

Introducing Rational Suite

VERSION: 2001.03.00

PART NUMBER: 800-023717-000

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

Rational®
the e-development company™

IMPORTANT NOTICE

COPYRIGHT

Copyright ©1998-2000, Rational Software Corporation. All rights reserved.

Part Number: 800-023717-000

Version Number: 2001.03.00

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, the Rational logo, Rational the e-development company, ClearCase, ClearCase Attache, ClearCase MultiSite, ClearDDTS, ClearQuest, DDTS, Object Testing, Object-Oriented Recording, Objectory, PerformanceStudio, PureCoverage, PureDDTS, PureLink, Purify, Purify'd, Quantify, Rational Apex, Rational CRC, Rational Rose, Rational Summit, Rational Visual Test, Requisite, RequisitePro, SiteCheck, SoDA, TestMate, and The Rational Watch are trademarks or registered trademarks of Rational Software Corporation in the United States and 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, Active Accessibility, Active Channel, Active Client, Active Desktop, Active Directory, ActiveMovie, Active Platform, ActiveStore, ActiveSync, ActiveX, Ask Maxwell, Authenticode, AutoSum, BackOffice, the BackOffice logo, BizTalk, Bookshelf, Chromeffects, Clearlead, ClearType, CodeView, Computing Central, DataTips, Developer Studio, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, DirectXJ, DoubleSpace, DriveSpace, FoxPro, FrontPage, Funstone, IntelliEye, the IntelliEye logo, IntelliMirror, IntelliSense, J/Direct, JScript, LineShare, Liquid Motion, the Microsoft eMbedded Visual Tools logo, the Microsoft Internet Explorer logo, the Microsoft Office Compatible logo, Microsoft Press, the Microsoft Press logo, Microsoft QuickBasic, MS-DOS, MSDN, Natural, NetMeeting, NetShow, the Office logo, One Thumb, OpenType, Outlook, PhotoDraw, PivotChart, PivotTable, PowerPoint, QuickAssembler, QuickShelf, Realmation, RelayOne, Rushmore, SourceSafe, TipWizard, TrueImage, TutorAssist, V-Chat, VideoFlash, Virtual Basic, the Virtual Basic logo, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, the Visual Studio logo, Vizact, WebBot, WebPIP, Win32, Win32s, Win64, Windows, the Windows CE logo, the Windows logo, Windows NT, the Windows Start logo, and XENIX are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. 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 ©1992-2000, 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

- Preface** **vii**
 - Audiencevii
 - Other Resourcesvii
 - Rational Suite Documentation Roadmap viii
 - Contacting Rational Technical Support ix
- 1 Welcome to Rational Suite** **1**
 - The Challenges 1
 - Meeting the Challenges 1
 - About This Book 2
 - What is Rational Suite? 2
 - How Can Rational Suite Help You? 3
 - Unify Your Team 3
 - Optimize Your Team's Productivity 4
 - Simplify the Solution for Your Team's Development Needs 5
 - Incorporate Best Practices into Your Team's Process 5
 - Adopting these Practices 8
 - Customizing the Rational Unified Process 9
 - What's Next? 9
- 2 Defining the Right System: The Analyst** **11**
 - Understand the Problem Before You Invest 11
 - Iterative Development and the Cycle of Changing Requirements 11
 - Capture and Manage Changing Requirements with Rational Suite AnalystStudio. . 12
 - Defining Requirements for Your Team 13
 - Managing Change for Your Project 13
 - Helping Your Team Communicate Visually 15
 - Providing Project Reports 15
 - Capitalizing on the Power of Use Cases to Test all Dimensions of Quality . . . 15
 - Optimizing Your Work with the Rational Synchronizer 16
 - Summary 16

3	Managing Complexity: The Developer	17
	Managing a Complex Process	17
	Design, Coding, Unit Testing: The Next Steps in Iterative Development	17
	Creating Resilient Component-Based Architectures with Rational Suite	
	DevelopmentStudio	18
	Communicating Visually with Models of Your System	18
	Accelerating Code Implementation	19
	Keeping Code and Models Consistent	19
	Evaluating Changing Requirements	19
	Validating Changes to the System	19
	Managing Changes to the System	20
	Keeping the Team Up to Date	20
	Testing Code Early and Often	21
	Tracking Test Results	21
	Optimizing Your Work with the Rational Synchronizer	22
	Creating Component-Based Executable Architectures with Rational Suite	
	DevelopmentStudio – RealTime Edition for Windows	22
	Building Complex, Real-Time Systems	22
	Working the Way Real-Time Systems Operate	23
	Summary	23
4	Deciding to Release: The Tester	25
	Making the Crucial Decision to Release	25
	Subsystem and System Tests: The Last Step in Iterative Development	25
	Verifying Software Quality with Rational Suite TestStudio	26
	Unifying the Team by Keeping Members Informed	26
	Making a Plan and Measuring Progress	26
	Does Your Application Meet Requirements?	27
	Is Your Application Reliable?	27
	Does Your Application Have Memory Leaks?	28
	Does Your Application Perform Fast Enough?	28
	Does Your System Perform Under Production Load?	28
	Optimizing Defect Tracking	28
	Summary	29

5	Managing Change and Risk: The Project Leader	31
	How Well Can You Plan Ahead?	31
	Rational Suite: The Complete Solution for Iterative Development	31
	Developing Complex Products Using Unified Change Management	33
	Change Request Management	34
	Configuration Management	34
	Project Status and Measurement.	35
	Optimizing Your Work with the Rational Synchronizer	35
	Summary	35
6	Next Steps	37
	Adopting Practices and Tools	37
	Adopting all of Rational Suite	37
	Adopting Rational Suite Gradually	38
	Exploring the Tools and the Rational Unified Process	38
	Using the Documentation Set	38
	Contacting Rational's Professional Services	39
	Assessment Services.	39
	Rational University	39
	Technical Support	40
	Technical Resources	40
	Additional Resources	40
	Rational Suite: Summary	41
	Glossary	43
	Index	47

Preface

This manual provides an introduction to Rational Suite. Rational Suite delivers a comprehensive set of integrated tools that embody software engineering best practices and span the entire software development life cycle. Rational Suite's unparalleled level of integration improves communication both within teams and across team boundaries, accelerating development time and improving software quality.

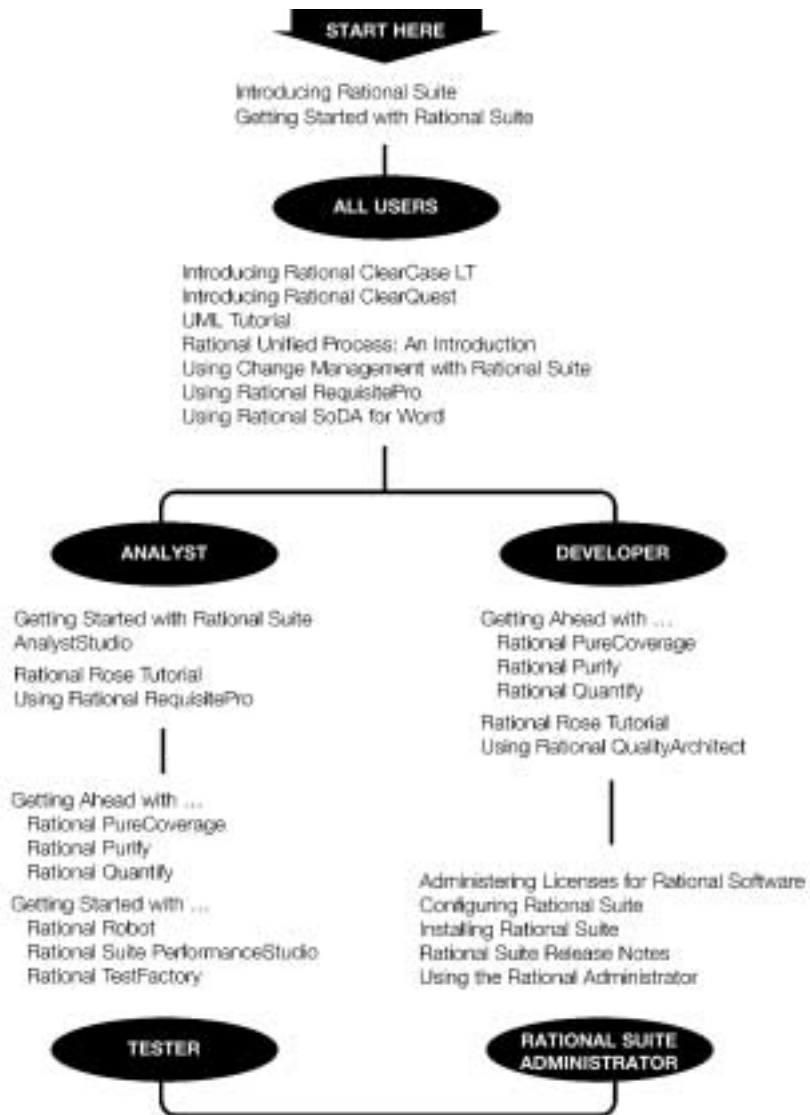
Audience

This guide is intended for all readers, including managers, project leaders, analysts, developers, and testers.

Other Resources

- Online Help is available for Rational Suite.
From a Suite tool, select an option from the **Help** menu.
- 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.
- 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>.

Rational Suite Documentation Roadmap



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) 20-4545-201 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, telephone number, and company name
- Your computer's make and model
- Your operating system and version number
- Product release number and serial number
- Your case ID number (if you are following up on a previously-reported problem)

As a software professional, you face growing demands for larger, faster, and more complex software systems and applications. At the same time, you are probably asked to deliver software on increasingly shorter schedules. This is especially true in organizations that develop e-commerce applications, internet infrastructure software, and devices that connect to the internet.

The Challenges

In your job, you encounter many challenges to delivering software on target and on time. Think of your last project as you read the following questions:

- Did your last project cost more than the business budgeted?
- Was your last project late, compromising the business activities it was meant to support?
- Was the quality of your last project less than satisfactory?
- Did your customer see inconsistent or unpredictable behavior, software crashes, confusing user interfaces, or slow performance?
- Did your project team fail to deliver a product at all?

Meeting the Challenges

To meet these challenges, your team must effectively manage communication, change, and risk. When your team can manage these key project elements well, you and your customers benefit in the following ways:

- **Manage communication.** When team members communicate well, they understand their customers' needs and objectives. They can more effectively gather requirements and correct mistakes early in the project. Your project team then releases a product that customers can quickly start using.

- **Manage change.** Tracking changed requirements and enhancements allows team members to easily reconstruct who changed what, when, where, and why. Managing requirements more accurately and efficiently helps your project team deliver a product that meets customer expectations. The team builds a product that solves the right problem at the right time. Effectively managing change throughout the project lifecycle helps the team deliver a project on time and within budget.
- **Manage risk.** If your project team implements and tests the most risky requirements first, it can correct serious project flaws early in the development cycle. Your group then delivers a quality product within schedule.

About This Book

Read this book to learn:

- How Rational Suite can help your team meet the challenges of developing quality software.
- How Rational Suite can help team members work more efficiently and communicate more effectively.
- How Rational Suite can help you plan for and manage change throughout the software development lifecycle.
- Where to find more information about how your team can use Rational Suite and Rational Software's comprehensive customer services.

What is Rational Suite?

Rational Suite is a family of integrated software development tools bundled as Suite editions for team members. Rational Suite products support the major functions of the development team, including: system definition, software development, content management, and system testing. The Suite provides tools that address individual productivity, team communication, and cross-functional team productivity. Rational Suite helps you meet the challenges your team faces in designing, developing, and testing and deploying e-commerce applications, internet infrastructure software, e-devices, and other software systems and applications.

Each Suite edition of Rational Suite is optimized to support a team role: analyst, architect and developer, and tester. 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

Rational Suite Team Unifying Platform, and focuses on one or more product domains: requirements, modeling, testing, and change management. All Suite editions include tools from the Team Unifying Platform (Rational Unified Process, Rational RequisitePro, Rational ClearQuest, Rational SoDA, Rational TestManager, and Rational ClearCase LT). These tools help team members obtain the information they need to manage change and control risk associated with all phases of software development.

Rational Suite works with popular Integrated Development Environments (IDEs) such as Microsoft Visual Studio and IBM VisualAge for Java. On Solaris and HP-UX platforms, Rational Suite works with IDEs such as HP Workbench and Sun WorkShop. It also works with reporting and management tools (Crystal Reports and Microsoft Project) and authoring tools (Microsoft Word and Adobe FrameMaker).

Rational Suite supports both the IBM Application Framework and the Microsoft Windows Distributed Internet Architecture to develop database management systems for the internet. It also supports the prevailing component models, COM and Enterprise JavaBeans, along with widely-used Web servers, including Microsoft IIS, and Web Application Servers, such as IBM WebSphere.

How Can Rational Suite Help You?

Unify Your Team

The Rational Suite family of products is customized into editions to support the major roles on your development team: analyst, developer and architect, and tester. **Rational Suite Team Unifying Platform** supports project leaders in their role in the project lifecycle. The Suite edition unifies software development teams to maximize productivity and quality. It provides best practices and integrated tools for managing change, building quality, and communicating results from requirements to release. These tools include:

- **Rational Unified Process**—Unifies best practices from several software development disciplines into a consistent framework covering the full lifecycle.
- **Rational RequisitePro**—Helps you manage requirements and make them accessible so that team members better understand and more effectively perform their part in the project throughout the lifecycle.
- **Rational ClearQuest**—Lets you track, manage, and report on product defects and change requests.

- **Rational ClearCase LT**—Helps all team members manage change. ClearCase LT helps you track changes to all files related to software development, including requirements, visual models, tests, and code.

Note: Rational ClearCase LT is not included inside any UNIX Suite.

- **Rational SoDA (for Microsoft Word or Adobe FrameMaker)**—Allows you to create documents and helps to keep project documentation current by extracting information directly from application databases and files.
- **Rational TestManager**—Helps you track software testing information through all phases and iterations of the software development, test, and revision cycles.

Optimize Your Team's Productivity

Rational Suite editions are optimized to enhance the productivity of each member of your team.

Rational Suite AnalystStudio is customized for development professionals who gather and manage project requirements.

Rational Suite DevelopmentStudio (Windows and UNIX) is tailored for software architects, designers, and developers who want to test the architecture early in the development process and throughout the system's evolution.

Rational Suite DevelopmentStudio – RealTime Edition is tailored for software architects, designers, and developers of real-time embedded software. This Suite provides an integrated set of tools to optimize the development of complex real-time embedded software for e-devices, such as cell phones and pagers, and e-infrastructure software for the routers and hubs that connect and power the internet.

Rational Suite TestStudio is designed for team members who are responsible for software quality and load testing.

Rational Suite Enterprise Edition includes all of the tools from Rational Suite AnalystStudio, DevelopmentStudio, and TestStudio, providing a comprehensive tool set for the entire team.

Rational Suite Team Unifying Platform is useful to 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. For example, project and program managers, project administrators, and development managers use the tools in this Suite.

Simplify the Solution for Your Team's Development Needs

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

- one integrated tool set that is tested and updated together so you don't have to integrate tools yourself.
- one installation program that allows your team to install all the tools at once or one tool at a time. This flexibility is especially helpful if your team has a need it must address immediately.
- one location for online Help, Rational Suite documentation, and Rational Technical Support.

Incorporate Best Practices into Your Team's Process

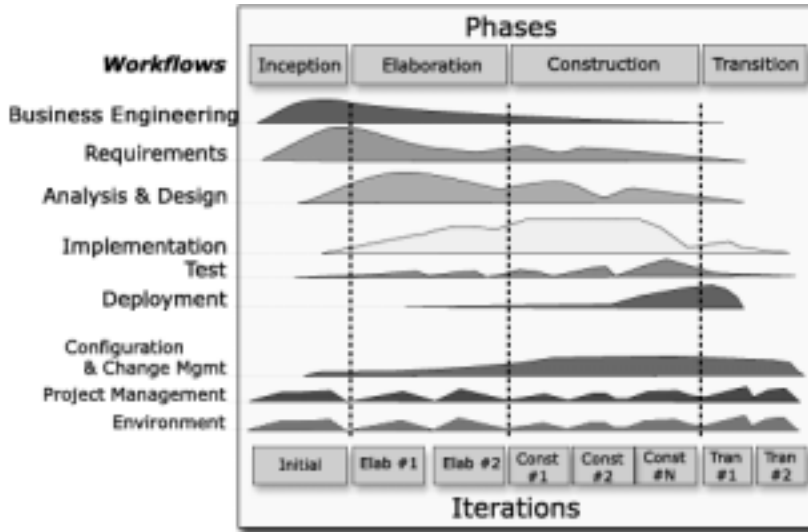
One of the first steps when solving common development problems is to evaluate your team's software development process. Rational advocates following a set of software-engineering practices based on its own experience as well as the experience of its many customers. All Rational Suite editions are designed to support these six best practices:

Develop software iteratively. Iterative development means developing the software in planned increments. 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.

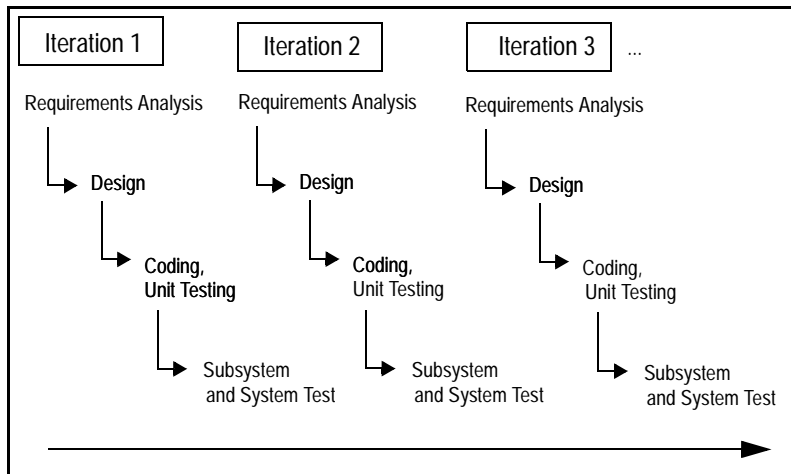
In the project lifecycle, there are multiple iterations. Each iteration focuses on one of these development phases:

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

The following diagram represents the full development lifecycle. Notice the multiple iterations within each phase. The height of each workflow shows when the work becomes more intense for each phase in the project.



The following diagram illustrates the steps in an iterative process: Requirements Analysis, Design, Coding, and Unit Testing, and Subsystem and System Testing.



The steps in an iterative process correlate with the responsibilities of specific team roles: analyst, architect and developer, and tester. 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, the analyst tracks requirements throughout the iteration and throughout the project lifecycle.

Manage requirements. Requirements define what the customer and other stakeholders need. Other sources of requirements are your organization's competitive environment, legal constraints, and performance criteria. Requirements change throughout the project. Managing requirements involves evaluating these changes and determining their impact.

Use component-based architectures. A component is a non-trivial, nearly independent, part of a system that combines data and functions to fulfill a clear purpose. You can build components from scratch, re-use components you have previously built, or even purchase components from other companies. These components form the architecture – the fundamental framework – for your software project.

Visually model software. Visual models improve your team's ability to manage and clearly communicate software complexity. Creating a graphical blueprint of the system's architecture conveys complex information in a common language that all members can understand. A visual model helps team members detect inconsistencies and mistakes in the product architecture. From a model, you can quickly understand the impact of changes over the course of a project.

Continuously verify quality. Verifying quality means testing how well the project meets its requirements. It also means testing reliability and performance. With iterative development, you can start testing early. This early testing enables you to discover and resolve problems as early as possible in the development cycle, where they are least costly.

Manage change. It is important to manage changes in a trackable, repeatable, and predictable manner. Enhancement requests and defect reports help facilitate clear communication among team members. Your team should define repeatable processes for changing requirements and code.

Adopting these Practices

Rational has developed the Rational Unified Process, a Web-enabled, searchable knowledge base that helps you analyze your own development practices and helps you incorporate these software engineering practices into your development plan. The Rational Unified Process is a team-unifying tool. It is included in every Rational Suite edition.



The Rational Unified Process provides the following links between process guidelines and the tools contained in Rational Suite:

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

Customizing the Rational Unified Process

The Rational Unified Process is not a static resource. You can easily customize the Process by adding information from your project, such as guidelines and standards. Moreover, you do not have to take an all-or-nothing approach to the Rational Unified Process. You can start by using a subset of the Process, perhaps to address an area that presents difficulties to your team. When you are ready to tackle another area of difficulty, you can incorporate another part of Rational Unified Process into your development practices.

What's Next?

The remainder of this book:

- Illustrates how the analyst, developer and architect, tester, and project leader benefit by using the Rational Suite.
- Shows how the tools fit into an effective iterative process by focusing on requirements management, design, coding and unit testing, subsystem and system testing, and unified change management.
- Directs you to more information about Rational's 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 and/or behind schedule?

Are your users dissatisfied with your product?

Understand the Problem Before You Invest

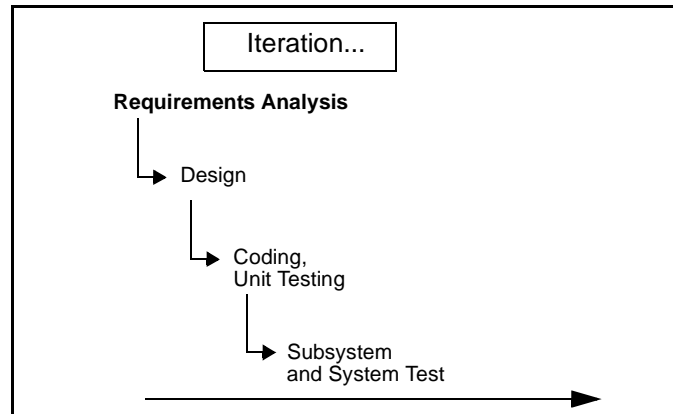
As the analyst, you interpret requests from *stakeholders* and determine *what* the system will do. You represent the customer by capturing, organizing, and managing requirements. Upon identifying the problem your organization will address, you need to define the right system, communicating that vision to the rest of the team.

In an effort to solve the right problem and define the right system, it is essential that you help all stakeholders clearly understand the problem space and solution requirements. Even as the development objectives and requirements change during the project, it is critical to maintain open communication and a shared understanding of the requirements.

Iterative Development and the Cycle of Changing Requirements

In iterative development, your project team plans, develops, and tests one subset of system functionality at a time. Each iteration results in either an internal or external release. Subsequently, your team develops the next increment of functionality, integrates it with the first release, and the cycle continues.

At the start of an iteration, you prioritize, schedule, and assign responsibilities for requirements. Throughout the iteration, you track these requirements and evaluate how well the team has met them.



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 these new features 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 the requirement.

Capture and Manage Changing 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 the analyst capture, manage, and articulate requirements in a form accessible to all stakeholders. It combines the power of a relational database with the freedom of Microsoft Word.

You, as the analyst, 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's 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 management of the requirements. The database allows you to 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 needs to 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 capability enables UNIX users and other team members who do not have RequisitePro on their desktops to 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, and visual models, as the system evolves. As an analyst, you can use ClearCase LT to manage changes to your project's requirements. The tool allows you to associate requirements with releases and other project assets. ClearCase LT also enables you to archive all requirements. ClearCase LT 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, as well.

Your team and other stakeholders can evaluate requests, determine their impact on the system, and when applicable, validate the changes. To establish how change request fit into the structure of features and main requirements, you can link requests to an existing or new project requirement in Rational RequisitePro. Later on, this feature can aid your team in risk management when, for example, you are considering elimination of a system feature that originated as an enhancement request from your most important customer.

ClearQuest Change Request



Requisite Pro Feature Tree

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 to 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's 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 Unified Modeling Language (UML) - an easily understood, industry standard language for designing software. For example, analysts use Rose to describe a system at a high level and architects 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.

Providing Project Reports

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's reporting features provide templates in either Microsoft Word for Windows or Adobe FrameMaker on UNIX. You can easily customize these templates or create new ones.

Capitalizing on the Power of Use Cases to Test all Dimensions of Quality

Rational TestManager allows testers to manage test planning, design, development, execution, and analysis and share these assets with your 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 allows you to build test assets from this foundation of use cases. In addition, all test assets, including test requirements, test plans, testing procedures, manual and automated tests, and test results are stored in a central location using TestManager. These artifacts are stored using a single tracking system and a single interface accessible to all team members.

Optimizing Your Work with the Rational Synchronizer

The **Rational Synchronizer** automatically creates items in your project based on the existence and status of related items. This tool helps your team manage project elements by ensuring that all the relationships between the items remain intact and valid. For example, in Rational Suite AnalystStudio, once you've captured feature and use case requirements in the RequisitePro database, you can use the Synchronizer to generate the related test requirements. You can also use the Synchronizer to generate a use case diagram in Rose.

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's 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 properly?
- 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?

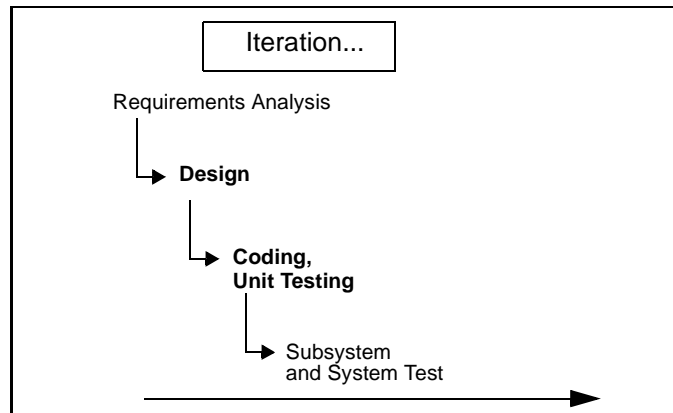
Managing a Complex Process

Architects and developers define *how* the system works. As an architect, 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 can be quickly and easily modified as your project's requirements change. As the developer, you create, modify, and manage code for the system based on this model.

Design, Coding, Unit Testing: The Next Steps in Iterative Development

As the architect, 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 working with component-based architectures, your project 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 allows you to discover and respond to problems early enough to minimize their impact on the project.

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, multi-user tests of system performance, this Suite helps your team design and build the right project the first time. You can use these tools to test performance as soon as your system architecture is designed.

Rational Suite DevelopmentStudio – RealTime Edition (Windows only) is further tailored 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 for Windows* on page 22.

Communicating Visually with Models of Your System

Rational Rose (Enterprise 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. 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 Unified Modeling Language, 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 the coding process 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, IBM Visual Age for Java, and IDE's including HP Workbench and Sun Workshop.

Keeping Code and Models Consistent

Once you modify your code, Rose allows you to bring code changes into your model, ensuring that your model and code remain consistent throughout the project. This process is called *reverse engineering*.

Keeping your code and model consistent helps you clearly see the impact of changes on the system architecture throughout the development lifecycle. It is important to assess these changes early because they may violate project standards, 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 enables all team members to review and update requirements.

When you, the architect, 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, and you are able to communicate these changes and impacts to your team members, successfully unifying developers, analysts, testers, 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 review ideas and provide feedback.

Using Rose models, you can visualize the impact of these requests on the system architecture.

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. It helps development teams accelerate build cycles by ensuring accuracy of releases and organizing an effective development process. Furthermore, ClearCase LT optimizes individual productivity because you can use it 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.

Optionally, you can use Unified Change Management (UCM), Rational's built-in 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 undergoing change throughout the lifecycle of the software project.

UCM raises the level of abstraction when working with software configuration management and change management tools, like ClearQuest and ClearCase LT. For example, a project leader uses ClearQuest to identify which activities to assign to each developer. ClearCase LT maintains a *change set* - a list of changed artifacts - associated with each activity. When you finish working on an activity, you *deliver* the work to a shared workspace. Within this *integration 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: Rational ClearCase LT is not included in Rational DevelopmentStudio for UNIX.

Keeping the Team Up to Date

Rational SoDA (for Microsoft Word or for Adobe FrameMaker) allows you to 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 Rose automates the mechanical aspects of test code creation by generating test code from visual models. This allows developers to 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.

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 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 executed. This information allows you to determine if your tests have actually executed 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. To effectively accomplish this, Rational Suite provides tools to use throughout the development lifecycle, allowing you to track and report change requests, test your models and code more often and more completely, keep abreast of project status, and communicate your results with the rest of your team.

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. With this tool, team members can evaluate how well they are meeting project requirements from early on in the development lifecycle, and communicate these findings to project stakeholders.

Rational ClearQuest tracks the defects that you find in your software project. Rational testing tools are integrated with ClearQuest to simplify the process of entering defect information. ClearQuest provides **ClearQuest Web**, enabling all team members to review and update defects. This tool tracks the defect's history and provides a description and other details about the bug.

Note: Rational TestManager is not included in Rational DevelopmentStudio for UNIX.

Optimizing Your Work with the Rational Synchronizer

The **Rational Synchronizer** automatically creates items in your project, based on the existence and status of related items. This tool helps your team manage project elements by ensuring that all related items exist at the right time and that none of them are lost. For example, in Rational Suite DevelopmentStudio, once you've created Rational Rose diagrams that model specific interactions in the system, the Synchronizer can create requirements in Rational RequisitePro that correspond to those diagrams.

Note: Rational Synchronizer is not included in Rational DevelopmentStudio for UNIX.

Creating Component-Based Executable Architectures with Rational Suite DevelopmentStudio – RealTime Edition for Windows

Rational Suite DevelopmentStudio – RealTime Edition (Windows only) is tailored 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 or C++ 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 allows you to build real-time systems the way real-time systems operate. Rose RealTime:

- Uses the UML and a UML profile specialized for real-time systems to represent all the 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 or C++ applications from UML models.
- Supports multiple Real-Time operating systems out of the box.

Summary

- Rational Suite DevelopmentStudio (Windows or UNIX) offers an integrated tool set to optimize the design, coding, and unit testing of your software development project.
- Rational Suite DevelopmentStudio – RealTime Edition for Windows offers an integrated tool set to optimize the definition, design, application generation, and testing of your real-time, embedded development project.
- Rational Suite DevelopmentStudio allows you to 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 which enables you to easily keep your models and your code in sync through 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?

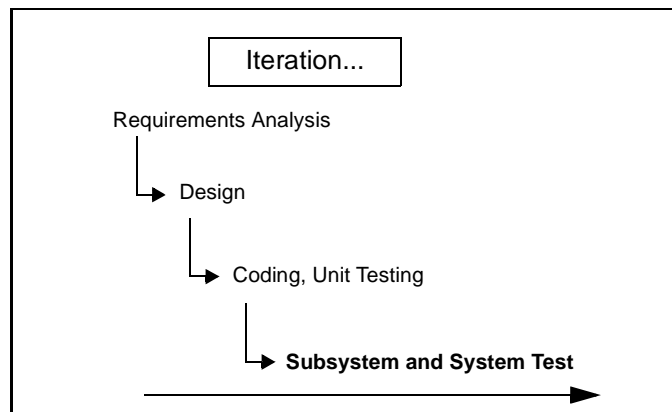
How often do you have a complete picture of your software's readiness before you release?

Making the Crucial Decision to Release

As the tester, you ensure that software meets its requirements and is stable. You look for application defects and pinpoint performance problems in each iteration.

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.



Over the course of the project, you manage and track use cases, requirements, and code which have been tested. 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.

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 an increasingly important 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, the 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. RequisitePro’s Web interface, **RequisiteWeb**, enables team members and customers who do not have RequisitePro on their desktops to review and update requirements.

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

SoDA’s 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.

Making a Plan and Measuring Progress

Rational TestManager allows you to use all types of project artifacts to plan, design, and execute tests. Requirements, visual models, and source code can be used to create a test plan so that all aspects of your system can be tested, including product features, system architecture, and code. With TestManager, artifacts used your test plan are used to create a specific test asset. When the original artifact changes, you will automatically be prompted to re-evaluate the related test asset. You can then decide to update the test artifact, if necessary.

This verifiable relationship between project assets allows you to test for quality early in the development lifecycle, often beginning with product features, and continue through implementation and release.

Rational TestManager also provides access to all test-related information and artifacts so your team can easily obtain an objective assessment of project status. Using development tools like Rational Rose and Rational Robot, TestManager enables team members to share development progress on early and iterative testing activities. TestManager also helps you track how many tests have been planned, scripted, and run. It helps you follow which requirements have been covered, and the number of tests that have passed and failed.

As part of each Studio and Suite edition, team members can use TestManager to evaluate how well they are meeting project requirements, monitor the project's overall status, and more effectively share and discuss information about testing activities with other project stakeholders.

Does Your Application Meet Requirements?

Automated Functional Testing—Rational Robot determines whether the system meets requirements by testing how it responds to user input. With Robot's fast and intuitive interface, you record a test, and then play it back as often as you need to.

Once 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. TestManager can also identify test inputs associated with a test case, creating a tangible correlation to and logical progression from the test results.

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.

Does Your Application Perform Fast Enough?

Automated Application Performance Testing—Rational Quantify detects performance bottlenecks, 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.

Automated Application Performance Testing—Rational PureCoverage checks every component in your program for lines of code that the system fails to exercise.

Rational Robot can rerun scripts using Rational Purify, PureCoverage, and Quantify to ensure all reliability and application performance tests are repeatable.

Does Your System Perform Under Production Load?

Automated System Performance Testing—Rational LoadTest allows you to run multi-user performance tests for e-business, multi-tier, 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 LoadTest runs these tests, it collects data that helps your team measure and predict your project's system performance.

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 changes in repair defects are properly verified. ClearQuest provides **ClearQuest Web**, a Web-based version of ClearQuest that enables all team members to review and update defects and change requests from any platform. ClearQuest Web provides the same features available on the Web that are available through the desktop client so that 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, configurations, and testers. With this tool, you can create reports to communicate project status to project stakeholders.
- 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.
- 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.

Managing Change and Risk: The Project Leader

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?

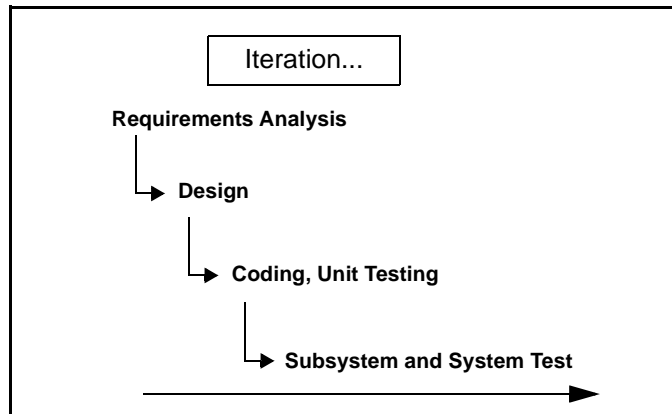
Is the project over budget?

How Well Can You Plan Ahead?

As the 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.

Rational Suite: The Complete Solution for Iterative Development

Each Studio edition of Rational Suite is optimized to support a major team role: analyst, architect, developer, and tester. Each Studio 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 focuses on one or more product domains: requirements, modeling, testing, and change management. All Studio editions include the Rational Team Unifying Platform (Rational Unified Process, Rational RequisitePro, Rational ClearQuest, Rational SoDA, Rational TestManager, and Rational ClearCase LT) that help 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 spans the functional boundaries 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 into the product in a controlled, iterative manner, you can collect feedback about the system. This early feedback, which includes defect reports, enables the team to change or adjust system architecture when it is easier and less expensive to do so.

The Rational Suite family of products supports effective software engineering practices. The Suite's 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 development 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 the analysts to prioritize, refine, and update requirements. In early iterations, your team negotiates guidelines with other stakeholders on how to review and evaluate enhancement 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 re-use 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 – blueprints – of the system architecture. The entire team uses visual models to gain a quick understanding of the system.
- **Verify software quality.** As early as possible in the development cycle, your team tests 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. Rational Suite provides your team with tools to track, organize, and validate these changes. For example, managing changes using enhancement requests and defect reports facilitates clear communication among team members. Rational Suite helps your team define and organize the procedures for changing requirements and code.

Developing Complex Products Using Unified Change Management

Unified Change Management (UCM) is Rational’s 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.
- **Project Status and Measurement.** Focuses on compiling information to assess status, trends, quality, and other aspects critical to project management and articulation of progress.

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 enables your team to 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 and ClearQuest. When 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.
- Once complete, 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.
- Systems are tested and released.

Project Status and Measurement

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 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 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.

Optimizing Your Work with the Rational Synchronizer

The **Rational Synchronizer** automatically creates items in your project based on the existence and status of related items. This tool helps your team manage project elements by ensuring that all related items exist at the right time and that none of the relationships between them are lost.

Summary

- The Rational Suite family of products supports software development best practices and offers tools so that project leaders realize greater success in planning iterations, ensuring integrity of artifacts, and enhancing communication among stakeholders.
- Iterative development allows your team to 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 allows you to 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.
- SoDA allows you to extract and compile critical information so that you can articulate the project's status to stakeholders.

Rational Suite can help your project team manage communication, change, and risk, so your group can 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's integrated tools help team members work together more effectively by enhancing communication between the major team roles: analyst, architect, developer, tester, and project leader.
- 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.

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 can help. You can use the Rational Unified Process' 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:

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

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. Once 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 the Tools and the Rational Unified Process

To gain a basic understanding of how you can use Rational Suite to plan, design, implement, and test applications, start by reading *Getting Started with Rational Suite*, included in your documentation set. *Getting Started with Rational Suite* shows how software teams use Rational Suite to plan, design, implement, and test software applications. This book also points you to other information about Rational Suite, so that you can learn more on your own.

Using the Documentation Set

In addition to *Getting Started with Rational Suite*, each Rational Suite edition includes the following manuals:

- *Installing Rational Suite*—A guide to installing Rational Suite software.
- *Administering Licenses for Rational Software*—A guide to installing and configuring licenses for Rational Suite software.
- *Configuring Rational Suite*—A guide to Rational Suite for Windows tools and features that support administration, synchronization, and out-of-the-box functions.
- *Using the Rational Administrator*—A guide and reference to the primary administration program, Rational Administrator (Windows only).
- *Using Change Management with Rational Suite*—A guide to automating change throughout software development using Rational Suite for Windows tools.

- *Release Notes*—Updated technical information about Rational Suite products. Online release notes for Rational Suite DevelopmentStudio for UNIX are located in the `Install_Path/docs/file_type/relnote` directory, where *file_type* is `html`, `pdf`, or `ps` (postscript) depending on the file type you prefer to use.
- *Product Documentation*—Each Rational Suite edition provides documentation for the tools included with that edition.

The complete Rational Suite documentation set, available separately, provides comprehensive documentation on the tools included with Rational Suite. Electronic versions are available as follows:

- For Rational Suite for Windows, see the *Rational Solutions for Windows Online Documentation* CD, included with your Rational Suite media kit.
- For Rational Suite DevelopmentStudio for UNIX, see the `/docs` directory of the *Rational Suite DevelopmentStudio for UNIX* CD, included with your Rational Suite media kit.
- If you install Rational Suite DevelopmentStudio – RealTime Edition, online manuals for Rose RealTime are installed in `%ROSSERT_HOME%\help`.

For a complete printed documentation set, contact your Rational Sales office.

Contacting Rational's 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 on-site delivery of these courses. To find information about courses, schedules, and registration, see <http://www.rational.com/university>.

Technical Support

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

Technical Resources

You can use Rational's online resources to answer your support questions. 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/sitewide/support/resources.jttml>.

Additional Resources

Rational Software authors have written extensively about project management, application development, visual modeling, and other related topics. An extensive, annotated bibliography is available in the Rational Unified Process. You can also find current papers and presentations about the Rational Unified Process on Rational's Web site at <http://www.rational.com>. To learn more about the Unified Modeling Language, see Rational Software's UML Resource Center at <http://www.rational.com/uml>.

Rational Suite: Summary

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

Rational Tool	Analyst Studio	Development Studio (Windows or UNIX)	Development Studio - RealTime Edition	Test Studio	Team Unifying Platform	Enterprise
Rational Unified Process	X	X	X	X	X	X
Rational RequisitePro	X	X	X	X	X	X
Rational ClearQuest	X	X	X	X	X	X
Rational SoDA for Word (Windows) or for FrameMaker (UNIX)	X	X	X	X	X	X
Rational ClearCase LT	X	X (Windows)	X	X	X	X
Rational TestManager	X	X	X	X	X	X
Rational Rose	Data Modeler Edition	Enterprise Edition	RealTime Edition			Enterprise Edition
Rational Robot				X		X
Rational TestFactory				X		X
Rational PureCoverage		X	X	X		X
Rational Purify		X	X	X		X
Rational Quantify		X	X	X		X
Rational QualityArchitect		X (Windows)				X

Glossary

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

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.

component. A non-trivial, 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.

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 that provides links to the Rational Unified Process and any customized information you want to add.

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.

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. The 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 *Using the Rational Administrator*.

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

Rational ClearQuest. A highly customizable Windows and Web-based 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.

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

Rational Purify. Automatically pinpoints hard-to-find runtime memory errors in Windows NT 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, multi-user database.

Rational Robot. Helps with functional 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 SoDA for Word. 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. Edition of Rational Suite optimized for system definition. Contains the Team Unifying Platform –Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager – and Rational Rose (Professional Data Modeler Edition).

Rational Suite DevelopmentStudio. Edition of Rational Suite optimized for software development. Contains the Team Unifying Platform –Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager – plus Rational Rose (Enterprise Edition), Rational Purify, Rational Quantify, and Rational PureCoverage.

Rational Suite DevelopmentStudio - RealTime Edition. Edition of Rational Suite optimized for system developers and designers of real-time or embedded systems. Contains the Team Unifying Platform – Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager – plus 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 editions includes these Team Unifying Platform –Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager. All other Suite editions include the Team Unifying Platform.

Rational Suite TestStudio. Edition of Rational Suite optimized for test engineers. Contains the Team Unifying Platform –Rational Unified Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager – and Rational PureCoverage, Rational Purify, Rational Quantify, Rational Robot and Rational TestFactory.

Rational Synchronizer. Uses rules, either predefined or user-supplied, to give you a quick start on new work. Creates or updates project items based on the existence of other items in your project, ensuring that details do not fall through the cracks.

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

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

real-time application. 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 updating a Rose visual model from code, so that the visual model and code are kept in sync. 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's 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 engineer. A person who creates, manages, and executes tests; ensures that the software meets all its requirements; and reports the results and verifies fixes.

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). Rational's approach to managing change in software development, from requirements to release. UCM spans the development lifecycle, defining how to manage change 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. It simplifies software design, and communication about the design.

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 in a business that produces a valuable result to an individual actor in the business.

Index

A

activity 7, 13, 20, 27, 33, 34, 43
analyst 43
 definition 11
 tools 13, 14, 15
AnalystStudio 44
architect
 definition 17
 tools 18, 19, 20, 22
artifact 13, 20, 26, 33, 43
automated testing 43

B

baseline 43
bottlenecks, finding 21

C

change requests
 managing 33
ClearCase LT 13, 20, 34, 44
ClearQuest 14, 19, 34, 44
ClearQuest Web 22
code
 implementation 19
 models 19
 testing 21
component 43
component-based architecture 7, 43
component-based architectures 32
 Rational Suite DevelopmentStudio 18
configuration management 33, 43

D

defect tracking 28
developer 17, 43
 tools 15, 18, 19, 20, 22
DevelopmentStudio 44
DevelopmentStudio - RealTime Edition 18, 44
documentation for projects 38, 39

E

element 43
Extended Help 8, 37, 43

F

forward engineering 43

H

Help, Extended 8, 37

I

iterative development 5, 32, 43
 coding 17
 design 17
 requirements analysis 11
 subsystem tests 25
 system tests 25
 unit testing 17

L

load testing 28
LoadTest 28

M

manage risk 2
managing change 2, 7, 33
managing communication 1
managing requirements 7, 32
memory leaks
 finding 21
 Rational Purify 28
metrics 43

P

production load 28
professional services
 assessment 39
 Rational University 39
 technical resources 40
 technical support 40
project leader 31, 43
 definition 31
 tools 31, 35
project report 15
project status, measuring 33, 35
PureCoverage 21, 28, 44
Purify 21, 28, 44

Q

quality 7
Quantify 21, 28, 44

R

Rational Administrator 44
Rational ClearCase LT 4, 13, 20
 project leader 34
Rational ClearQuest 3
 analyst 14
 architect and developer 19, 22
 project leader 34
Rational LoadTest 28
Rational PureCoverage 21, 28
Rational Purify 21, 28
Rational Quantify 21, 28
Rational RequisitePro 3
 analyst 13
 architect and developer 19
 tester 26
Rational Robot 27
Rational Rose 15, 18, 21
Rational Rose RealTime 22
Rational SoDA 4, 20
 analyst 15
 project leader 35
 tester 26
Rational Suite 2, 44
 adopting 37, 38
 AnalystStudio 4, 12, 44
 DevelopmentStudio 4, 18, 44
 RealTime Edition (Windows only) 18
 Windows and UNIX 18
 Enterprise Edition 4, 44
 iterative process 31
 summary table 41
 Team Unifying Platform 3, 4, 44
 project leader 31
 TestStudio 4, 26, 45
Rational Synchronizer 45
 AnalystStudio 16
 DevelopmentStudio 22

Rational TestFactory 27, 45
Rational TestManager 4, 15, 21, 26
Rational Unified Process 3, 37, 45
 customizing 9
 tool 8
real-time application 45
requirement 7, 45
 managing 45
requirements
 managing 32
RequisitePro 13, 19, 26, 44
reverse engineering 45
risk 45
Robot 27, 44
Rose 15, 21, 44
Rose RealTime 22
round-trip engineering 45

S

SoDA 20, 26, 35, 44
software development
 best practices 5, 32
 adopting 8
 the challenges 1
software quality 33
software, visual modeling 7
stakeholder 11, 45

T

Team Unifying Platform 3, 31, 44
test engineer 45
tester 25
 tools 26, 27, 28
TestFactory 27, 45
testing 21, 25
 application performance 28
 automated 27
 functional 27
 reliability 27
 system performance 28
TestManager 15, 21, 26
TestStudio 26, 45
Tool Mentor 45
Tool Mentors 8, 37

U

Unified Change Management 33, 45
Unified Modeling Language 15, 19, 45

V

version control 46
visual modeling 33, 46

W

workflow 46

