Rational Software Corporation®

Rational Suite®

Introduction

VERSION: 2003.06.00

PART NUMBER: 800-026064-000

WINDOWS/UNIX

Rational®
the software development company

support@rational.com
http://www.rational.com

Legal Notices

Part Number: 800-026064-000
Version Number: 2003.06.00

U.S. Government Restricted Rights

Licensee agrees that this software and/or documentation is delivered as "commercial computer software," a "commercial item," or as "restricted computer software," as those terms are defined in DFARS 252.227, DFARS 252.211, FAR 2.101, OR FAR 52.227, (or any successor provisions thereto), whichever is applicable. The use, duplication, and disclosure of the software and/or documentation shall be subject to the terms and conditions set forth in the applicable Rational Software Corporation license agreement as provided in DFARS 227.7202, subsection (c) of FAR 52.227-19, or FAR 52.227-14, (or any successor provisions thereto), whichever is applicable.

Warranty Disclaimer

This document and its associated software may be used as stated in the underlying license agreement. Except as explicitly stated otherwise in such license agreement, and except to the extent prohibited or limited by law from jurisdiction to jurisdiction, 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 , non-infringement, title or fitness for a particular purpose or arising from a course of dealing, usage or trade practice, and any warranty against interference with Licensee's quiet enjoyment of the product.

Third Party Notices, Code, Licenses, and Acknowledgements

Contents

Preface

This book provides an introduction to Rational Suite tools and to Rational Software best practices so that you can:

- Improve software quality.
- Reduce development time.
- Improve team communication.

Audience

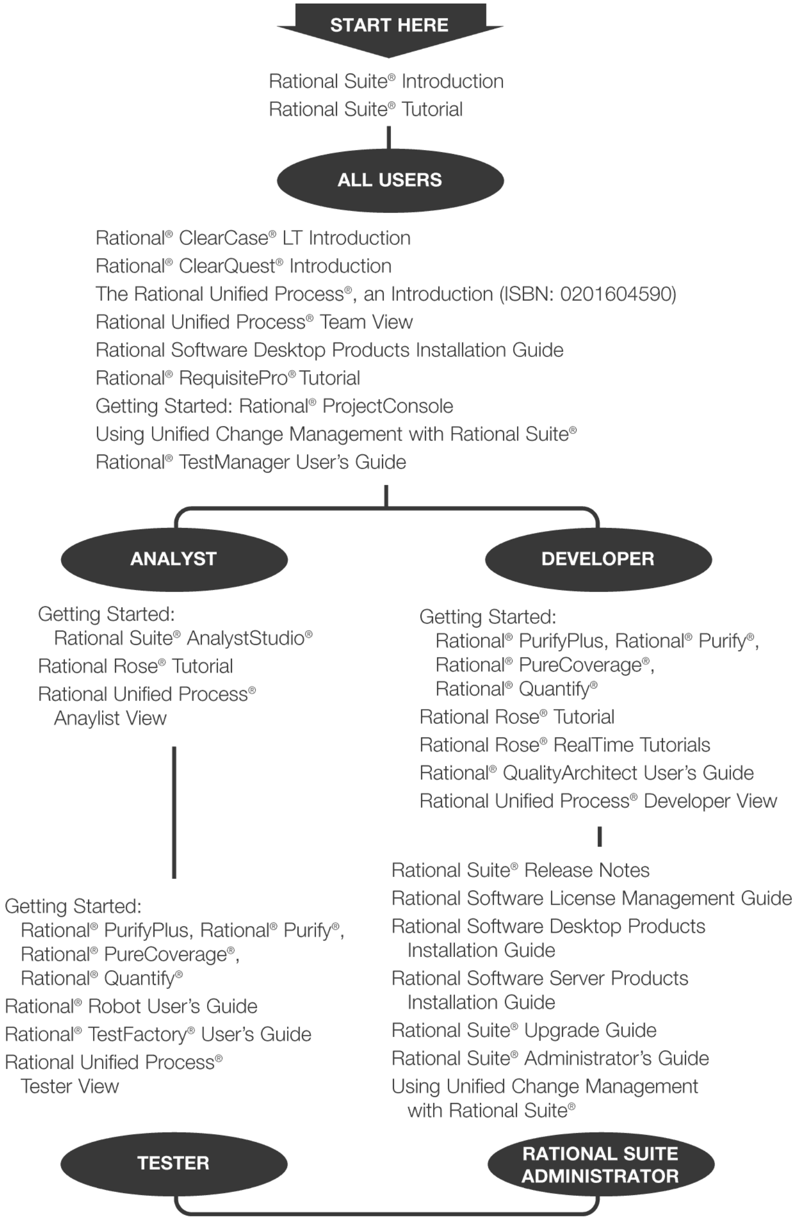
This book is intended for software development professionals who are:

- Beginning to use the Rational Suite.
- Interested in learning about integrated software development using Rational Suite.

Other Resources

- Technical documentation is available from:
  - http://www.rational.com/documentation.
  - The Rational Solutions Online Documentation CD.
- Feedback: Send questions or comments about Rational documentation to techpubs@rational.com.
- Rational University: http://www.rational.com/university.
- Rational Developer Network: http://www.rational.net.

Rational Suite Documentation Roadmap - Windows



START HERE

Rational Suite® Introduction
Rational Suite® Tutorial

ALL USERS

Rational® ClearCase® LT Introduction
Rational® ClearQuest® Introduction
The Rational Unified Process®, an Introduction (ISBN: 0201604590)
Rational Unified Process® Team View
Rational Software Desktop Products Installation Guide
Rational® RequisitePro® Tutorial
Getting Started: Rational® ProjectConsole
Using Unified Change Management with Rational Suite®
Rational® TestManager User's Guide

ANALYST

Getting Started:
Rational Suite® AnalystStudio®
Rational Rose® Tutorial
Rational Unified Process®
Anaylist View

DEVELOPER

Getting Started:
Rational® PurifyPlus, Rational® Purify®,
Rational® PureCoverage®,
Rational® Quantify®
Rational Rose® Tutorial
Rational Rose® RealTime Tutorials
Rational® QualityArchitect User's Guide
Rational Unified Process® Developer View

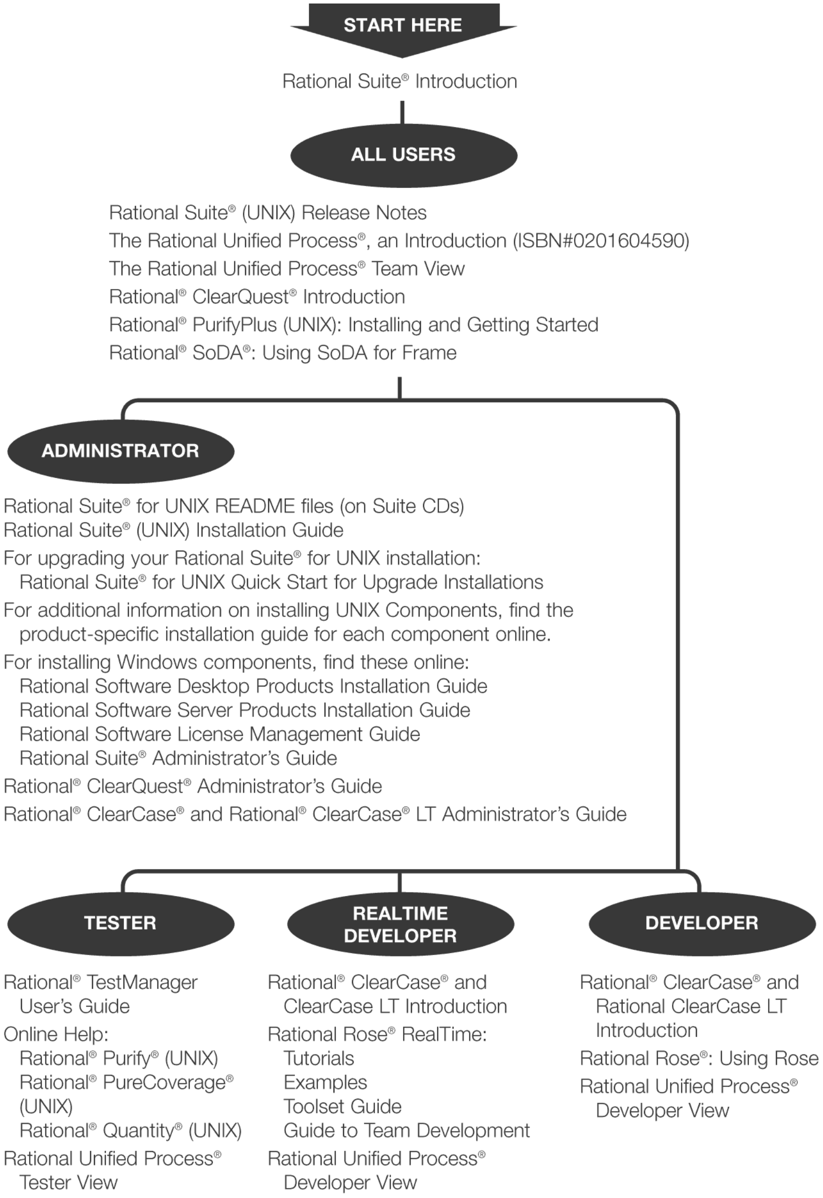
Getting Started:
Rational® PurifyPlus, Rational® Purify®,
Rational® PureCoverage®,
Rational® Quantify®
Rational® Robot User's Guide
Rational® TestFactory® User's Guide
Rational Unified Process®
Tester View

TESTER

Rational Suite® Release Notes
Rational Software License Management Guide
Rational Software Desktop Products
Installation Guide
Rational Software Server Products
Installation Guide
Rational Suite® Upgrade Guide
Rational Suite® Administrator's Guide
Using Unified Change Management
with Rational Suite®

RATIONAL SUITE
ADMINISTRATOR

Rational Suite Documentation Roadmap - UNIX



START HERE

Rational Suite® Introduction

ALL USERS

Rational Suite® (UNIX) Release Notes
The Rational Unified Process®, an Introduction (ISBN#0201604590)
The Rational Unified Process® Team View
Rational® ClearQuest® Introduction
Rational® PurifyPlus (UNIX): Installing and Getting Started
Rational® SoDA®: Using SoDA for Frame

ADMINISTRATOR

Rational Suite® for UNIX README files (on Suite CDs)
Rational Suite® (UNIX) Installation Guide

For upgrading your Rational Suite® for UNIX installation:
Rational Suite® for UNIX Quick Start for Upgrade Installations

For additional information on installing UNIX Components, find the product-specific installation guide for each component online.

For installing Windows components, find these online:
Rational Software Desktop Products Installation Guide
Rational Software Server Products Installation Guide
Rational Software License Management Guide
Rational Suite® Administrator's Guide

Rational® ClearQuest® Administrator's Guide

Rational® ClearCase® and Rational® ClearCase® LT Administrator's Guide

TESTER

Rational® TestManager User's Guide

Online Help:
Rational® Purify® (UNIX)
Rational® PureCoverage® (UNIX)
Rational® Quantity® (UNIX)

Rational Unified Process® Tester View

REALTIME DEVELOPER

Rational® ClearCase® and ClearCase LT Introduction

Rational Rose® RealTime:
Tutorials
Examples
Toolset Guide
Guide to Team Development

Rational Unified Process® Developer View

DEVELOPER

Rational® ClearCase® and Rational ClearCase LT Introduction

Rational Rose®: Using Rose

Rational Unified Process® Developer View

Contacting Rational Customer Support

For questions about installing, using, or maintaining this product, contact Rational Customer Support:

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: Be prepared to supply your:

- Name, company name, telephone number, and e-mail address.
- Operating system, version number, and any service packs or patches applied.
- Rational product name and version number.
- Case ID number (for previously reported problems only).

Welcome to Rational Suite

1

Think about your last software project:

- Was it delivered on time?
- Was it released within its budget?
- Did it meet requirements, satisfy users, and perform reliably?
- Was communication among team members clear and timely?
- Was your development process repeatable?

Many project teams experience problems in these areas. Subsequently:

- Projects finish late (or not at all).
- Results do not match requirements.
- Serious design flaws are uncovered late in development.
- Defects are found after the software ships, instead of during development.

Making Projects More Successful

Rational Software helps organizations overcome these challenges and develop software more successfully by offering:

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

Rational Software Best Practices

Best practice	Project lifecycle tasks	Benefits
Develop software iteratively.	Analyze, design, and implement incremental subsets of the system.	▪ Mitigate risks earlier. ▪ Achieve higher quality. ▪ Increase reusability of project assets.
Manage requirements.	Find, document, organize, and track changing requirements.	▪ Mitigate risks associated with change. ▪ Ensure products meet their stated goals.
Use component architectures.	Design the structural elements of a system, how those elements behave, and decide how they fit into progressively larger systems. Identify and decide which subsystem components to develop, reuse, and buy.	▪ Improve product reliability, maintainability, and extensibility.
Model visually.	Create graphical designs of system architectures.	▪ Manage design complexity. ▪ Improve understanding of complex systems. ▪ Explore and compare design alternatives at a low cost. ▪ Form a basis for implementation.
Continuously verify quality.	Test in each iteration.	▪ Verify that the system delivers required functionality, reliability, and performs under load. ▪ Discovers problems with requirements, designs, and implementations early.
Manage change.	Establish repeatable development process.	▪ Improve team communication. ▪ Control risk. ▪ Improve resource allocation. ▪ Enable parallel development.

Rational Suite Tools

Rational puts these best practices to work by offering tools that:

- Unify teams and enhance communication.
- Optimize individual productivity.
- Simplify adoption with common installation, licensing, and user support plans.

Rational Suite editions are customized with sets of tools best suited for each member of your team. Click the hyperlinks in the table below to learn more about each edition.

		Project Leaders/ Managers	Analysts	Developers				Testers	All Roles
		Team Unifying Platform	Analyst Studio	DevelopmentStudio		DevelopmentStudio RealTime Edition		TestStudio	Enterprise
		Windows	Windows	Windows	UNIX	Windows	UNIX	Windows	Windows
Team Unifying Platform	Rational Unified Process	●	●	●	●	●	●	●	●
	Rational RequisitePro	●	●	●	● WEB	●	● WEB	●	●
	Rational ClearQuest	●	●	●	●	●	●	●	●
	Rational SoDA	● SW	● SW	● SW	● SF	● SW	● SF	● SW	● SW
	Rational ClearCase LT	●	●	●	●	●	●	●	●
	Rational TestManager	●	●	●	● TA	●	● TA	●	●
	Rational ProjectConsole	●	●	●		●		●	●
	Rational Rose		● M	● E	● U	● RT	● RT/U		● E
	Rational PureCoverage			●	●	●	●	●	●
	Rational Purify			●	●	●	●	●	●
	Rational Quantify			●	●	●	●	●	●
	Rational Robot							●	●
	Rational TestFactory							●	●

E = Enterprise Edition
M = Professional Data Modeler Edition
RT = RealTime Edition
SF = Rational SoDA for FrameMaker
SW = Rational SoDA for Word
TA = Used in conjunction with test agents
U = UNIX Edition
WEB = Can be accessed through Web interface only

Tools That Unify Your Team

Rational Suite Team Unifying Platform provides integrated tools for managing change, building quality, and communicating results from requirements to release:.

Rational Unified Process (RUP). Helps you create and implement a development process so your team can work more efficiently and communicate more effectively. You can configure the RUP to:

- Use only the process components you need for each stage of a project.
- Develop your own process components.
- Exchange best practices with peers and industry leaders.

The RUP is available for Windows and UNIX.

Rational RequisitePro. Helps you manage requirements more effectively by using a database to:

- Organize requirements.
- Prioritize requirements.
- Track requirements.
- Develop requirement documents.

RequisitePro includes the RequisiteWeb interface so users can view, 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.
- Defects.
- Documentation modifications.

The ClearQuest Web interface allows Windows and UNIX users to perform basic ClearQuest operations from a Web browser, such as submitting and finding records, creating or editing queries and reports, and creating shortcuts. ClearQuest MultiSite enables sharing of information across a geographically distributed team. ClearQuest is available for Windows and UNIX client workstations. For either platform, you must configure a Windows workstation as the administrator for the ClearQuest repository.

Rational ClearCase LT. Provides software configuration management and a built-in process to track changes to all software project assets, including:

- Requirements.
- Visual models.
- Code.
- Documentation.

It also provides a Web interface that lets users perform basic ClearCase LT operations. Rational ClearCase LT supports *Unified Change Management*, the Rational 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 development artifacts, including files produced by Rational tools and source code. SoDA uses predefined or customized templates to format the information. SoDA is integrated with Microsoft Word for Windows and Adobe FrameMaker for UNIX.

Rational TestManager. Helps you create realistic functional and multiuser tests to assess the performance and reliability of Web, multitier, and database applications. TestManager tracks:

- How many tests have been planned, developed, and run.
- Which requirements have been tested.
- How many tests have passed and failed.

Using this data, TestManager helps you objectively assess project status and create reports to communicate these findings to project stakeholders. TestManager is available for Windows client workstations and UNIX agents as long as a Windows workstation is configured as the administrator for the TestManager repository.

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, display, and report on a centralized project plan. ProjectConsole organizes project artifacts on a central Web site so all team members can view them. ProjectConsole is available for Windows only.

Professional Services

The Rational Developer Network

The Rational Developer Network (RDN) is an online community for all Rational customers. RDN provides:

- A forum for exchanging ideas and best practices with other software professionals.
- Web-based training.
- Articles, white papers, *Getting Started* programs, and other content.

To join, go to http://www.rational.net.

More Rational Professional Services

Rational also offers:

- Consulting and mentoring.
- Packaged services, such as “Quickstarts,” deployment services, and workshops.
- Customized services and projects.

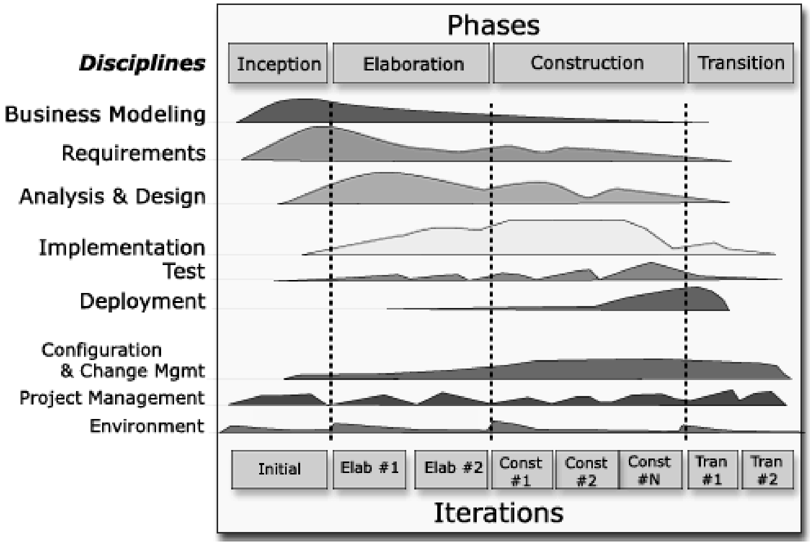
For more information, go to http://www.rational.com/services/service.jsp.

Getting Started with Rational Suite

Sustained delivery of high-quality software requires cohesive teamwork and a common understanding of development tasks. That is why implementing a predictable, repeatable process like the RUP is important to your success.

RUP Features	**RUP Benefits**
A library of processes and tools.	Select and deliver the best process for your projects.
A customizable Web interface so that you can design the process configuration.	Establish organizational process standards.
Tool Mentors provide instructions for using Rational tools. Extended Help (Windows only) from all Rational tools links to relevant topics in the RUP. You can also customize it by adding your own content.	Learn how to use Rational tools in your process.

The Rational Unified Process at a Glance



Rational collected empirical data to develop this overview that shows software development is best organized into *phases*:

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

Each phase is implemented in a series of *iterations*. Each iteration represents one complete pass through the *Disciplines*. And throughout development, team members from each discipline perform activities in one or more *workflows*.

The overview shows that emphasis on different workflows varies with each iteration. For example, in early iterations you spend more time on requirements; in later iterations you spend more time on implementation.

The steps in an iterative process correlate with the responsibilities for specific team roles: analyst, developer, tester, and manager/project leader. Each role performs their activities as early as possible and throughout each iteration. For example, analysts track requirements throughout each iteration and the project lifecycle even though it is emphasized at the start of an iteration.

Defining the Right System: The Analyst

2

Are you addressing your users' needs and solving their problems?

Did late feature requirements delay your last release?

Is your project over budget and behind schedule?

Are your users dissatisfied with or disappointed in your product?

Understanding the Problem Before You Invest

To clearly define the right system, analysts work with stakeholders to:

- Analyze the problem.
- Understand their needs.

Analysts interpret and articulate this information as requirements to help:

- Define the system.
- Manage the scope of the system.

Requirements and Iterative Development

During development, requirements change. For example:

- Competitors release new versions. To stay competitive, you add requirements for new features to your project.
- A customer was unclear or undecided about some requirements at first, so you might need to add, remove, or clarify requirements later on.
- Technology changes after you start development. You need to determine if you can incorporate this technology without jeopardizing the product's quality and schedule.
- A requirement becomes too expensive to implement or your team cannot implement it in the time allotted, so you decide to drop or postpone the requirement.

The challenge of managing requirements includes your understanding of how changing requirements affects the work of other team members. Analysts need to keep everyone informed about changes, prioritization of changes, and the impact the changes have on the development schedule.

Rational Suite AnalystStudio

Rational Suite AnalystStudio offers a complete set of tools to optimize requirements management.

Managing Requirements

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 flexibility of Microsoft Word:

- Analysts can use Microsoft Word to document requirements and communicate product vision, features, functionality, and attributes.
- Analysts store and track this information in the RequisitePro database. The database lets you assign attributes to requirements for organization, prioritization, and change history. These attributes can be traced to related requirements so you can quickly assess the impact of change on requirements throughout development.

Keeping Everyone Informed

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. Rational RequisiteWeb lets UNIX users and team members without access to RequisitePro create, review, and update requirements.

Working with Use Cases

RequisitePro helps analysts manage requirements and correlated *use cases* that express a system's intended functions and environment. Analysts develop use cases so:

- Developers can write system specifications and develop code.
- Testers can develop test plans and build test scripts.

Managing Change to Requirements

Rational ClearCase LT is a configuration management tool for small project teams. Analysts use it to manage changes to requirements. ClearCase LT lets you associate requirements with:

- Specific releases.
- Code.
- Visual models.
- Test scripts.

Because ClearCase LT tracks changes to all your project files, team members can work in parallel and integrate their changes to the project baseline.

Rational ClearQuest is a change management tool used to track defects, new features, product enhancements, and other change requests. It helps you:

- Evaluate change requests.
- Determine their impact on the system.
- Validate the changes (when applicable).

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.

Keeping Everyone Involved

Rational RequisiteWeb lets UNIX users and team members without access to RequisitePro create, review, and update requirements.

The Rational ClearQuest Web interface lets UNIX users and team members without access to ClearQuest participate in this collaboration.

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

Helping Your Team Communicate Visually

Rational Rose (Professional Data Modeler Edition) helps you create visual models to:

- Document the structure and behavior of your system's architecture.
- Manage and understand system complexity.
- Communicate impacts of change.

Analysts develop use case models to describe a system at a high level so:

- Developers can use the model as a basis for their system design work.
- Testers can plan their testing activities.

Team members can share and revise Rose models because they are written in the UML—an easily understood, industry-standard language for designing software.

Measuring Progress and Providing Project Reports

Rational ProjectConsole collects and stores data from Rational and some third-party development tools. It presents the results as reports and metrics on a common team Web site, ensuring that the entire team has access to current information. ProjectConsole can be configured so each team member has quick access to only that information required to perform their specific role.

Rational SoDA generates project reports with data extracted from Rational tools, such as RequisitePro and ClearQuest, so your team can communicate more efficiently and consistently. SoDA provides templates in Microsoft Word for Windows or Adobe FrameMaker for UNIX. You can customize templates or create new ones using ProjectConsole Template Builder.

Analysts use ProjectConsole and SoDA to:

- Monitor and communicate project status and progress.
- Determine how many requirements have been implemented.
- Determine how many requirements have been tested.

The Power of Use Cases

Analysts develop use cases that help testers build test assets with Rational TestManager, such as test plans and *test cases*, manual and automated tests, and test results. TestManager also helps manage test planning, design, development, execution, and analysis and share test results with team members. TestManager stores test assets in a central location, making them accessible to all team members.

Managing Complexity: The Developer 3

Do you discover design flaws too late to fix them?

Do project modules integrate correctly?

Can you maintain the integrity of the system architecture?

Can you easily extend the system design?

Can you reuse project components?

Does your team know what they are building?

How does your team communicate changes to each other?

Analysis and Design: The Next Steps in Iterative Development

Software developers use requirements to help define how a system should work and transform the idea into a system design. In Analysis and Design, developers:

- Refine the architecture.
- Define a candidate architecture.
- Perform architectural synthesis.
- Analyze behavior.
- Design components.
- Design the database.

Developers complete these activities to ensure that their system design is flexible, scalable, and can be modified as requirements change. They create, modify, and manage code for a system based on this concept of adaptability.

Rational Suite DevelopmentStudio

Rational Suite DevelopmentStudio (Windows and UNIX) offers a complete set of tools to design and build the right product the first time and test performance as soon as your system architecture is designed.

Using Visual Models to Manage Complexity

Rational Rose (Enterprise Edition for Windows, UNIX Edition for UNIX or Linux) helps you create visual models to:

- Document the structure and behavior of your system's architecture.
- Manage and understand system complexity.
- Communicate impact of change.

Developers use Rose and visual models to:

- Develop code in a familiar environment.
- Accelerate code implementation.
- Keep code and models consistent.
- Assess the risk of change.

Developing Code in a Familiar Environment

Rose supports many languages, such as Visual Basic, Visual C++, ANSI C++, and Java. Rose also supports many IDEs, such as IBM Visual Age for Java, Visual Studio, HP Workbench, and Sun Workshop.

Accelerating Code Implementation

Developers use Rose to *forward engineer* their applications, automatically generating code frameworks from existing models.

Keeping Code and Models Consistent

Developers use Rose to *reverse engineer* their applications, automatically synchronizing modified code with model, so they remain consistent throughout the project.

Assessing the Risk of Change

Using Rose to reverse engineer an application also helps you understand how changes affect system architecture. You can assess whether they violate project standards, compromise the requirements, or affect architectural decisions.

Evaluating Changing Requirements

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. Rational RequisitePro provides current requirements data in a form accessible to all stakeholders. Rational RequisiteWeb lets UNIX users and team members without access to RequisitePro create, review, and update requirements.

Developers use RequisitePro and RequisiteWeb to:

- Learn about changing requirements.
- Understand the impact changing requirements have on the system.
- Incorporate requirement changes into Rose models.

Validating Changes to the System

Rational ClearQuest is a change management tool used to track defects, new features, enhancements, and change requests. The Web interface lets UNIX users and team members without access to ClearQuest create, review, and provide feedback about this data.

Developers use ClearQuest with Rose to visualize and evaluate the effects change requests have 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 developers control changes to the following:

  - Rose model files
  - Source code
  - Binaries

- Executables
- Documentation
- Libraries
- User-defined objects

ClearCase LT also helps developers accelerate build and release cycles by using a development structure and process that supports parallel development.

Unified Change Management (UCM)

Unified Change Management (UCM) is the built-in Rational process that helps you use ClearCase LT with ClearQuest to:

- Define how to manage changing requirements, design models, documentation, components, and source code.
- Link the activities used to plan and track software development with the artifacts that are changing throughout development.

UCM Example

1. A project leader uses ClearQuest to assign an *activity* — change request — to a developer.
2. ClearCase LT with UCM maintains a *change set* — a list of changed artifacts — for the activity.
3. The developer works on the activity, creating or modifying artifacts associated with it.
4. The developer *delivers* the work to the *integration stream* — a project-wide, shared workspace.
5. Within this stream, team members can build and test the latest versions of the project's shared elements.
6. After, the project leader may decide to create a new *baseline* — a stable configuration of a project's components that becomes the basis for future development work.

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

Keeping the Team Up to Date

Rational ProjectConsole collects and stores data from Rational and some third-party development tools. It presents the results as reports and metrics on a common team Web site, ensuring that the entire team has access to current information. Project Console can be configured so each team member has quick access to only that information required to perform his or her specific role.

Rational SoDA generates project reports with data extracted from Rational tools so your team can communicate more efficiently and consistently. SoDA provides predefined and customizable templates in Microsoft Word for Windows or Adobe FrameMaker for UNIX.

Developers use ProjectConsole and SoDA to:

- Monitor and communicate project status and progress.
- Determine how many components have been developed.
- Determine how many components have been tested.
- Evaluate the quality of their components.

Testing Code Early and Often

Developers should test their code as soon as they implement it. Rational Suite provides testing tools to use as soon as you have an architecture or working program. These tools help you improve the reliability and performance of applications by:

- Generating test code.
- Detecting runtime errors.
- Analyzing path coverage in code.
- Identifying performance inefficiencies.

Generating Test Code

Rational QualityArchitect is a feature of Rational Rose (Enterprise Edition). It generates test code from visual models so developers can:

- Automatically generate component tests and building stubs and drivers before an application is complete.
- Determine whether a potential system architecture meets functional and performance requirements before they develop the design further.

Rational QualityArchitect supports Enterprise JavaBeans, COM, COM+, and DCOM models.

Note: Rational QualityArchitect is available with Rational Suite DevelopmentStudio for Windows only.

Detect Runtime Errors

Rational Purify checks for runtime or memory-access errors that often remain undetected until triggered by a random event. Rational Purify works checks C/C++ code (Windows and UNIX) and garbage-collection related memory-management issues in Java and .NET managed code (Windows only).

Analyzing Path Coverage

Rational PureCoverage generates a report that helps developers determine which lines of code have been tested.

Identifying Performance Inefficiencies

Rational Quantify helps developers build better software by:

- Pinpointing places in the application where the code is running inefficiently.
- Helping to identify the cause of inefficiencies.

Tracking Test Results

Rational TestManager helps manage test planning, design, development, execution, and analysis and share test results with team members. Developers use it to track which Rose model elements have been tested, and which tests pass or fail.

TestManager is available for use in both Windows and UNIX environments. TestManager runs on a Windows client and on Windows agents. 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. To learn more about this, see the *Rational TestManager User's Guide*.

Rational ClearQuest is integrated with Rational testing tools to simplify the process of entering defect information.

Rational Suite DevelopmentStudio—RealTime Edition

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

Building Complex, Real-Time Systems

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

- Uses 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.

Evaluating Quality: The Tester

4

How many critical bugs were discovered after you released your last product version?

Does your system scale to accommodate increasing user load?

Do you have a complete understanding of your software's quality before it is released?

How complete and thorough is your testing?

Assessing Quality and the Decision to Release

Testers plan, design, build, and run tests to produce data that helps determine whether software is ready to be released. Overall, testers:

- Validate build stability.
- Verify test approach.
- Improve test assets.
- Define evaluation missio.n
- Test and evaluate.
- Achieve acceptable mission.

Testing: The Last Step in Iterative Development

Typically, testers implement their tests incrementally, integrating and testing an executable release in each iteration. This helps testers verify quality throughout development.

To test iteratively, testers:

- Track use cases and requirements and use this information to develop test plans.
- Follow test plans to develop automated and manual tests.
- Implement tests to identify issues with reliability, functionality, and performance.
- Analyze test results to verify quality.

Rational Suite TestStudio

Rational Suite TestStudio offers a complete set of tools to assess software reliability, functionality, and performance.

Developing a Test Plan

Rational TestManager helps testers plan and design tests using project assets from other Rational tools. For example, TestManager helps testers:

- Create test plans and track testing progress.
- Use RequisitePro requirements, Rose models, and source code to develop *test inputs.*
- Use test inputs to create *test cases,* describing a testable and verifiable behavior in a system. Test cases can also describe the extent to which you will test an area of the application.
- Update tests when test inputs change to ensure that you are testing your application completely.
- Organize and run many types of tests, such as Rational Manual Tests, Perl scripts, batch files, and Rational Robot scripts.

TestManager is available for use in both Windows and UNIX environments. TestManager runs on a Windows client and on Windows agents. 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. To learn more about this, see the *Rational TestManager User's Guide*.

Testing Your Application

Automated Functional Testing

Rational Robot determines whether the system meets requirements by testing how it responds to a user-driven scenario. With Robot, you can:

- Record a test.
- Insert verification points to monitor expected behavior.
- Replay tests as often as needed.
- Run tests.
- View test results and details of test failures such as which test was running, what type of failure occurred, where the failure occurred, and which verification point failed.

Automated Reliability Testing

Rational Purify checks for runtime or memory-access errors that often remain undetected until triggered by a random event. Rational Purify works checks C/C++ code (Windows and UNIX) and garbage-collection related memory-management issues in Java and .NET managed code (Windows only).

Automated Application Performance Testing

Rational Quantify helps testers assess quality by:

- Pinpointing places in the application where the code is running inefficiently.
- Helping to identify the cause so developers can improve system performance.

Automated System Performance Testing

Rational TestManager helps testers simulate conditions in the system while it is being accessed by thousands of users. This lets you measure and predict system performance for Web, multitier, and database applications.

Building Comprehensive Tests

Rational Robot and Rational PureCoverage identify untested areas of your system so that you can:

- Build a more comprehensive set of tests.
- Increase quality.
- Mitigate risk.

Tracking Test Results

Rational ClearQuest is integrated with Rational testing tools so testers can easily enter defects and associate them with test descriptions and results.

Rational TestManager helps you share test results with other team members so:

- Analysts can track which requirements have been tested, and which tests pass or fail.
- Developers can track which Rose model elements have been tested, and which tests pass or fail.
- Managers and all team members can track the progress of testing and evaluate test results to assess quality.

Keeping the Team Up to Date

Rational ProjectConsole collects and stores data from Rational and some third-party development tools. It presents the results as reports and metrics on a common team Web site, ensuring that the entire team has access to current information. Project Console can be configured so each team member has quick access to only that information required to perform his or her specific role.

Rational SoDA generates project reports with data extracted from Rational tools, such as Rational RequisitePro and ClearQuest, so your team can communicate more efficiently and consistently. SoDA provides templates in Microsoft Word for Windows or Adobe FrameMaker for UNIX. You can customize templates or create new ones using ProjectConsole Template Builder.

Testers use ProjectConsole and SoDA to:

- Learn about requirements.
- Understand the system design and the components.
- Monitor project status.
- Communicate testing progress.
- Evaluate test results.

Managing Change and Risk: The Manager and Project Leader

5

Do you manage software changes or do they happen haphazardly?

How well is the team meeting your customers' needs?

Is the project behind schedule and over budget?

How Well Do You Plan?

When beginning a project, managers and project leaders develop project plans that address schedule, resources, responsibilities, and budget.

To manage projects effectively throughout development, managers and project leaders must also:

- Close-out project phases.
- Manage and plan for iterations.
- Conceive new projects.
- Evaluate project scope and risk.

Rational Suite Enterprise: The Complete Solution for Iterative Development

Rational Suite Enterprise offers the most complete set of project management and software development tools to support each role in your team and each phase of your project.

Implementing Best Practices

Project leaders and mangers guide their teams in following software development best practices by using Rational Suite Team Unifying tools. For example, use:

- The RUP to develop and implement a process so that your team can develop software iteratively.
- RequisitePro to manage requirements.

- Rose to develop component-based architectures and model visually.
- TestManager to continuously verify quality.
- ClearCase LT, ClearQuest, and UCM to manage change.

Managing Change with UCM

UCM helps project leaders trace activities from their initial change request through final software release, while maintaining the integrity of related project assets. Successful UCM includes:

- Change request management.
- Configuration management.

Change Request Management

Rational ClearQuest tracks defects, new features, enhancements, and change requests. It helps project leaders communicate with team members to understand the impact of change requests. It also generates charts and reports so you can:

- Analyze trends in change requests.
- Assess the workload of team members.
- Develop a plan for the next iteration.

Configuration Management

Rational ClearCase LT and UCM help project leaders accelerate development cycles, ensure accuracy of releases, and organize an effective distributed development process. It lets you organize and manage versions, releases, and parallel development of multiple releases and products. For example:

- 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, change 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

Unified Software Project Management (USPM) is the Rational approach to managing software projects from requirements to release. USPM focuses on compiling information to help assess project:

- Status.
- Trends.
- Quality.

Rational ProjectConsole collects and stores data from Rational and some third-party development tools. It presents the results as reports and metrics on a common team Web site, ensuring that the entire team has access to current information.

Project leaders and managers use ProjectConsole to:

- Collect objective metrics that lead to more accurate project assessments.
- Anticipate problems and delays, and make informed decisions to avoid them.
- Predict which areas of development require special attention and where to focus resources to stay on schedule.

Rational SoDA generates project reports with data extracted from Rational tools, such as Rational RequisitePro and ClearQuest, so your team can communicate more efficiently and consistently.

Project leaders and managers use SoDA to generate reports that help:

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

Next Steps 6

Implementing Rational Suite

Your team can start using Rational Suite all at once or incrementally. To deploy the Suite efficiently and effectively, you need to consider:

- Your hardware and software environment.
- Rational Suite installation requirements.
- An implementation strategy that includes training and an evaluation of the deployment sometime after initial implementation.

Adopting All of Rational Suite

If you decide to use an entire Rational Suite tool set in your next project, the RUP and the Rational Developer Network can help.

Use the RUP to help your team incorporate best practices into their work:

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

The Rational Process Workbench, a component of the RUP, contains extensive guidance on deploying the RUP in organizations, as well as tools for adding and modifying RUP content. It is available from the Rational Developer Network.

You can also use the Rational Developer Network to help your team speed projects to completion and expand professional skills. The Rational Developer Network offers:

- 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.

If you are interested in Rational Professional Services to help you deploy Rational Suite, see page 18.

Adopting Rational Suite Gradually

Rational Suite can also help your team work more effectively when you deploy Suite tools more gradually. For example, your team may decide to focus on requirements management. Analysts and the project leader learn to use RequisitePro and Rose, integrating some Rational Unified Process guidelines about requirements analysis into the development process.

Before you begin your next project, you should:

1. Re-evaluate your development process.
2. Prioritize problems.
3. Decide which ones to tackle first.
4. Select the Rational Suite tools that are best suited to address these problems.
5. Make use of the RUP, the Rational Developer Network, and Rational Professional Services to learn how to use Suite tools.

Exploring Rational Suite Tools

The *Rational Suite Tutorial* (for Windows only) teaches you the basics of using Rational Suite to plan, design, implement, and test applications.

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

1. If Rational Suite is installed on your system click **Start > Programs > Rational Software > Rational Suite Tutorial**. Follow the instructions to install the tutorial application files and documentation.
2. If Rational Suite is *not* installed on your system 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.

Note: Rational Suite must be installed on your system to perform the tutorial exercises.

Other Rational Tools

Rational ClearCase

Rational ClearCase is the configuration management solution for large projects and teams. In addition to ClearCase LT features, it offers:

- Advanced build management.
- Distributed server support.
- Automatic data replication.

To learn more, go to http://www.rational.com/products/clearcase.

Rational XDE

Rational XDE is an extended development environment that provides visual design and development capabilities. Integrated with Microsoft Visual Studio .NET and IBM WebSphere, XDE gives developers a single user experience. It can also be used alone through the Rational supported Java platform Integrated Development Environment (IDE), based on the Eclipse platform.

To learn more, go to http://www.rational.com/products/xde.

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. Agent computers are primarily used for performance testing using Rational Suite TestStudio, TeamTest, or Robot.

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. It 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.

metric. A measurement 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 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. Replicates a centralized database to each remote site and synchronizes 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 to all Rational customers.

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

Rational ProjectConsole Template Builder. A ProjectConsole plug-in that lets you build templates to automate the collection of project artifacts for display on your ProjectConsole Web site.

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 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 perform all major RequisitePro operations.

Rational Robot. Used to develop automated functional and performance test scripts to be run using TestManager.

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 SoDA. Software Documentation Automation – Produces reports from data in one or more Rational tools. Works with built-in templates or ones that you customize.

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 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 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.

Rational Unified Process (RUP). A flexible software development process platform that enhances team productivity and delivers software best practices using guidelines, templates, and Tool Mentors for software development activities.

Rational XDE. eXtended Development Environment – Incorporates Rational design and development tools into a single environment for both the .NET and Java platforms.

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. Identifying, anticipating, and addressing project risks and devising plans for risk mitigation, as a way of ensuring 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. 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. The 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