web application development teams. This Suite provides an integrated set of tools to unify code and content management for Web applications. Rational Suite ContentStudio is available for Windows only.

**Rational Suite Enterprise** includes all of the tools from Rational Suite AnalystStudio, DevelopmentStudio, and TestStudio, providing a comprehensive tool set for the entire team. Rational Suite Enterprise is available for Windows only.

## Tools That Unify Your Team

### Rational Suite Team Unifying Platform

**Rational Suite Team Unifying Platform** unifies all members of a software development team to maximize productivity and quality. It provides best practices and integrated tools for managing change, building quality, and communicating results from requirements to release.

The Team Unifying Platform Suite edition contains the following tools:

**Rational Unified Process (RUP).** An online collection of software best practices that guide your team through the software development process. RUP provides guidelines, templates, and Tool Mentors (instructions for applying the guidelines to specific Rational tools) for each phase of the development lifecycle. The Rational Unified Process is available for Windows and UNIX.

**Rational RequisitePro.** Helps you organize, prioritize, track, and control changing project requirements. RequisitePro comes with the **RequisiteWeb** interface which lets users to access, create, and manage requirements from a Web browser. RequisitePro is available for Windows only. RequisiteWeb is available for Windows and UNIX.

**Rational ClearQuest.** Manages change activity associated with software development, including enhancement requests, defect reports, and documentation modifications. The **ClearQuest Web** interface lets users to perform all major ClearQuest operations, such as submitting records, finding records, creating or editing queries and reports, and creating shortcuts, from a Web browser. **ClearQuest MultiSite** lets you to share information across a geographically distributed team. ClearQuest is available for Windows and UNIX client workstations when a Windows workstation is configured as administrator for the ClearQuest repository; the ClearQuest Web interface is available for Windows and UNIX.

**Rational ClearCase LT.** Provides software configuration management and a built-in process to track changes to all software project assets, including requirements, visual models, and code. It also provides a Web interface that lets users perform all major ClearCase LT operations. Rational ClearCase LT supports *Unified Change Management*, the Rational best-practices-process for managing change and controlling workflow. ClearCase LT is available for Windows and UNIX.

**Rational SoDA.** Automatically generates project documents by extracting information from files you produce during project development, including source code and files produced by Rational tools. SoDA uses templates, either predefined or ones that you customize, to format the information. SoDA is integrated with Microsoft Word for Windows and Adobe FrameMaker for UNIX to support easy customizing of templates.

**Rational TestManager.** Helps you create real-world functional and multiuser tests to determine the performance and reliability of Web, multitier, and database applications. TestManager tracks the number of tests that have been planned, scripted, and run; which requirements have been covered; and the number of tests that have passed and failed. TestManager gives your team the information it needs to objectively assess project status and create reports to communicate these findings to project stakeholders. TestManager is available for Windows, and UNIX Suite editions.

**Rational ProjectConsole.** Helps you track project metrics by automatically generating charts and gauges from data produced during software development. ProjectConsole is integrated with Microsoft Project so that you can create a centralized project plan. ProjectConsole helps you organize project artifacts on a central Web site so all team members can view them. ProjectConsole is available for Windows only.

## Accelerate Project Delivery with Professional Services

---

Rational Software offers the **Rational Developer Network**, an online community, providing useful information as well as an exchange of ideas and best practices for software development teams. The network delivers content, training, and artifacts tailored exclusively for the worldwide community of development professionals who use Rational tools and best practices.

Content on the site is edited for quality and relevance, and organized into technology and process containers for ease of use. Web-based training helps individual team members expand their professional skills and stay ahead of the technology curve. The Rational Developer Network is available exclusively to Rational Suite customers.

Rational Software also offers a complete range of professional services to support Rational Suite, including consulting services, customer support, and training services. The goal of these services is to help software development teams consistently produce quality software on time and within budget.

See *Contacting Rational Professional Services* on page 57.

## Simplify the Solution for Your Team's Development Needs

---

Rational Suite offers one comprehensive solution for your team's development needs:

- One installation program that lets your team install and setup the integration of the tools at once, or one tool at a time.
- One integrated tool set that is tested and updated so you don't have to integrate the tools yourself.
- One location for Help, Rational Suite documentation, and Rational Customer Support.

# Rational Suite: Summary

This table shows which tools are included with each edition of Rational Suite.

		Project Leaders/Managers	Analysts	Developers				Testers	All Roles	Web Teams
		Team Unifying Platform	Analyst Studio	DevelopmentStudio		DevelopmentStudio RealTime Edition		TestStudio	Enterprise	Content Studio
		Windows	Windows	Windows	UNIX	Windows	UNIX	Windows	Windows	Windows
Tool										
Team Unifying Platform	Rational Unified Process	•	•	•	•	•	•	•	•	•
	Rational RequisitePro	•	•	•	•WEB	•	•WEB	•	•	•
	Rational ClearQuest	•	•	•	•WEB	•	•WEB	•	•	•
	Rational SoDA	•SW	•SW	•SW	•SF	•SW	•SF	•SW	•SW	•SW
	Rational ClearCase LT	•	•	•	•	•	•	•	•	•
	Rational TestManager	•	•	•	•	•	•	•	•	•
	Rational ProjectConsole	•	•	•		•		•	•	•
	Rational Rose		•M	•E	•U	•RT	•RT/U		•E	
Rational PureCoverage			•	•	•	•	•	•		
Rational Purify			•	•	•	•	•	•		
Rational Quantify			•	•	•	•	•	•		
Rational Robot							•	•		
Rational TestFactory							•	•		
Rational Process Workbench								•		
Rational NetDeploy									•	
Rational SiteLoad									•	
M = Professional Data Modeler Edition E = Enterprise Edition U = UNIX Edition RT = RealTime Edition SW = Rational SoDA for Word SF = Rational SoDA for FrameMaker WEB = Can also be accessed through the tool's Web interface										

## Incorporate Best Practices into Your Team's Process

Sustained delivery of quality software requires cohesive teamwork and a common understanding of development tasks. That's why the implementation of a predictable, repeatable process based on best practices is crucial to your success. The Rational Unified Process, more commonly referred to as RUP, serves as a personal and team-centered guide for controlled, iterative software development. Many organizations worldwide have successfully used it for both small-scale and large-scale development efforts.

RUP is implemented as an online guide and knowledge base which you view with a Web browser. It provides links between process guidelines and the tools contained in Rational Suite.

You can customize this process to meet your exact needs and provide every member of your team with a customized roadmap to software development success.

### The Rational Unified Process at a Glance

The RUP overview presented in Figure 2 represents a complete software development project lifecycle.

**Figure 2 The Rational Unified Process Overview**

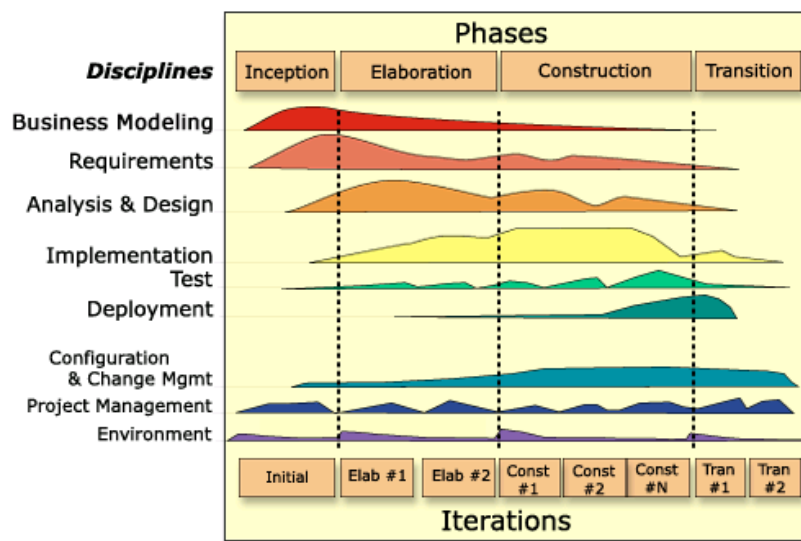


Figure 2 is based on empirical data collected by Rational Software, showing that software development is best organized into *phases*, each of which is implemented in a series of *iterations*. Throughout each phase, project team members from each of the software development disciplines (analysts, developers, testers, project leaders) perform activities in one or more *workflows*. The diagram shows how emphasis on different workflows varies with each iteration.

Each iteration focuses on one of these development phases:

- **Inception**—Define scope of project.
- **Elaboration**—Plan project, specify features, baseline architecture.
- **Construction**—Build and test product.
- **Transition**—Deliver product to the end-user community.

Notice, for example, that most of the work associated with requirements happens early in the development cycle, but continues throughout a project. Testing, however, can start early in the project, but typically becomes most intense at the end of construction.

The steps in an iterative process correlate with the responsibilities of specific team roles: analyst, developer, tester, and manager or project leader. Although these *activities* are presented as linear steps, each role performs these activities as early as possible and throughout each iteration. Requirements analysis, for example, is emphasized at the start of an iteration; however, analysts track requirements throughout each iteration and throughout the project lifecycle.

## Adopting the Rational Unified Process

RUP is a customizable framework that provides development teams with a common set of software development best practices. RUP can easily be adapted to the specific needs of your projects and your team. You can incorporate your own company's best practices, and you can create variants that best describe your development environment.

We recommend that you follow your company's implementation of RUP to support your development efforts. If your company has decided to use Rational Suite without adopting any of the Rational Unified Process, your projects can still be successful. (You can also use RUP with projects that do not use Rational Suite or its component tools.)

## The Rational Unified Process and Rational Suite

Even if you do not follow RUP, you can use it as a source of information about software engineering. For example, it contains topics to help you better understand Unified Modeling Language (UML) concepts.

RUP provides the following links between process guidelines and the tools in Rational Suite:

- **Tool Mentors** provide step-by-step instructions for performing activities using Rational tools.
- **Extended Help** is available in all Rational tools for Windows and provides paths to relevant topics in the Process, including Tool Mentors. You can add your own content to Extended Help to customize it to your team's work. Extended Help is available for Windows only.

## What's Next?

---

The remainder of this book:

- Shows how analysts, developers, testers, and project leaders benefit by using Rational Suite.
- Shows how Rational Suite tools fit into an effective iterative process by focusing on requirements management, analysis and design, implementation, testing, configuration and change management, project management.
- Directs you to more information about Rational products, services, and the Rational Unified Process.





# Defining the Right System: The Analyst

# 2

Are you solving your users' changing needs?

Did “feature creep” delay your last release?

Is your project over budget? Behind schedule?

Are your users dissatisfied with or disappointed in your product?

## Understand the Problem Before You Invest

---

As an analyst, your first job is to identify the problem that your organization will address. You represent *stakeholders* by capturing, organizing, and managing their expectations and requests for the system.

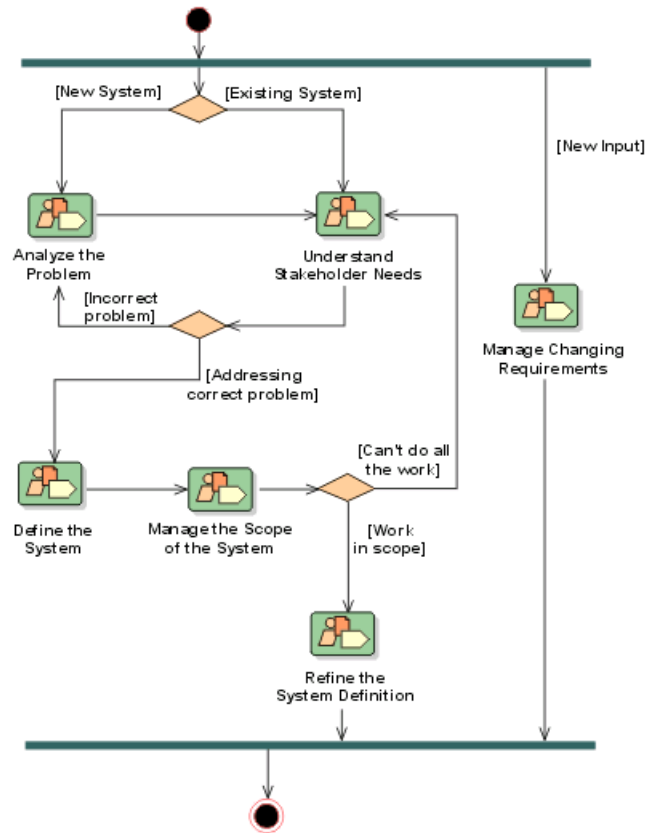
To clearly define the right system, you interpret stakeholder requests and articulate this information as requirements. These requirements:

- Define the boundaries of the system.
- Provide a basis for planning the technical contents of each iteration.
- Provide a basis for estimating cost and time to develop the system.
- Define a user interface for the system, focusing on the needs and goals of the users.

Throughout this effort, you must work with all stakeholders to clarify the problem space and solution requirements. Even as the development objectives and requirements change during the project, it is your job to maintain communication with stakeholders and a shared understanding of the requirements. Defining requirements lets you establish and maintain agreement with the customers and other stakeholders on what the system should do, and provides developers with a better understanding of the system requirements.

Figure 3 shows a typical Requirements workflow. Each detail of the workflow represents an activity to perform to ensure effective requirements management.

**Figure 3 Requirements Workflow**



The Rational Unified Process (RUP) uses workflow diagrams as a high-level map for a sequence of related *activities* that team members perform. The arrows between activities represent the typical, though not required, flow of work between activities.

## Iterative Development and the Challenge of Managing Requirements

During development, requirements change and evolve. For example:

- Your competitors release new or updated versions of their products. To stay competitive, you add requirements for new features to your project.
- The customer was unclear or undecided about some requirements at first, so you might add, remove, or clarify requirements later on.

- Technology advances after you start product development. You need to determine if you can incorporate this technology into your project without jeopardizing the schedule or other deliverables.
- You discover that a requirement is too expensive to implement or that your team cannot implement it in the time allotted, and you decide to drop or postpone the requirement.

## Capture and Manage Requirements with Rational Suite AnalystStudio

---

Rational Suite AnalystStudio is the complete solution for development professionals who gather and manage project requirements. This Rational Suite edition increases customer satisfaction by helping your team build a system that meets your customers' needs.

### Defining Requirements for Your Team

**Rational RequisitePro** enhances team communication by helping analysts capture, manage, and articulate requirements in a form accessible to all stakeholders. RequisitePro combines the power of a relational database with the freedom of Microsoft Word:

- As an analyst, you use Microsoft Word to document project requirements. Throughout the software development lifecycle, you use requirements to communicate what to build and test. The requirements form the foundation of the system definition by defining the product vision and describing the system's features, functionality, and attributes.
- You then store and track this information in the RequisitePro database to optimize requirements management. The database lets you assign attributes to requirements for organization, prioritization, and change history. These attributes can be traced to related requirements so that you can quickly assess the impact of any changes on requirements as development evolves.

Because every team member must share a common understanding of project goals and objectives, you have to keep your team up to date on requirements activities. RequisitePro makes this easy by providing **Rational RequisiteWeb**. This Web-based tool lets UNIX users and other team members who do not have RequisitePro on their desktops to create, review, and update requirements.

## Managing Change for Your Project

**Rational ClearCase LT** is a configuration management solution for small project teams. This tool helps your team manage changing *artifacts*, such as Web content, code, visual models, and test assets as the system evolves. As an analyst, you can use ClearCase LT to manage changes to your project requirements. The tool lets you associate requirements with releases and other project assets.

You can also use ClearCase LT to archive requirements. It tracks changes to all your project files, allowing team members to work in parallel and continuously integrate their changes to the project baseline.

**Rational ClearQuest** encourages collaboration by tracking new feature, enhancement, or change requests from team members and other stakeholders. The ClearQuest Web interface ensures that UNIX users and other team members who do not have ClearQuest on their desktops can participate in this collaboration.

With Rational ClearQuest, your team and other stakeholders can evaluate requests, determine their impact on the system, and when applicable, validate the changes. To establish how change requests fit into the structure of features and main requirements, you can link requests to an existing or new project requirement in Rational RequisitePro.

**ClearQuest MultiSite** lets geographically distributed teams replicate a centralized database at each remote site and then synchronize the changes between sites.

## Helping Your Team Communicate Visually

**Rational Rose** (Professional Data Modeler Edition) helps you visualize, specify, construct, and document the structure and behavior of your system's architecture. With Rose, you can provide a visual overview of the system using the Unified Modeling Language (UML), the industry-standard language for visualizing and documenting software systems.

Using visual models helps manage system complexity because you can see the “big picture.” Rational Rose unifies the team by helping you create such models so that all stakeholders gain perspective and share a common understanding of the project goals, path, and expected deliverables. Modeling with Rose is also an effective way to continuously communicate change and the impact of change throughout the development lifecycle.

All team members can easily share and revise Rose models because they are written in the UML—an easily understood, industry-standard language for designing software. For example, analysts use Rose to describe a system at a high level and developers continue this work by using Rose to design the system in more detail. Therefore, your application, system, and data are managed by one tool, Rational Rose, and with one language, UML.

## Measuring Progress and Providing Project Reports

**Rational ProjectConsole** extracts information from data produced during software development by tools like RequisitePro and ClearQuest. ProjectConsole automatically generates pre-formatted or customized charts and graphs depicting the collected metrics. These metrics let you share project status information with members of your team. ProjectConsole also lets you analyze data in a single view that reflects data collected from multiple sources. For example, you can use ProjectConsole to learn how many requirements and defects have been implemented and tested.

**Rational SoDA** generates up-to-date project reports of data extracted from one or more tools in Rational Suite. SoDA can work with one Rational tool, such as RequisitePro, or combine information from more than one tool, such as Rational RequisitePro and ClearQuest. These reports provide a way for your team to communicate more efficiently and consistently. SoDA offers reporting features that provide templates in either Microsoft Word for Windows or Adobe FrameMaker on UNIX. You can easily customize predefined templates or create new ones.

## Capitalizing on the Power of Use Cases to Test All Dimensions of Quality

**Rational TestManager** lets testers manage test planning, design, development, execution, and analysis and share these assets with team members throughout the development lifecycle. Use cases provide a consistent representation of the problem being solved or the behavior of the system being developed, throughout the requirements, analysis, design, and testing activities. As a result, use cases are the foundation from which developers and testers develop test plans and build test scripts.

Rational TestManager lets you build test assets from this foundation of use cases. In addition, all test assets, including test plans and *test cases*, manual and automated tests, and test results are stored in a central location. These artifacts are stored in a single database, using a single interface, and are accessible to all team members.

## Summary

---

- Rational Suite AnalystStudio provides an integrated set of tools to help you effectively capture, manage, and communicate requirements to your project group.
- Rational RequisitePro, the primary tool in AnalystStudio, helps you interpret requests from stakeholders and define the right system to solve the user's problem. The entire team uses these requirements to understand the project vision and as a foundation for individual work.
- Effectively managing requirements and communicating a shared understanding of the project as it evolves helps your team avoid common development pitfalls and deliver products on time and within budget.

# Managing Complexity: The Developer

# 3

- Can you build the system right the first time?
- Do you discover design flaws too late to fix them?
- Do project modules integrate correctly?
- Can you maintain the integrity of the system's architecture?
- Can you easily extend the system's design?
- Can you reuse project components?
- Does your team know what they are building?
- How does your team communicate changes to each other?

## Managing a Complex Process

---

Software developers define *how* the system works. As a developer, you define the system's components and its interfaces, then decompose these components into successively smaller components. You must design a flexible, scalable system that you can quickly and easily modify as project requirements change. As the developer, you create, modify, and manage code for the system based on this model.

## Analysis and Design: The Next Steps in Iterative Development

---

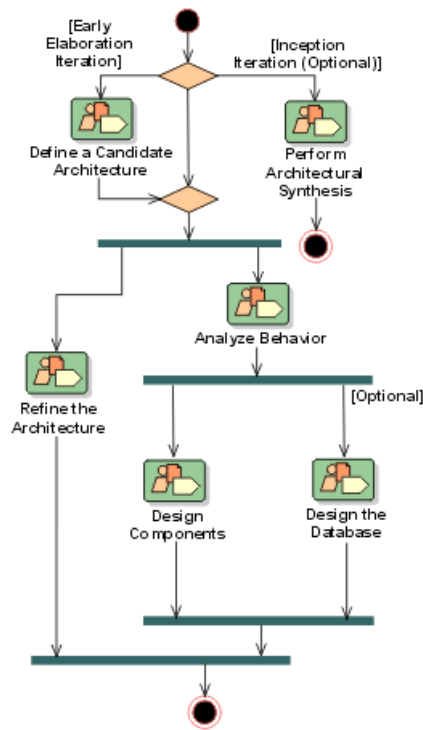
As the developer, you review project requirements and then make decisions about the structural elements and interfaces of the system. As requirements change and new problems arise in each iteration, you must refine the system architecture.

In the system design, you lay out the components of the architecture. These individual components are independent, replaceable parts of the system that have clearly-defined functions and interfaces. When you work with component-based architectures, your team can easily create new components. Team members can also reuse, or even customize, existing components from previous projects or commercially available sources.

As a developer, you produce executable code that can be evaluated against the project requirements and the design during each iteration. As a responsible developer, you test your code before you release it to other developers or to your testing group. This approach lets you discover and respond to problems early enough to minimize their impact on the project.

One purpose of analysis and design is to transform the requirements into a system design. As described in the Rational Unified Process (RUP), the workflow in Figure 4 shows how developers verify that a design model fulfills system requirements, and that it serves as a good basis for its implementation.

**Figure 4 Analysis and Design Workflow**





# Creating Resilient Component-Based Architectures with Rational Suite DevelopmentStudio

---

**Rational Suite DevelopmentStudio (Windows and UNIX)** is the complete solution for software architects, designers, and developers. Using tools that create real-world, multiuser tests of system performance, this Suite helps your team design and build the right product the first time. You can even use these tools to test performance as soon as your system architecture is designed.

**Rational Suite DevelopmentStudio—RealTime Edition (Windows and UNIX)** is further customized for practitioners who focus on real-time and embedded development. For more information, please see *Creating Component-Based Executable Architectures with Rational Suite DevelopmentStudio—RealTime Edition* on page 38.

## Communicating Visually with Models of Your System

**Rational Rose (Enterprise Edition for Windows, UNIX Edition for UNIX)** helps you visualize, specify, construct, and document the structure and behavior of your system's architecture. With Rose, you can provide a visual overview of the system using the Unified Modeling Language (UML), the industry-standard language for visualizing and documenting software systems. Rose unifies the team by helping you create high-quality architecture models that all team members can share, test, and revise. Since Rose uses the industry-standard UML, it ensures that all members have the same understanding of the project. During design and code reviews, team members use project models to assess the ramifications of changes they want to make to the code.

## Accelerating Code Implementation

You can accelerate coding by generating code frameworks from models developed in Rational Rose. This process is called *forward engineering*. Rose supports many languages, including Visual Basic, Visual C++, ANSI C++, Java, and IDEs including IBM Visual Age for Java, Visual Studio, HP Workbench, and Sun Workshop.

## Keeping Code and Models Consistent

After you modify your code, you can use Rose to bring code changes into your model. This process, *reverse engineering*, ensures that your model and code remain consistent throughout the project.

Keeping your code and model consistent helps you clearly see the impact of changes on the system architecture throughout development. It is important to assess these changes early because they may violate project standards and requirements, or impact architectural decisions.

## Evaluating Changing Requirements

**Rational RequisitePro** provides up-to-date requirements data in a form accessible to all stakeholders. This access is provided with **RequisiteWeb**, a Web interface that lets all team members create, review, and update requirements.

When you, the developer, see additions or changes to requirements in RequisitePro, you can incorporate these changes into the project Rose models. As you change these models, you gain an understanding of the impact these changes have on the system. You can communicate these changes and impacts to team members, successfully unifying analysts, developers, testers, project leaders, and other stakeholders.

## Validating Changes to the System

**Rational ClearQuest** tracks features, enhancements, or change requests from team members and other stakeholders in a form accessible to everyone. The ClearQuest Web interface ensures that UNIX users and other team members who do not have ClearQuest on their desktops can create and review project data, and provide feedback. You can use Rose models to visualize the effects of these requests on the system architecture.

**Note:** ClearQuest is available for UNIX client workstations only when a Windows workstation is configured as administrator for the ClearQuest repository.

## Managing Changes to the System

**Rational ClearCase LT** helps you control changes to source code and other project items, such as Rational RequisitePro databases and Rational Rose model files. ClearCase LT tracks changes to every file and directory, maintaining histories of source code, binaries, executables, documentation, test suites, libraries, and user-defined objects.

You can use ClearCase LT from your favorite development environment, for example, Microsoft Visual Studio, IBM Visual Age for Java, as well as Microsoft Word, FrontPage, Visual InterDev, and PowerBuilder. By organizing an effective development structure that supports parallel development, ClearCase LT helps software teams accelerate build and release cycles.

Optionally, you can use Unified Change Management (UCM), the built-in Rational process that helps you use ClearCase LT and ClearQuest together to manage change in your software development environment. UCM defines how to manage evolving requirements, design models, documentation, components, test cases, and source code. UCM links the activities used to plan and track software development with the artifacts that are changing throughout development.

UCM raises the level of abstraction when working with software configuration management and change management tools, such as ClearQuest and ClearCase LT. For example, a project leader uses ClearQuest to identify which activities to assign to each developer. UCM maintains a *change set* — a list of artifact changed for — each activity.

After you finish working on an activity, you *deliver* the work to a project-wide, shared workspace, the *integration stream*. Within this stream, you can build and test the latest versions of the project's shared elements. After building and testing the delivered activities from all developers, the project leader may decide to create a new *baseline*. A baseline represents a stable configuration of a project's components, and when *promoted* by a project leader, becomes the basis for future work.

**Note:** UCM is available for use with ClearCase LT on Windows only.

## Keeping the Team Up to Date

**Rational ProjectConsole** provides Web access to the project's current artifacts, metrics, and best practices. ProjectConsole automatically collects metrics at regular, user-specified, intervals, from Rational tools and Microsoft Project. These metrics are graphically displayed in personalized views on the project Web site. This reporting mechanism provides accurate, objective, and up-to-date information to team members about the status and trends.

**Rational SoDA** (for Microsoft Word or for Adobe FrameMaker) lets you generate up-to-date project reports for the entire team by extracting data from one or more tools. SoDA can work with one Rational tool or combine information from more than one tool. Its reporting features provide templates in either Microsoft Word on Windows, or Adobe FrameMaker on UNIX. You can easily customize these templates or create new ones.

## Testing Code Early and Often

As a developer, you test your code as soon as you implement it. Rational Suite provides testing tools to use as soon as you have a working program, allowing you to test all dimensions of quality with automated debugging, performance testing, and verification of code coverage.

**Rational QualityArchitect** is a feature of **Rational Rose (Enterprise Edition)** that automates the mechanical aspects of test code creation by generating test code from visual models. This lets developers automatically generate component tests and build stubs and drivers before an application is complete. This feature helps to reduce project risk because your team can test early and often, determining how a potential system

architecture meets functional and performance requirements before developing the design further. Enterprise JavaBeans, COM, COM+, and DCOM models are supported in this feature.

**Note:** Rational QualityArchitect is not included in Rational Suite DevelopmentStudio for UNIX.

**Rational Purify** checks every active C++ and Java (Windows only) component in your program for run-time errors and memory leaks, the most difficult errors to find. They are the most important to correct because they often remain undetected until triggered by some random event. A program can appear to work correctly for a long time before these types of errors are discovered.

**Rational PureCoverage** provides a report of each line in your code that has been run. This information lets you determine if your tests have actually run the lines of code that were intended to be tested.

**Rational Quantify** detects performance bottlenecks, which are places where the code is running inefficiently. It pinpoints where the application is spending its time, and helps you discover why a specific function is particularly slow. Quantify helps you improve system performance so that you can deliver efficient software.

## Tracking Test Results

As a developer, it is critical to communicate the status of projects and activities with your team members and other project stakeholders. Rational Suite provides tools to help you do this effectively throughout the development lifecycle. You can track and report change requests, test your models and code more often and more completely, keep abreast of project status, and communicate your results to the rest of your team.

**Rational TestManager** helps you keep track of the number of tests that have been planned, implemented, and run. It also helps you track which requirements or Rose model elements have been covered in testing, and the number of tests that have passed and failed. Team members can use TestManager to evaluate how well they are meeting project requirements from early in the development lifecycle, and communicate these findings to project stakeholders.

Test Manager is available for use in both Windows and UNIX development environments. The UNIX environment requires that a Windows server hosts TestManager, and that the UNIX computers run the test *Agents* to perform the testing and data collection activities.

In a UNIX environment, users typically configure their computers and use TestManager as follows:

- 1** An administrator installs TestManager on a single Windows computer (known as the *Local computer*), and test agent software, either on its own or as part of the Rational Suite DevelopmentStudio for UNIX, on one or more UNIX workstations (known as *Agent computers*).
- 2** Testers coordinate all testing activities from the Local computer, where TestManager is installed.
- 3** Testers use Agent computers to:
  - a** Add workload to the local computer, if they are running a *performance test*.
  - b** Run test scripts on more than one computer. To save time running a functional test, run the test scripts on the next available Agent computer instead of running them all on the Local computer.
  - c** Run test configurations. For example, to test different hardware and software configurations, run test scripts on specific Agent computers set up with these configurations.

**Rational ClearQuest** tracks software and related defects. Rational testing tools are integrated with ClearQuest to simplify the process of entering defect information. ClearQuest provides the ClearQuest Web interface, enabling all team members to create, review, and update defects. ClearQuest tracks a defect's history and provides a description and other details. **ClearQuest MultiSite** lets a team that is widely distributed geographically share project information. UNIX developers will use the tool's Web interface.

# Creating Component-Based Executable Architectures with Rational Suite DevelopmentStudio—RealTime Edition

---

Rational Suite DevelopmentStudio—RealTime Edition is customized for developers who focus on real-time and embedded development. This Suite edition contains all the tools in Rational Suite DevelopmentStudio, replacing Rational Rose with Rational Rose RealTime.

## Building Complex, Real-Time Systems

Rational Rose RealTime is a comprehensive visual development environment that delivers a powerful combination of notation, processes, and tools to meet the challenges of real-time development.

Using Rose RealTime, you can:

- Create executable models, allowing you to compile and observe simulations of your UML designs on the host or target platform. The result is that you can refine your design early and you can continually verify quality.
- Generate complete, deployable, executables in C, C++, or Java directly from UML design models targeted to real-time operating systems. This eliminates the need for manual translation and avoids costly design interpretation errors.

## Working the Way Real-Time Systems Operate

Rational Rose RealTime is optimized for building complex, real-time systems. Rose RealTime:

- Uses UML and a UML profile specialized for real-time systems to represent all structural and behavioral detail of real-time and embedded systems.
- Allows selective and complete management of concurrency.
- Supports monitoring, execution, and debugging of models on the host or the target platform.
- Generates complete C, C++, or Java applications from UML models.
- Supports multiple real-time operating systems out of the box.

## Summary

---

- Rational Suite DevelopmentStudio offers an integrated tool set to optimize the design, coding, and unit testing of your software development project.
- Rational Suite DevelopmentStudio—RealTime offers an integrated tool set to assist with the definition, design, application generation, and testing of real-time, embedded development projects.
- Rational Suite DevelopmentStudio lets you easily communicate with your team by offering web publishing tools.
- Rational Rose, the primary tool in DevelopmentStudio, helps you design your system's architecture. Rose models help team members see the ramifications of any changes they want to make to the code. It is important to assess these changes early, as they may violate project standards or architectural decisions.
- Rational Rose improves your team's ability to manage software complexity. For example, Rational Rose provides round-trip engineering that lets you easily keep your models and your code synchronized throughout the development lifecycle.
- ClearCase LT helps developers manage changes to project artifacts and makes those changes available to the project leader for integration in the next baseline.
- Rational TestManager helps you track how many tests have been planned, implemented, and run. It also helps you track which requirements or Rose model elements have been covered, and the number of tests that have passed and failed.





# Deciding to Release: The Tester

# 4

Is your project behind schedule? How far?

Does your system scale to accommodate increasing load?

How many critical bugs were discovered after your last release?

How often do you have a complete picture of your software's readiness before you release?

## Making the Crucial Decision to Release

---

As a tester, you use all types of project artifacts to plan, design, and run tests. You ensure that software meets its requirements and is stable by testing the relationship between project assets. In each iteration, you communicate application defects and pinpoint performance problems so that your team can determine the software's readiness to be released.

## Subsystem and System Tests: The Last Step in Iterative Development

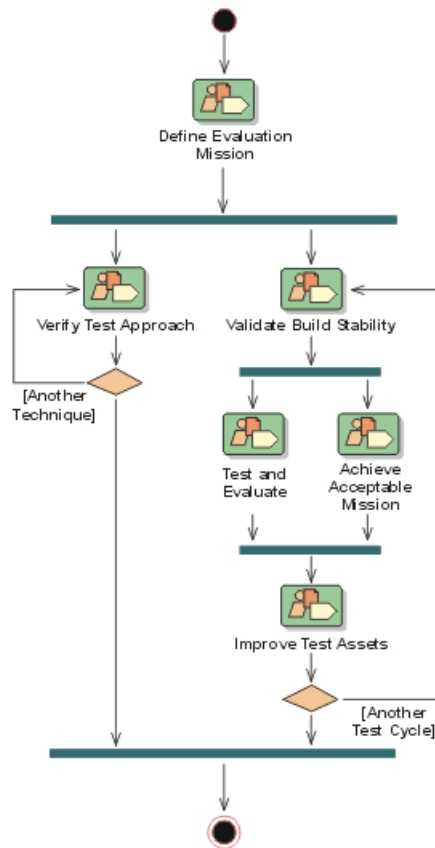
---

In iterative development, your team tests in planned increments. You thoroughly integrate and test an executable release within each iteration to test all dimensions of quality.

During the project lifecycle, you manage and track use cases, requirements, and tested code. Often, you retest code because of updated requirements or repaired defects. Your group also runs regression tests on new builds to detect whether new bugs have appeared where they did not exist in previous builds. During each iteration, your team analyzes the type and number of defects in each build, and decides which modules need to be tested again.

The workflow in Figure 5 shows a typical Test workflow as described in the Rational Unified Process (RUP). Each workflow detail represents the key skill needed to verify the proper integration of all components of the software, to verify that all requirements have been correctly implemented, and to ensure that defects are addressed before releasing the software.

**Figure 5 Test Workflow**



## Verifying Software Quality with Rational Suite TestStudio

---

Rational Suite TestStudio is a complete solution for team members who verify the reliability, functionality, and application performance of software. As the demands of developing e-commerce solutions increase, system performance becomes a critical dimension of software quality. Rational Suite TestStudio helps your team answer the crucial question, “Are you ready to release?”

### Unifying the Team by Keeping Members Informed

**Rational RequisitePro** provides current, accurate information about requirements. RequisitePro indicates to you, a tester, that there are new or revised requirements that need to be tested. You use the requirements and related artifacts to create your test plan and track testing progress. The RequisitePro Web interface, **RequisiteWeb**, lets team members and customers who do not have RequisitePro on their desktops create, review, and update requirements.

**Rational ProjectConsole** unifies the team by allowing all contributors Web access to current project artifacts, and metrics. This tool extracts information from data produced during software development and automatically generates charts and gauges, either predefined ones or ones that you customize. These metrics are graphically displayed in personalized views on your project Web site, providing accurate, objective, and up-to-date information about the project status and trends.

ProjectConsole also lets you analyze data in a single view that has been collected from more than one source. For example, you can use ProjectConsole to learn how many enhancement requests and requirements have been implemented and tested.

**Rational SoDA** extracts information from one or more tools and combines this information into reports about your project. Your team can evaluate test results against requirements data from RequisitePro.

SoDA offers reporting features that provide templates in either Microsoft Word (on Windows) or Adobe FrameMaker (on UNIX). You can easily customize these templates or create new ones.

## Making a Plan and Measuring Progress

**Rational TestManager** lets you use all types of project artifacts to plan, design, and run tests. Requirements, visual models, and source code are some of the *test inputs* that you can use to create a test plan, so that all aspects of your system can be tested, including product features, system architecture, and code. With TestManager, test inputs are used to create specific *test cases*. A test case describes a testable and verifiable behavior in a system. It can also describe the extent to which you will test an area of the application.

When the original artifact changes, you are automatically prompted to re-evaluate the related test asset. You can then decide to update the test artifact, if necessary. For example, TestManager can verify whether you have tested all your requirements, and whether the test cases based on these requirements pass the tests. This relationship between project assets lets you test for quality early in the development lifecycle, often beginning with product features, and continuing through implementation and release.

## Does Your Application Meet Requirements?

**Automated Functional Testing—Rational Robot** determines whether the system meets its requirements by testing how it responds to a user-driven scenario. With the intuitive interface of Robot, you can record a test and insert verification points to monitor expected behavior. You can also replay the test as often as you need.

After you've run the test, you can view the results and the complete details of any failures: what test was running, what type of failure occurred, where it occurred, and which verification point failed.

## Is Your Application Reliable?

**Automated Reliability Testing—Rational TestFactory** automatically generates tests that pinpoint severe defects: the places where the application crashes, hangs, or behaves in unexpected ways. It also generates test scripts that exercise the maximum amount of code using the least number of steps.

TestFactory stores the test scripts, results, and defect scripts in a *project* that it shares with Robot and other Rational testing tools. Your team can generate coverage and progress reports from test results in this project. Rational Robot can later rerun TestFactory scripts to ensure that all tests are repeatable.

Because TestFactory generates its own tests, you can start reliability testing early in the development process without having to budget more time to develop and run these tests yourself.

## Does Your Application Have Memory Leaks?

**Automated Reliability Testing—Rational Purify** checks every active C++ and Java component in your program for run-time errors and memory leaks, the most difficult errors to find. They are the most important to correct, though, because they often remain undetected until triggered by some random event. A program can seem to work correctly for a long time before these types of errors are discovered.

## Is Your Application Fast Enough?

**Automated Application Performance Testing—Rational Quantify** detects performance bottlenecks, highlighting places where the code is running inefficiently. It pinpoints where the application is spending its time, and why a specific function is particularly slow. Quantify helps you improve system performance so that you can deliver efficient software.

## Does Your System Perform Under Production Load?

**Automated System Performance Testing—Rational TestManager** lets you run multiuser performance tests for Web, multitier, and database applications. Using simple point-and-click operations, you can create usage scenarios that simulate conditions in the system while it is being run by thousands of users. As TestManager runs these tests, it collects data that helps your team measure and predict your project's system performance.

## Have You Tested Enough?

**Building Comprehensive Tests—Rational Robot and Rational PureCoverage** highlight untested areas of your system so that you can build a more comprehensive set of tests. This information increases the quality of your system and mitigates risk.

## Optimizing Defect Tracking

**Rational ClearQuest** tracks the defects that are found in your software project and provides a description, as well as other details, about the bugs. Rational testing tools are integrated with ClearQuest to simplify the process of entering defect information and to ensure that repaired defects are verified. The ClearQuest Web interface is a Web-based version of ClearQuest that lets all team members create, review, and update defects and change requests from any platform. The ClearQuest Web interface provides the same features available through the desktop client. Team members are able to access and create records, queries and reports on areas of interest specific through the Web module as they would through the client version.

## Summary

---

- Rational Suite TestStudio is the complete solution for team members who verify the reliability, functionality, and application performance of software.
- Rational TestManager helps you to plan, design, and execute tests, manage test assets, analyze results, and measure progress. TestManager also tracks the test coverage of requirements, models, and configurations. With this tool, you can create reports to communicate project status to project stakeholders.
- Performing tests early and often in the project lowers the cost of completing and maintaining software. It also greatly increases software quality and user satisfaction
- The integrated testing tools Rational Robot and Rational TestFactory promote testing as early as possible in the development process. With the help of these and other Rational testing tools, your project group can detect and fix many critical defects before releasing the product.

# Managing Change and Risk: The Project Leader and Web Teams

# 5

Have you successfully managed changing requirements?

Do you manage software changes or do they happen haphazardly?

How well is the team meeting your customers' needs?

Is the project behind schedule? Over budget?

## How Well Can You Plan Ahead?

---

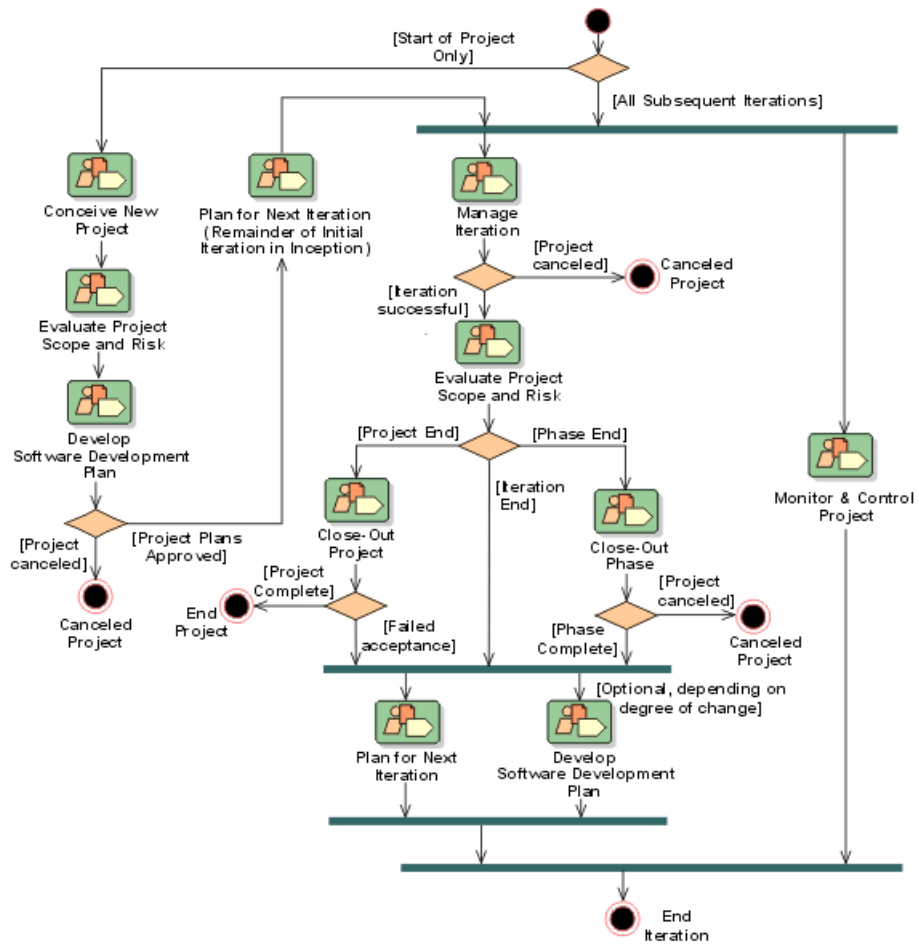
As a project leader, you identify and manage project risks, monitor your team's progress, and plan each iteration. Your ongoing responsibilities are to assign and schedule work, and to monitor the progress of the project.

Early in the development lifecycle, your team identifies, implements, and tests the most risky features and architectures. Monitoring project progress involves collecting and assessing the latest metrics or status reports from each team member. Throughout development, you analyze project data to determine how well the team is meeting its objectives. You also use this data to manage change and plan subsequent iterations.

In general, the art of project management involves balancing competing objectives, managing risk, and overcoming constraints to successfully deliver a product that meets the needs of all stakeholders.

The workflow in Figure 6 shows how project leaders plan an iterative project, its iterations, and monitor progress, as described in the Rational Unified Process (RUP).

**Figure 6 Project Management Workflow**





# Rational Suite: The Complete Solution for Iterative Development

---

Each edition of Rational Suite is optimized to support a major team role: analyst, developer, tester, and project leader. Each Suite edition also includes software that unifies teams with proven processes for managing change, ensuring quality, and improving communication from requirements to release.

This tool set is called the Team Unifying Platform, and it focuses on all product domains: requirements, modeling, testing, and change management. All Windows Suite editions include the Rational Team Unifying Platform which helps team members obtain the information they need to manage change and control risk associated with all phases of software development.

For the project leader who also represents another role typically found in software development organizations, Rational Suite Enterprise edition provides a comprehensive tool set that includes each Studio edition, as well as the Rational Suite Team Unifying Platform. Rational Suite Enterprise includes tools and processes to support each role in your team and each phase of your software development project.

When your team develops iteratively, you can more effectively manage risk and change. As your team implements more enhancements and features in a controlled, iterative manner, you can collect feedback about the system. This early feedback, which includes defect reports, lets the team change or adjust system architecture when it is easier and less expensive to do so.

## Implementing Best Practices

The Rational Suite family of products supports effective software engineering practices. The Suite integrated tools help project teams quickly develop high quality software in a repeatable and predictable manner. As the leader of a project team, you oversee and participate in the following software development best practices:

- **Develop software iteratively.** You assess the team's progress during each step of the iterative process: requirements analysis, design, coding and unit testing, subsystem and system testing. You request reports from the team to help you plan the next iteration.
- **Manage requirements.** You work with analysts to prioritize, refine, and update requirements. In early iterations, your team negotiates with other stakeholders about how to review and evaluate enhancements and change requests.

- **Use component-based architectures.** You, along with the architects and developers, plan which components of the system your team may build from scratch, buy from another company, or reuse from a previous project. These components form the fundamental framework for your software project.
- **Visually model software.** In each iteration, you work with architects and developers to create or update models, the blueprints, of the system architecture. The entire team uses visual models to gain a quick understanding of the system.
- **Continuously verify quality.** As early as possible in the development cycle, your team starts testing the project to identify run-time errors, memory leaks, and other defects. You examine the test results to evaluate how well the team is meeting project requirements. You and the testers determine which code the team should retest in the next iteration because of changed requirements or repaired defects.
- **Manage change.** It is important to manage changes in a trackable, repeatable, predictable manner. This includes facilitating parallel development, tracking and handling enhancement and change requests, defining development processes so that they are repeatable, and reliably reproducing software builds. Rational Suite provides your team with tools to accomplish these activities.

## Developing Complex Products Using Unified Change Management

Unified Change Management (UCM) is the Rational approach to managing change throughout the software development lifecycle, from requirements to release. UCM makes it easy to trace an activity from initial change request through final software release, all the while maintaining the integrity of the project assets. UCM helps project leaders protect their team activities and make them readily available for change as the project evolves. Successful Unified Change Management includes these critical and interdependent functions:

- **Change Request Management.** Focuses on capturing and managing requested project and system changes as contributed by external and internal project stakeholders.
- **Configuration Management.** Focuses on configuration control for sets of artifacts. Facilitates parallel development within and across project teams by providing private workspaces and powerful project integration capabilities.

**Note:** UCM is available for Windows only.

## Change Request Management

**Rational ClearQuest** tracks the defects that you find in your software project. ClearQuest can synthesize this information into easy-to-read charts and reports. You use these metrics to analyze defect trends. For example, is the team resolving fewer bugs now than it was a month ago? You can also use ClearQuest to assess the number of unassigned defects and the defect workload of each team member. You can then allocate unassigned defects appropriately.

## Configuration Management

**Rational ClearCase LT** lets your team organize and manage versions, releases, and, when necessary, parallel development of multiple products. ClearCase LT empowers software teams to accelerate development cycles, ensure accuracy of releases, and organize an effective distributed development process. For example, your project team can use ClearCase LT to control changes to source code and other project configuration items such as Rational RequisitePro documents and Rational Rose model files. ClearCase LT tracks changes to every file and directory, maintaining histories of source code, binaries, executables, documentation, test suites, libraries, and user-defined objects.

UCM is a built-in process that you can use with ClearCase LT alone, and integrated with ClearQuest. As you work with UCM:

- Software files and directories are organized into versioned components.
- Project leaders create projects and assign project teams to work on these components.
- Team members make changes based on assigned activities (tasks, defects, changes requests).
- New file and directory versions are collected during development and associated with activities.
- After completion, activities and their modified artifacts are delivered and integrated in a shared project integration area.
- New component baselines are created, tested, and promoted.
- Multiple components are assembled to form a new application iteration.

## Managing Project Status Using Unified Software Project Management

Unified Software Project Management (USPM) is the Rational approach to managing software projects from requirements to release. USPM focuses on compiling information to assess project status, trends, quality, and other aspects critical to project management and articulation of progress. Rational ProjectConsole and Rational SoDA provide tool support for Unified Software Project Management.

**Rational ProjectConsole** extracts information from data produced during software development by tools like RequisitePro and ClearQuest. ProjectConsole automatically generates pre-formatted or customized charts and graphs depicting the collected metrics. These metrics let you share project status information with members of your team. ProjectConsole also lets you analyze data in a single view that reflects data collected from multiple sources. For example, you can use ProjectConsole to learn how many requirements and defects have been implemented and tested.

**Rational SoDA** extracts current information from one or more Rational tools and generates project reports. For instance, to plan the next iteration, you might create a SoDA report containing the latest information on requirements from RequisitePro and defects from ClearQuest. To create a project report, you start with a Microsoft Word template, either one provided with SoDA or one you create yourself. This report provides the information you need to plan the next iteration.

You can use the project report to:

- Evaluate how well the team is meeting project requirements.
- Select requirements and enhancements that the team will implement in the next iteration.
- Identify defects the team must fix in the next iteration to fulfill requirements.

## Managing Change and Risk in Web Development

---

As Web sites grow in size, complexity, and strategic value, so does the need to control changes. **Rational Suite ContentStudio** helps Web teams accelerate site changes and deploy them more quickly and confidently by:

- Enabling parallel development of Web applications.
- Providing predefined workflow options.
- Providing a simple to-do list for all team members.
- Automatically deploying code and content assets, simultaneously.

ContentStudio is optimized for Web development teams, particularly Web content managers and Web developers. In addition to the Team Unifying Platform, ContentStudio provides an integrated set of tools so Web teams can leverage scalable processes to build high quality applications at Internet speed.

## Using Templates to Accelerate Web Site Changes

**ContentStudio** offers distributed authoring capabilities, allowing Web content contributors to add and update content using templates. These templates allow the Web development team to separate the text and editable portions of a site from the design elements, so content contributors can modify Web material while maintaining a consistent look and feel of the Web site.

**ClearCase LT** brings versioning capabilities to the authoring environment. It versions Web application assets, including HTML pages, images, multimedia files, PDF files, and other file assets.

**ClearQuest** helps Web teams create customized workflows for site updates that best reflect the way their organization operates. It provides task and approval workflows that automate the circulation of proposed site changes to authorized reviewers for approval or edits. The workflow cycle helps to assure site integrity and reduce the need for Web teams to be involved in every site change. ClearQuest also provides a simple to-do list interface, so all team members have a clear picture of their priorities and the steps needed to manage them.

## Deploying Quickly and Confidently

After a workflow cycle is complete, content changes can be automatically posted to the Web site.

**Rational NetDeploy** simultaneously delivers all changes to related code and content to multiple staging or production servers. Rational NetDeploy lets you schedule one-time, recurring, or dependency-based deployments. It can also automatically identify groups of files (code and content) that need to be deployed together.

**ClearCase LT** is integrated with Rational NetDeploy to actively manage code and content changes so Web teams understand which site changes were made, where, and when. This unique integration of automated deployment with change management means Web teams always know exactly what is deployed on the site. And because all code and content assets are securely stored and versioned, there is a complete archive and history of your Web site, which can be rolled back to any previous state.

## Summary

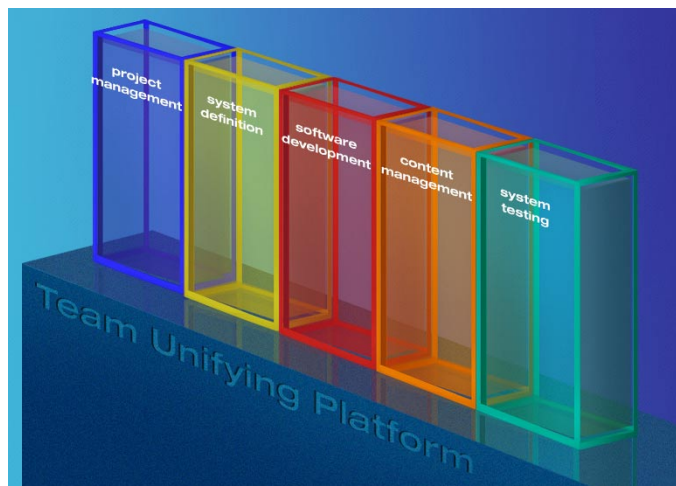
---

- Rational Suite supports software development best practices and offers tools to help project leaders realize greater success in planning iterations, ensuring integrity of artifacts, and enhancing communication among stakeholders.
- Iterative development lets your team manage risk and change early in the development process.
- UCM makes it easy to trace an activity from initial change request through final software release, all the while maintaining the integrity of the project assets.
- ClearQuest lets you manage the project's activities (tasks, defects, and requests for enhancements) and provides the charting and reporting tools necessary to track project progress. ClearCase LT helps you accelerate the development cycle by enabling your team to organize and manage versions, releases, and, when necessary, parallel development of multiple products.
- Unified Software Project Management focuses on compiling information to assess the project status.
- ProjectConsole automatically generates project metrics by extracting information from data produced during software development so that you can obtain an accurate and objective assessment of the project status.
- SoDA lets you extract and compile critical information so that you can articulate the project status to stakeholders.
- ContentStudio is optimized for Web development teams, providing an integrated set of tools to unify code and content management for Web sites. Web teams use ContentStudio to accelerate Web site changes and deploy those changes quickly and confidently.

Rational Suite can help your project team manage communication, change, and risk, to effectively meet the challenges of developing quality software:

- The different Rational Suite editions maximize the productivity of your team by integrating products and automating tasks.
- The Suite integrated tools help team members work together more effectively by enhancing communication between the major team roles: analyst, developer, tester, and project leader (see Figure 7).
- Rational Suite gives you the support to communicate and track changes in a repeatable and predictable manner so you can control changes to your project.
- By managing change and implementing the most risky project features in early iterations, your team can correct serious project flaws with less difficulty and at less expense.
- Within the framework of iterative development, Rational Suite promotes the most effective software engineering practices to help you solve the right problem and define the right solution.

**Figure 7 Rational Suite: Tools Optimized for the Entire Development Team**



# Adopting Practices and Tools

---

## Adopting All of Rational Suite

Your team may decide to use the entire Rational tool set in your next project. The Rational Unified Process (RUP) and the Rational Developer Network can help.

You can use the RUP searchable knowledge base to help your team incorporate effective software engineering practices into your project lifecycle. To support this integration, the Rational Unified Process provides links between the guidelines and tools:

- Tool Mentors provide instructions for performing activities using Rational tools.
- Extended Help (Windows only) links the tools to Rational Unified Process guidelines. Your team can also add its own content to Extended Help.

You can use the Rational Developer Network to help your team speed projects to completion and expand professional skills by offering:

- Instant access to relevant content, best practices, and industry experts.
- Web-based training and a comprehensive third-party course catalog.
- Discussions monitored by technology experts.

## Adopting Rational Suite Gradually

Rational Suite can also help your team work effectively when your group gradually incorporates Suite tools into its process.

As your team re-evaluates its software development process, it prioritizes its development problems and decides which ones the group should tackle first. In the next project, your team should identify the Rational Suite tools that will address the most severe problems. After you've identified the next Rational tools to use, Tool Mentors and Extended Help can guide you in using them.

For example, your team may decide to focus on requirements management. The analysts and project leader learn to use RequisitePro and Rose, and perhaps, integrate some Rational Unified Process guidelines about requirements analysis into the development process. Or, team members may be familiar with Rational Rose, so your group makes full use of visual modeling, relying on Rational Unified Process guidelines to direct modeling work.



## Exploring Rational Suite Tools and the Rational Unified Process

To gain a basic understanding of how software teams use Rational Suite, start by reading the *Rational Suite Tutorial* (for Windows only). The *Rational Suite Tutorial* teaches you the basics of using Rational Suite to plan, design, implement, and test applications. This tutorial also points you to other information about Rational Suite tools, so that you can learn more on your own.

There are several ways to start the *Rational Suite Tutorial*:

- 1 If Rational Suite is installed on your system, do one of the following:
  - a Click **Start > Programs > Rational Suite > Rational Suite Tutorial**. Then, follow the instructions to install the tutorial application files and documentation.
  - b Go to the Rational Developer Network by clicking **Start > Rational Suite > Logon to the Rational Developer Network**. You may need to log on to, or create, your Rational Member Profile. Browse to and click the link **Rational Suite Tutorial**. Then, follow the instructions to install the tutorial application files and documentation.
- 2 If Rational Suite is *not* installed on your system, you can still benefit from reading the tutorial, but you will not be able to perform the exercises. Go to <http://www.rational.com/documentation>. Browse to and click the link **Rational Suite Tutorial**. You may need to log on to, or create, your Rational Member Profile. After you complete these steps, follow the instructions to download the documentation.

## Contacting Rational Professional Services

---

Rational Software offers a complete range of professional services to support Rational Suite. The goal of these services is to help software development teams consistently produce quality software on time and within budget.

### Assessment Services

Rational Software offers mentoring and consulting, Project Assessment Services, and Project Implementation Services to organizations needing a comprehensive assessment of their development environment. Our consultants study your operations, identify the risks, and determine the most effective implementation plan for your unique situation. For more information, see <http://www.rational.com/services>.

## Rational University

Professionally trained instructors deliver courses on Rational tools and effective development practices at locations nationwide and at Rational Partner locations around the world. Rational and its partners also offer onsite delivery of these courses. To find information about courses, schedules, and registration, see <http://www.rational.com/university>.

## Rational Customer Support

If you have questions regarding the installation, use, or maintenance of Rational Suite, see the Rational Customer Support Web site for the appropriate e-mail address or phone number: <http://www.rational.com/support/contact>.

## Online Technical Resources

You can use Rational online resources to answer your support questions. You can download patches and upgrades and read our technical papers, release notes, and answers to frequently asked questions. You can also join Rational user groups to share advice and the latest information with other organizations.

All resources are free to use and are instantly available, twenty-four hours a day at <http://www.rational.com/support>.

## Additional Resources

Rational Software authors have written extensively about project management, application development, visual modeling, and other related topics. An annotated bibliography is available in the Rational Unified Process.

For articles, discussion forums, and Web-based training courses on developing software with Rational Suite products, join the Rational Developer Network by clicking **Start > Rational Suite > Logon to the Rational Developer Network**.

You can also find current papers and presentations about Rational Suite on the Rational Web site at <http://www.rational.com>.

To learn more about the Unified Modeling Language, see the Rational Software UML Resource Center at <http://www.rational.com/uml>.

# Glossary

**activity.** A unit of work that a team member performs.

**Agent computer.** A computer that has the Rational Agent software installed and that plays back a test script. You can set TestManager to play back a test script on a specific Agent computer, or on the next Agent computer that is free to run test scripts. Agents computers are typically used in UNIX editions of Rational Suite DevelopmentStudio.

**analyst.** A person who determines what the system does, specifies and manages requirements, and represents the user's needs to the development organization.

**artifact.** A piece of information that is produced, modified, or used by a process; defines an area of responsibility; and is subject to version control. There are many types of artifacts, including requirements, models, model elements, and documents.

**automated testing.** A testing technique wherein you use software tools to replace repetitive and error-prone manual work. Automated testing saves time and enables a reliable, predictable, and accurate process.

**baseline.** A consistent set of artifact versions that represent a stable configuration for a project's components.

**build.** An operational version of a system or part of a system that demonstrates a subset of the capabilities to be provided in the final product.

**component.** A nontrivial, nearly independent, and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture.

**component-based architecture.** A design technique in which a software system is decomposed into individual components.

**configuration management.** Helps teams control their day-to-day management of software development activities as software is created, modified, built, and delivered. Comprehensive software configuration management includes version control, workspace management, build management, and process control to provide better project control and predictability.

**developer.** A person who determines how the system works; defines the architecture; and creates, modifies, and manages the code.

**edition.** Sets of Rational Suite tools that are customized for each functional area of a software development team.

**element.** An object that encompasses a set of versions, organized into a version tree. Elements can be either files or directories.

**Extended Help.** A powerful feature of Rational Suite for Windows that provides links to the Rational Unified Process and any customized information you want to add.

**feature creep.** A term used by software development teams to describe the tendency to add unplanned changes to product features throughout (and often late in) the development process.

**forward engineering.** The process of generating code from a Rational Rose visual model. See *visual model*.

**iterative development.** The process of delivering a distinct sequence of executable files according to a plan and evaluation criteria over the course of a project. Each executable file is more robust or contains more features than the previous executable file; each new iteration moves you closer to the goal of delivering a successful project.

**metrics.** The measurements of project activity.

**phase.** The time between two major project milestones, during which a well-defined set of objectives is met, artifacts are completed, and decisions are made to move or not move into the next phase.

**project leader.** A person who allocates resources, shapes priorities, coordinates interactions with the customers and users, and generally tries to keep the project team focused on the right goal. A project leader also establishes a set of practices that ensures the integrity and quality of project activities and artifacts.

**Rational Administrator.** Tool that manages Rational projects and associates repositories to define a Rational project. For more information, see *Rational Suite Administrator's Guide*.

**Rational ClearCase LT.** Provides comprehensive configuration management, including version control, workspace management, and process control.

**Rational ClearQuest.** A highly customizable change request management tool that lets users track any type of change activity – defects and fixes, enhancement requests, documentation changes, and so on – throughout the software development lifecycle. The ClearQuest Web interface lets users to perform all major ClearQuest operations.

**Rational ClearQuest MultiSite.** A highly customizable Windows and Web-based change request management tool that lets geographically distributed users track any type of change activity – defects and fixes, enhancement requests, documentation changes, and so on – throughout the software development lifecycle by easily replicating a centralized database to each remote site and then synchronizing the changes made at each site with changes made at other sites.

**Rational Developer Network.** A Web-enabled, searchable knowledge base that aggregates best practices, reusable artifacts and assets, and Web-based training to help software professionals expand their professional skills. The Rational Developer Network is available exclusively to Rational Suite customers.

**Rational NetDeploy.** Automates Web deployment of code and content with scheduling and expiration features.

**Rational Process Workbench.** A customizable, Web-enabled, searchable knowledge base that enhances team productivity and delivers company-specific best practices using guidelines, templates, and Tool Mentors for software development activities.

**Rational ProjectConsole.** A highly customizable project management tool that lets users select best practices, plan and manage iterative projects, and measure progress and quality throughout the software development lifecycle.

**Rational PureCoverage.** Automatically pinpoints areas of code that have not been tested.

**Rational Purify.** Automatically pinpoints hard-to-find run-time memory errors in software applications.

**Rational Quantify.** Automatically pinpoints performance bottlenecks in Visual Basic, Visual C++, and Java applications.

**Rational RequisitePro.** Helps teams easily and comprehensively organize, prioritize, track, and control changing requirements of a system or application. Rational RequisitePro does this through a deep integration with Microsoft Word and a secure, multiuser database. The RequisiteWeb interface lets users to perform all major RequisitePro operations.

**Rational Robot.** Helps with functional and load testing by automating record and playback of test scripts. Lets you organize, write, and run test suites, and capture and analyze the results.

**Rational Rose.** The world's leading visual component modeling and development tool; lets you model software applications that meet current business needs.

**Rational Rose RealTime.** The UML-based modeling and development environment for real-time, embedded software development.

**Rational SiteLoad.** Testing tool that simulates Internet traffic and provides developers with precise real-time information on Web site performance.

**Rational SoDA.** Software Documentation Automation – Overcomes the obstacles of consolidating data from different development tools. Lets you automate the creation of comprehensive software, systems, and project documents from multiple sources.

**Rational Suite.** An easy-to-adopt-and-support solution that unifies software teams and optimizes the productivity of analysts, developers, testers, and project managers.

**Rational Suite AnalystStudio.** Windows edition of Rational Suite optimized for system definition. Contains the *Team Unifying Platform* and Rational Rose (Professional Data Modeler Edition).

**Rational Suite ContentStudio.** Windows edition of Rational Suite optimized for companies managing content and code in complex Web applications. Contains the *Team Unifying Platform*, Rational NetDeploy and Rational SiteLoad.

**Rational Suite DevelopmentStudio.** Edition of Rational Suite optimized for software development, available for Windows or UNIX. Contains tools from the *Team Unifying Platform* and Rational Rose (Enterprise Edition), Rational Purify, Rational Quantify, and Rational PureCoverage.

**Rational Suite DevelopmentStudio—RealTime.** Edition of Rational Suite optimized for system developers and designers of real-time or embedded systems, available for Windows or UNIX. Contains the *Team Unifying Platform* and Rational Rose RealTime, Rational Purify, Rational Quantify, and Rational PureCoverage.

**Rational Suite Enterprise.** Edition of Rational Suite containing all Rational Suite tools.

**Rational Suite Team Unifying Platform.** Edition of Rational Suite optimized for all members of software development teams to maximize productivity and quality. This Suite edition includes Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, TestManager, and ProjectConsole. All Suite editions include tools from the *Team Unifying Platform*.

**Rational Suite TestStudio.** Edition of Rational Suite optimized for testers. Contains the *Team Unifying Platform* and Rational PureCoverage, Rational Purify, Rational Quantify, Rational Robot and Rational TestFactory.

**Rational TestFactory.** Automates reliability testing by combining automatic test generation with source code coverage analysis.

**Rational TestManager.** Provides management and control of all test activities from a single, central point, including the ability to control and view legacy and proprietary test assets. It improves team productivity by making test results and progress toward goals immediately available to all team members. The TestManager Web interface lets users to perform all major TestManager operations.

**Rational Unified Process.** A Web-enabled, searchable knowledge base that enhances team productivity and delivers software best practices using guidelines, templates, and Tool Mentors for all critical software development activities.

**real-time software.** An application or system with stringent requirements for latency, throughput, reliability, and availability.

**requirement.** A condition or capability of a system, either derived directly from user needs or stated in a contract, standard, specification, or other formally imposed document.

**requirements management.** A systematic approach to eliciting, organizing, and documenting a system's changing requirements, and establishing and maintaining agreement between the customer and the project team.

**reverse engineering.** The process of creating or updating a Rose visual model from existing code, so that the visual model and code are synchronized. See *visual model*.

**risk.** The probability of adverse project impact (for example, schedule, budget, or technical).

**risk management.** Consciously identifying, anticipating, and addressing project risks and devising plans for risk mitigation, as a way of ensuring the project success.

**round-trip engineering.** The ability to generate code from a Rose visual model (see *forward engineering*), and to update a Rose model file from source code (see *reverse engineering*).

**stakeholder.** An individual who is materially affected by the outcome of the system.

**test case.** A set of test inputs that describe a testable and verifiable behavior in a system, the extent to which you will test an area of an application, and the results of each test.

**tester.** A person who creates, manages, and runs tests; ensures that the software meets all its requirements; and reports the results and verifies fixes.

**test input.** Any artifact used to develop a system, and that may be used to influence testing.

**Tool Mentor.** Step-by-step instructions on how to use a specific Rational tool to perform an activity described in the Rational Unified Process.

**Unified Change Management (UCM).** The Rational approach to managing change in software development, from requirements to release. UCM spans the development lifecycle, defining how to manage changes to requirements, design models, documentation, components, test cases, and source code.

**Unified Modeling Language (UML).** The industry-standard language for specifying, visualizing, constructing, and documenting software systems. UML simplifies software design, and communication about the design.

**Unified Software Project Management (USPM).** The Rational approach to managing software projects, from requirements to release. USPM spans the development lifecycle, focusing on compiling information to assess status, trends, quality, and other aspects critical to project management and articulation of progress.

**version control.** The process of tracking the revision history of files and directories.

**visual model.** A graphic representation of a system's structure and interrelationships.

**workflow.** The sequence of activities performed by roles to attain an observable value.





# Index

## A

- activities 22, 26
  - about 59
- Agent computers 37
  - about 59
- analysis and design
  - about 31
  - and developers 31
  - workflow 32
- analysts
  - about 59
  - common problems 25
  - defining requirements 25
  - managing change 28
  - managing requirements 25, 26, 27
  - measuring progress 29
  - understanding problems 25
  - using ClearCase LT 28
  - using ClearQuest 28, 37
  - using ProjectConsole 29
  - using RequisitePro 27
  - using SoDA 29
  - using TestManager 29
  - using visual models 28
  - verifying quality 29
- AnalystStudio 16, 17, 27
  - about 61
- architectures
  - component-based 15, 33, 38, 50, 59
  - executable 38
- artifacts 28, 59
- automated testing 44, 45, 59

## B

- baselines 59
- best practices 14, 15, 43, 49, 50
  - implementing 21, 49

## C

- change management 50
  - about 15
  - and analysts 28
  - and developers 33, 34
  - and project leaders 50
  - and testers 45
  - and Web teams 53
- ClearCase LT 18, 28, 34, 35, 51, 53
  - about 60
  - and analysts 28
  - and developers 34
  - and project leaders 51
  - and Unified Change Management 18, 34, 50, 51
  - and Web teams 53
  - Web interface 18
- ClearQuest 18, 28, 29, 34, 35, 37, 45, 51, 52, 53
  - about 60
  - and analysts 28, 37
  - and developers 34
  - and project leaders 51
  - and testers 45
  - and Unified Change Management 34, 50, 51
  - and Web teams 53
  - MultiSite 18, 28, 60
  - Web interface 18, 28, 34, 37, 45
- collecting project metrics 18, 29, 35, 43, 52
- component testing 36, 45
- configuration management 18, 51
  - about 59
- ContentStudio 17, 52, 53
  - about 61

## D

- defects 18, 28, 34, 37, 44, 45, 51
- designing architectures 31, 32, 33, 38
- developers
  - about 31, 59
  - common problems 31, 32
  - designing systems 33, 35
  - embedded systems 33, 38
  - managing change 33, 34
  - managing requirements 34
  - measuring progress 35
  - modeling 33
  - real-time software 33, 38
  - testing 35, 36
  - using ClearCase LT 34
  - using ClearQuest 34, 37
  - using DevelopmentStudio 33
  - using ProjectConsole 35
  - using PureCoverage 36
  - using Purify 36
  - using QualityArchitect 35
  - using Quantify 36
  - using RequisitePro 34
  - using Rose RealTime 38
  - using SoDA 35
  - using TestManager 36
  - verifying quality 35, 36
- developing iteratively 14, 31, 41, 47, 49
  - using Rational Unified Process 22
- developing software
  - common problems 31
  - developers 31, 32
  - embedded systems 33, 38
  - for the Web 53
  - real-time systems 33, 38
  - testers 41, 47
  - using Rose 28, 33, 35, 38
- DevelopmentStudio 17, 33
  - about 61
- DevelopmentStudio - RealTime 17, 33, 38
  - about 61

## diagrams

- analysis and design workflow 32
- project management workflow 48
- Rational Unified Process overview 21
- testing workflow 42

## E

- embedded systems 33, 38
- Enterprise edition
  - about 17
- Extended Help 23, 56
  - about 59

## F

- feature creep 59
- forward engineering 33, 59
- functional testing 44
- functionality testing 43

## G

- generating code 33
- generating project reports
  - using SoDA 18, 29, 35, 43, 52

## H

- Help
  - Extended 23, 56

## I

- implementing code 33
- iterative development
  - about 60
  - and analysts 26, 27
  - and developers 31
  - and project leaders 47, 49
  - and testers 41
  - challenges 26, 27, 31, 41, 47

## L

load testing 45

## M

managing change 15, 33, 34, 50  
  about 28  
  and project leaders 47  
  using ClearCase LT 18, 28, 34, 51, 53  
  using ClearQuest 18, 28, 34, 37, 45, 51, 53  
managing project status 18, 29, 35, 43, 52  
managing projects 47  
managing requirements 14, 26, 27, 34, 43, 49  
  about 25  
  using RequisitePro 18, 27, 34, 43  
managing risk 47  
measuring progress 35, 43, 44, 52  
  about 29  
  and analysts 29  
  and developers 35  
  and project leaders 52  
  and testers 43, 44  
memory leaks 36, 45  
metrics 18, 29, 35, 43, 52, 60  
modeling 15, 33, 50  
  about 28

## P

performance testing 43, 45  
  about 36, 45  
planning projects 47  
planning test 44  
production load 45  
professional services  
  about 19  
  assessment 57  
  customer support 58

  online technical resources 58  
  Rational University 58  
project leaders  
  about 47, 60  
  common problems 47  
  managing change 50  
  measuring progress 52  
  using ClearCase LT 51  
  using ClearQuest 51  
  using ProjectConsole 52  
  using SoDA 52  
project management 47  
  about 47  
  and project leaders 47  
  workflow 48  
project planning 47  
ProjectConsole 18, 29, 35, 43, 52  
  about 29, 35, 43, 52, 60  
  and analysts 29  
  and developers 35  
  and project leaders 52  
  and testers 43  
PureCoverage 36, 45  
  about 36, 60  
  and developers 36  
  and testers 45  
Purify 36, 45  
  about 36, 60  
  and developers 36  
  and testers 45

## Q

QualityArchitect 35  
  about 35  
  and developers 35  
  in analysis and design 35  
Quantify 36, 45  
  about 36, 60  
  and developers 36

## R

- Rational Administrator 60
- Rational ClearCase LT
  - See ClearCase LT
- Rational ClearQuest
  - See ClearQuest
- Rational Developer Network 19, 56, 58
  - about 60
- Rational NetDeploy 60
  - about 53
  - and Web teams 53
- Rational Process Workbench 60
- Rational ProjectConsole
  - See ProjectConsole
- Rational PureCoverage
  - See PureCoverage
- Rational Purify
  - See Purify
- Rational QualityArchitect
  - See QualityArchitect
- Rational Quantify
  - See Quantify
- Rational RequisitePro
  - See RequisitePro
- Rational Robot
  - See Robot
- Rational Rose
  - See Rose
- Rational Rose RealTime
  - See Rose RealTime
- Rational SoDA
  - See SoDA
- Rational Software
  - learning resources ix
  - mission 13
  - professional services 19, 57
- Rational Suite
  - about 61
  - AnalystStudio 16, 27, 61
  - benefits 16, 49, 55
  - ContentStudio 17, 61
  - DevelopmentStudio 17, 33, 61
  - DevelopmentStudio - RealTime 17, 33
  - DevelopmentStudio -RealTime 38
  - documentation roadmap - UNIX xi
  - documentation roadmap - Windows x
  - Enterprise 17, 61
  - Extended Help 23, 56
  - implementing 56
  - overview ix
  - professional services 16
  - Rational Developer Network 19
  - summary table 20
  - Team Unifying Platform 16, 17, 49, 61
  - TestStudio 17, 61
  - Tutorial 57
- Rational Synchronizer 61
- Rational TestFactory
  - See TestFactory
- Rational TestManager
  - See TestManager
- Rational Unified Process 16, 17, 21
  - about 17, 62
  - analysis and design workflow 32
  - implementing 22, 56
  - iterations 22
  - overview diagram 21
  - phases and iterations 22
  - project management workflow 48
  - requirements workflow 26
  - testing workflow 42
  - Tool Mentors 23, 56
  - using 21
  - workflows 22

- real-time software 33, 38
  - about 62
- reliability testing 43, 44
- requirements 25, 27, 34, 43
  - about 62
  - and analysts 25, 26, 27
  - and developers 34
  - and project leaders 49
  - and testers 43
  - communicating 27
  - defining 25
  - managing 14, 49, 62
  - testing 44
  - workflow 26
- requirements management
  - and analysts 27
- RequisitePro 18, 27, 34, 43
  - about 61
  - and analysts 27
  - and developers 34
  - in analysis and design 34
  - Web interface 18, 27, 43
- reverse engineering 62
- risk 47, 62
  - about 62
- Robot 44, 45
  - about 44, 61
  - and testers 44
- roles 22
- Rose 33, 35, 61
  - about 28, 33, 38, 61
  - and developers 38
  - Enterprise 35
  - QualityArchitect 35
  - RealTime 38
- round-trip engineering 62

## S

- SiteLoad 61
- SoDA 18, 29, 35, 43, 52
  - about 61
  - and analysts 29
  - and developers 35
  - and project leaders 52
  - and testers 43
  - for FrameMaker 29, 35, 43
  - for Word 29, 35, 43
- software development
  - activities 22
  - best practices 14, 15, 49, 50
  - builds 59
  - common problems 13
  - disciplines 22
  - iterations 22
  - phases 22, 60
  - principles 13
  - team roles ix, 22
  - Web applications 52, 53
- stakeholders 25
  - about 62

## T

- Team Unifying Platform 16, 49
  - about 61
  - tools 17
- test agents 37, 59
- test cases 29, 44
  - about 62
- test input 62
- test planning 44

- testers
  - about 41, 62
  - common problems 41
  - managing requirements 43
  - measuring progress 43, 44
  - testing 44
  - using ClearQuest 45
  - using ClearQuest Web 45
  - using ProjectConsole 43
  - using PureCoverage 45
  - using Purify 45
  - using Quantify 45
  - using RequisitePro 43
  - using Robot 44
  - using SoDA 43
  - using TestFactory 44
  - using TestManager 44, 45
  - using TestStudio 43
- TestFactory 61
  - about 44
  - and testers 44
- testing 35, 36, 41, 43
  - about 29
  - and testers 41
  - automated 44, 45
  - code 35, 36, 45
  - components 36, 45
  - early 35
  - finding memory leaks 36, 45
  - for quality 15, 18, 29, 35, 36, 43, 44, 50
  - functional 43, 44
  - performance 36, 43, 45
  - reliability 43, 44
  - requirements 29, 44
  - results 36
  - using TestManager 18, 29, 36, 44
  - workflow 42
- TestManager 18, 29, 36, 44
  - about 62
  - and analysts 29
  - and developers 36
  - and testers 44
  - UNIX test agents 37, 59
- TestStudio 17, 43
  - about 61
  - and testers 43
- Tool Mentors 23, 56
  - about 62
- tracking defects
  - using ClearQuest 18, 28, 34, 37, 45, 51
- tracking test results 36, 44
- tutorials
  - Rational Suite 57

## U

- understanding problems
  - analysts 25
- Unified Change Management 18, 34, 50, 51
  - about 62
  - and ClearCase LT 18, 50
  - and ClearQuest 34, 50
- Unified Modeling Language 15, 29
  - about 62
- Unified Process
  - See Rational Unified Process
- Unified Software Project Management 52
- use cases 29

## V

- verifying quality 15, 29, 35, 36, 43, 50
  - and analysts 29
  - and developers 35, 36
  - and testers 41
  - using TestManager 18, 29, 36, 44
- version control 63
- visual modeling 15, 28, 33, 50
  - about 63
  - and analysts 28
  - and developers 33

## W

Web development 52, 53

Web teams

about 52

common problems 52

using ClearCase LT 53

using ClearQuest 53

Web teams (continued)

using ContentStudio 53

using Rational NetDeploy 53

workflow diagrams

about 26

workflows 22

about 63

