Introducing Rational Suite®

VERSION: 2001A.04.00

PART NUMBER: 800-024442-000

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

Rational®
the e-development company™

IMPORTANT NOTICE

Part Number: 800-024442-000

Version Number: 2001A.04.00

Apache Software Foundation Notice

The Tomcat software, which is distributed with Rational RequisiteWeb, is protected by the Apache Software License, Version 1.1. Copyright © 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

4. The names "The Jakarta Project", "Tomcat", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

PATENT

Portions covered by U.S. Patent Nos.5,193,180 and 5,335,334 and 5,535,329 and 5,835,701 and 5,574,898 and 5,649,200 and 5,675,802.

U.S. Patents Pending.

International 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

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 lifecycle. Rational Suite's unparalleled level of integration improves communication both within teams and across team boundaries, reducing development time and improving software quality.

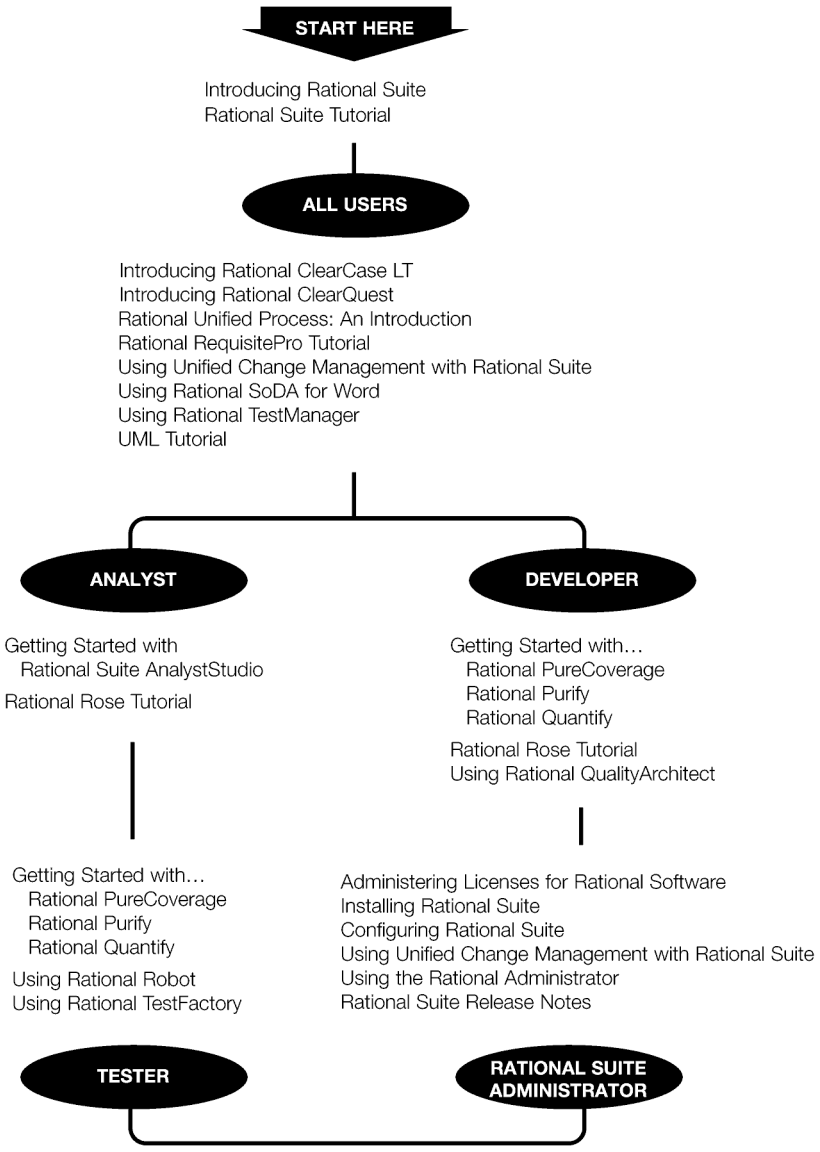
Audience

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

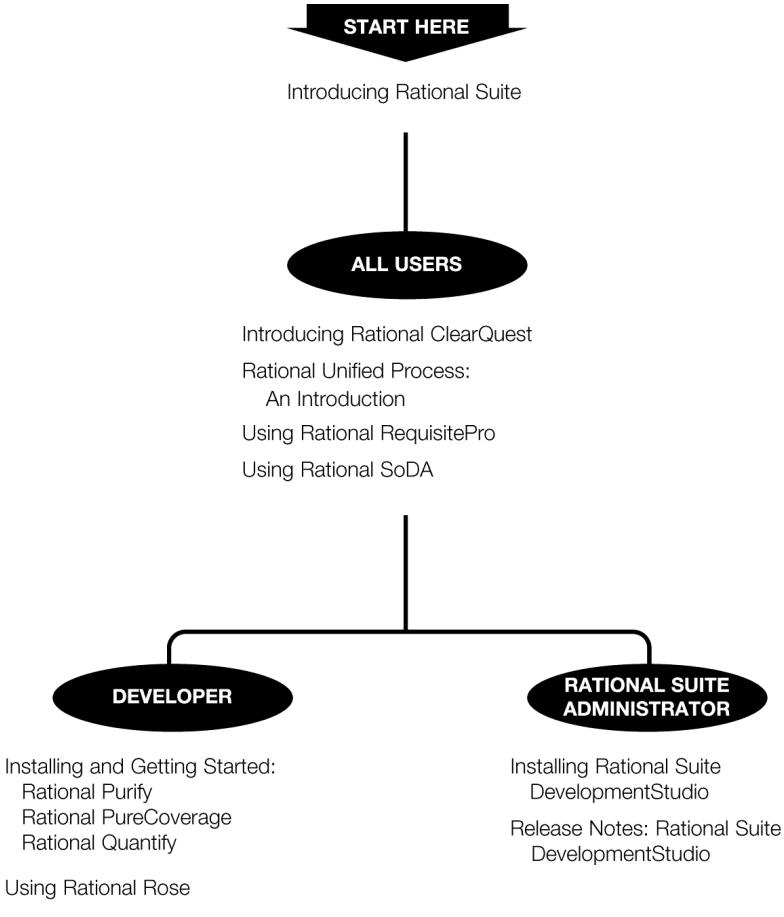
Other Resources

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

Rational Suite Documentation Roadmap - Windows



Rational Suite Documentation Roadmap - UNIX



Contacting Rational Technical Support

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

Your Location	Telephone	Facsimile	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4545-201 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, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your case ID number (if you are following up on a previously reported problem)

Welcome to Rational Suite

1

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

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

How can you make your next project more successful?

Principles of Software Development

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

- software engineering best practices
- integrated tools that automate these best practices
- professional services that accelerate the adoption and implementation of these best practices and tools.

Figure 1 Using Best Practices and Tools to Develop Better Software

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

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

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

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

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

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

A component is a nontrivial, independent, and replaceable part of a system that combines data and functions to fulfill a clear purpose. You can build components from scratch, reuse components you previously built, or even purchase components from other companies.

Designing a component-based architecture enables you to improve your project's predictability and helps enhance maintainability and extensibility.

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

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

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

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

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

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

Rational Suite Can Help

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

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

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

What's in Rational Suite?

Rational Suite editions are sets of tools customized for every member of your team. Each Suite edition contains the tools from the Rational Suite Team Unifying Platform. The Team Unifying Platform is a common set of tools that focus on helping your team perform more effectively. Each Rational Suite edition also contains tools selected for a specific practitioner on your development team. The following sections describe each Suite edition and the tools they contain.

Optimize Your Team's Productivity

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

Rational Suite Team Unifying Platform is a Suite edition, itself, and is contained in every Suite edition. It 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 edition.

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

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

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

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

Rational Suite ContentStudio is optimized for e-business application development teams. This Suite provides an integrated set of tools to unify code and content management for Web applications.

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

Tools That Unify Your Team

Rational Suite Team Unifying Platform

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

The Team Unifying Platform Suite edition contains the following tools:

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

Rational RequisitePro. Helps you organize, prioritize, track, and control changing project requirements. **RequisiteWeb** allows users to access, create, and manage requirements data from a Web browser.

Rational ClearQuest. Manages change activity associated with software development, including enhancement requests, defect reports, and documentation modifications. **ClearQuest Web** allows users to perform all major ClearQuest operations such as submitting records, finding records, creating or editing queries and reports, creating shortcuts, from a Web browser. **ClearQuest MultiSite** allows you to share information with a geographically distributed team.

Rational ClearCase LT. Provides software configuration management and a built-in process to track changes to all software project assets, including requirements, visual models, and code. Rational ClearCase LT supports *Unified Change Management*, Rational's best practices process for managing change and controlling workflow.

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

Rational TestManager. Helps you create real-world functional and multiuser tests to determine the performance and reliability of e-business, multitier, and database applications. TestManager tracks how many tests have been planned, scripted, and run; which requirements have been covered; and the number of tests that have passed and failed. TestManager allows your team to obtain an objective assessment of project status and create and customize reports to communicate these findings to project stakeholders.

Simplify the Solution for Your Team's Development Needs

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

- 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.
- One location for online Help, Rational Suite documentation, and Rational Technical Support.

Rational Suite: Summary

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

		The Project Leader	The Analyst	The Developer			The Tester	All Roles
		Team Unifying Platform	AnalystStudio	DevelopmentStudio		DevelopmentStudio RealTime Edition	TestStudio	Enterprise
	Rational Tool	**Windows**	**Windows**	**Windows**	**UNIX**	**Windows**	**Windows**	**Windows**
Team Unifying Platform	Rational Unified Process	•	•	•	•	•	•	•
	Rational RequisitePro	•	•	•	• W	•	•	•
	Rational ClearQuest	•	•	•	• W	•	•	•
	Rational SoDA for Word	•	•	•	• F	•	•	•
	Rational ClearCase LT	•	•	•		•	•	•
	Rational TestManager	•	•	•		•	•	•
	Rational Rose		• M	• E	• E	• RT		• E
	Rational Robot						•	•
	Rational TestFactory						•	•
	Rational PureCoverage			•	•	•	•	•
	Rational Purify			•	•	•	•	•
	Rational Quantify			•	•	•	•	•
	Rational Process Workbench							•

M = Data Modeler Edition
E = Enterprise Edition
RT = RealTime Edition
F = Rational SoDA for FrameMaker
W = Accessed through the tool's Web interface

NOTE: For information on **Rational Suite ContentStudio**, see http://www.rational.com/products/cstudio.

Incorporate Best Practices into Your Team's Process

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

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

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

The Rational Unified Process at a Glance

The overview diagram of the Rational Unified Process in Figure 2 presents a rich visual representation of a full development lifecycle for a software project.

Figure 2 The Rational Unified Process Overview

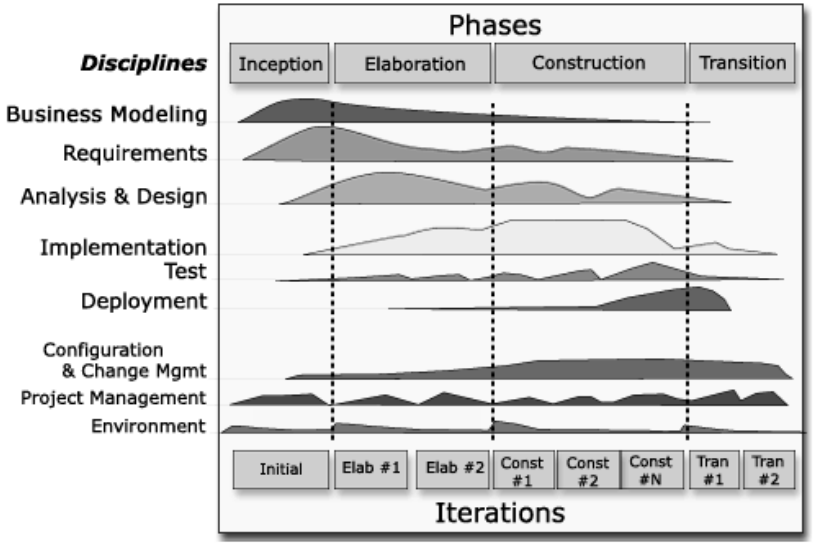


Figure 2 represents empirical data collected by Rational Software, showing that software development is best organized into *phases*, each of which is performed in a series of *iterations*. Throughout each phase, project team members from each of the software development disciplines (analysts, developers,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

Each iteration focuses on one of these development phases:

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

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

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

Adopting the Rational Unified Process

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

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

The Rational Unified Process and Rational Suite

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

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

What's Next?

The remainder of this book:

- Shows how analysts, developers, testers, and project leaders benefit by using Rational Suite.
- Shows how Rational Suite tools fit into an effective iterative process by focusing on requirements management, analysis and design, implementation, testing, configuration and change management, project management.
- Directs you to more information about Rational'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 application or disappointed in your product?

Understand the Problem Before You Invest

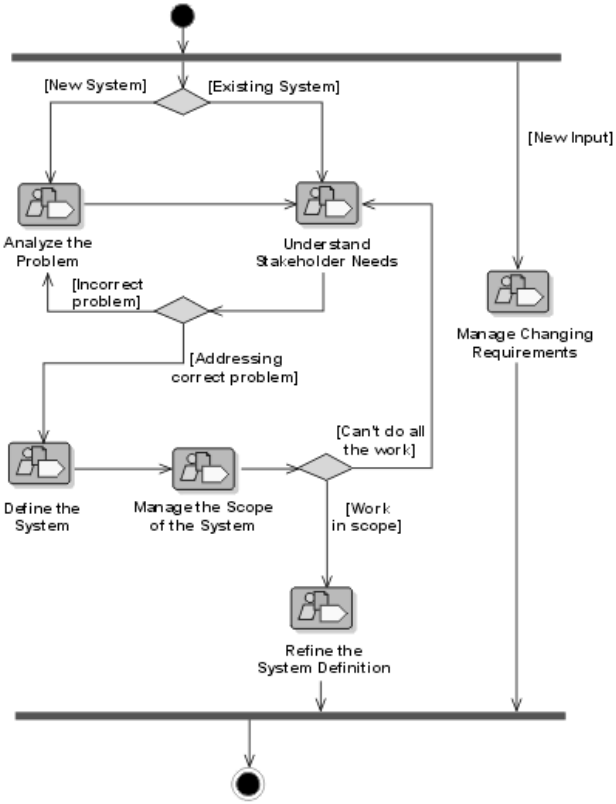
As an analyst, you actively elicit and gather information from *stakeholders* to help determine *what* the problem is. Your first job is to identify the problem your organization will address. You represent the stakeholders by capturing, organizing, and managing their expectations and requests for the system.

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

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

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

Figure 3 Requirements Workflow



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

Figure 3 shows a typical Requirements workflow. Each workflow detail represents a key skill that needs to be applied to perform effective requirements management.

Iterative Development and the Challenge of Managing Requirements

During development, requirements change and evolve. For example:

- Your competitors release new or updated versions of their products. To stay competitive, you add requirements for new features to your project.
- The customer was unclear or undecided about some requirements at first, so you might add, remove, or clarify requirements later on.
- Technology advances after you start product development. You need to determine if you can incorporate 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 or postpone the requirement.

Capture and Manage Requirements with Rational Suite AnalystStudio

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

Defining Requirements for Your Team

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

- As an analyst, you use Microsoft Word to document project requirements. Throughout the software development lifecycle, you use requirements to communicate what to build and test. The requirements form the foundation of the system'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-based tool enables 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, visual models, and test assets, 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.

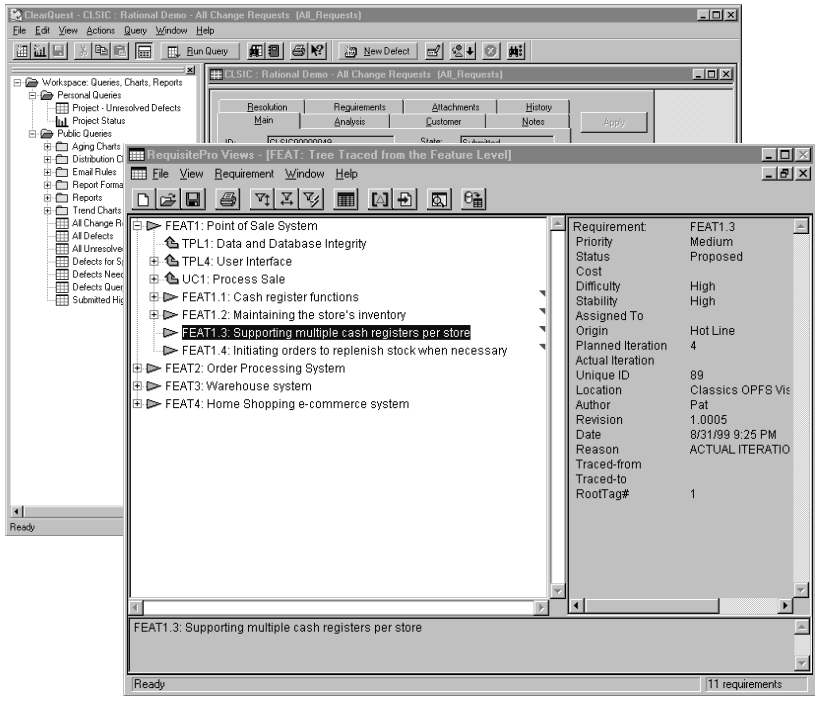
Rational ClearQuest encourages collaboration by tracking new feature, enhancement, or change requests from team members and other stakeholders. The **ClearQuest Web** interface ensures that UNIX users and other team members who do not have ClearQuest on their desktops can participate in this collaboration,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

With Rational ClearQuest, your team and other stakeholders can evaluate requests, determine their impact on the system, and when applicable, validate the changes. To establish how change requests fit into the structure of features and main requirements, you can link requests to an existing or new project requirement in Rational RequisitePro, as shown in Figure 4.

ClearQuest MultiSite allows geographically distributed teams to replicate a centralized database to each remote site and then synchronize the changes made at each site with changes made at other sites.

Figure 4 Using ClearQuest and RequisitePro to Manage Project Changes

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.

Measuring Progress and 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 plans and *test cases*, manual and automated tests, and test results are stored in a central location using TestManager. These artifacts are stored in a single database, using a single interface, and are 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 use case requirements in RequisitePro, you can use the Rational 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 correctly?

Can you maintain the integrity of the system's architecture?

Can you easily extend the system's design?

Can you reuse project components?

Does your team know what they are building?

How does your team communicate changes to each other?

Managing a Complex Process

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.

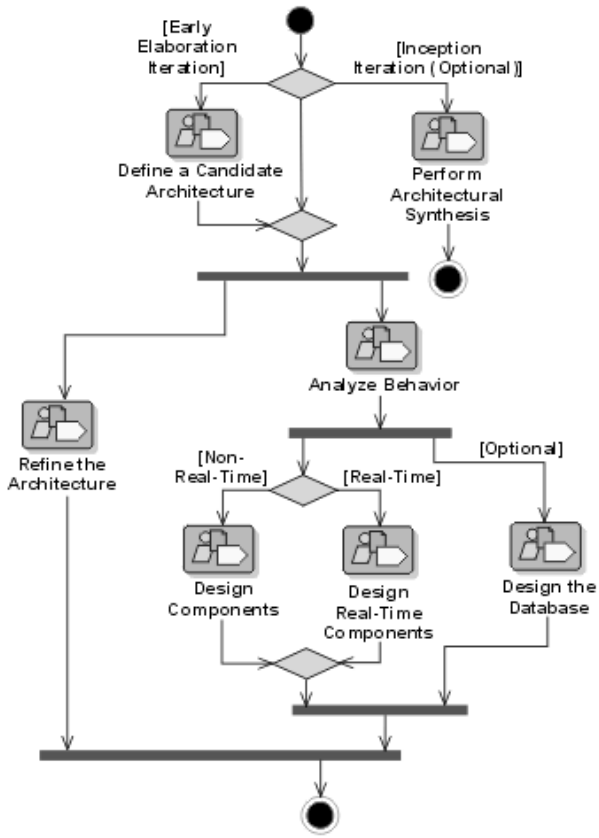
Analysis and Design: The Next Steps in Iterative Development

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 you work 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.

Figure 5 Analysis and Design Workflow



One purpose of analysis and design is to transform the requirements into a system design. As described in the Rational Unified Process, the workflow in Figure 5 shows how developers establish whether the system as envisioned is feasible, and assess potential technologies for the solution. Developers design an architecture for the system and continuously adapt the design to match the implementation environment and meet performance criteria.

Creating Resilient Component-Based Architectures with Rational Suite DevelopmentStudio

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

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

Communicating Visually with Models of Your System

Rational Rose (Enterprise Edition) 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 Studio, Visual C++, ANSI C++, Java, and IDEs including IBM Visual Age for Java, HP Workbench, and Sun Workshop.

Keeping Code and Models Consistent

After 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 theses 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 analysts, developers, testers, project leaders, and other stakeholders.

Validating Changes to the System

Rational ClearQuest tracks features, enhancements, or change requests from team members and other stakeholders in a form accessible to everyone. The **ClearQuest Web** interface ensures that UNIX users and other team members who do not have ClearQuest on their desktops can 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, 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, 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 Suite 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 QualityArchitect is a feature of **Rational Rose (Enterprise Edition)** that automates the mechanical aspects of test code creation by generating test code from visual models. This 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 run. This information allows you to determine if your tests have actually run the lines of code that were intended to be tested.

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

Tracking Test Results

As a developer, it is critical to communicate the status of projects and activities with your team members and other project stakeholders. 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.

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

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. **ClearQuest MultiSite** allows you to share information with a geographically distributed team.

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, after 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 Suite DevelopmentStudio for UNIX.

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

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

Building Complex, Real-Time Systems

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

Using Rose RealTime, you can:

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

Working the Way Real-Time Systems Operate

Rational Rose RealTime allows you to build real-time systems the way real-time systems operate. Rose RealTime:

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

Summary

- Rational Suite DevelopmentStudio (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 synchronized throughout the development lifecycle.
- ClearCase LT helps developers manage changes to project artifacts and makes those changes available to the project leader for integration in the next baseline.
- Rational TestManager helps you track how many tests have been planned, implemented, and run. It also helps you track which requirements or Rose model elements have been covered, and the number of tests that have passed and failed.

Deciding to Release: The Tester

4

Is your project behind schedule? How far?

Does your system scale to accommodate increasing load?

How many critical bugs were discovered after your last release?

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

Making the Crucial Decision to Release

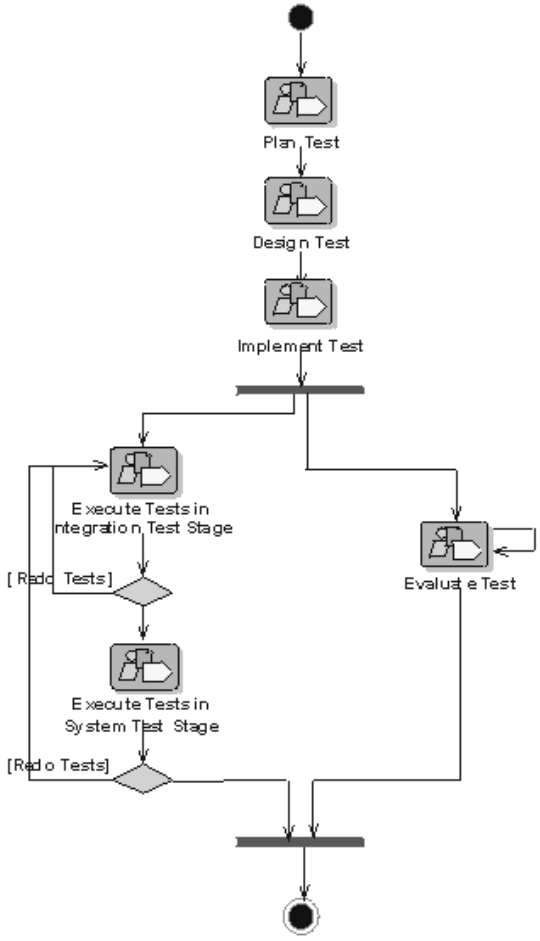
As a tester, you use all types of project artifacts to plan, design, and run tests. You ensure that software meets its requirements and is stable by testing the relationship between project assets. In each iteration, you communicate application defects and pinpoint performance problems so that your team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

Subsystem and System Tests: The Last Step in Iterative Development

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

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

Figure 6 Test Workflow



The workflow in Figure 6 shows a typical Test workflow as described in the Rational Unified Process. Each workflow detail represents the key skill needed to verify the proper integration of all components of the software, to verify that all requirements have been correctly implemented, and to ensure that defects are addressed before releasing the software.

Verifying Software Quality with Rational Suite TestStudio

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

Unifying the Team by Keeping Members Informed

Rational RequisitePro provides current, accurate information about requirements. RequisitePro indicates to you, a tester, that there are new or revised requirements that need to be tested. You use the requirements and related artifacts to create your test plan and track testing progress. 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 run tests. Requirements, visual models, and source code are some of the *test inputs* that you can use to create a test plan, so that all aspects of your system can be tested, including product features, system architecture, and code. With TestManager, test inputs are used to create specific *test cases*. A test case describes a testable and verifiable behavior in a system. It can also describe the extent to which you will test an area of the application.

For example, TestManager can verify whether you have tested all your requirements, and whether the test cases based on these requirements pass the tests. This relationship between project assets allows you to test for quality early in the development lifecycle, often beginning with product features, and continuing through implementation and release.

Does Your Application Meet Requirements?

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

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

Is Your Application Reliable?

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

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

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

Does Your Application Have Memory Leaks?

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

Is Your Application Fast Enough?

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

Does Your System Perform Under Production Load?

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

Have You Tested Enough?

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

Optimizing Defect Tracking

Rational ClearQuest tracks the defects that are found in your software project and provides a description, as well as other details, about the bugs. Rational testing tools are integrated with ClearQuest to simplify the process of entering defect information and to ensure that repaired defects are verified. **ClearQuest Web** is 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 that are available through the desktop client. Team members are able to access and create records, queries and reports on areas of interest specific through the Web module as they would through the client version.

Summary

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

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?

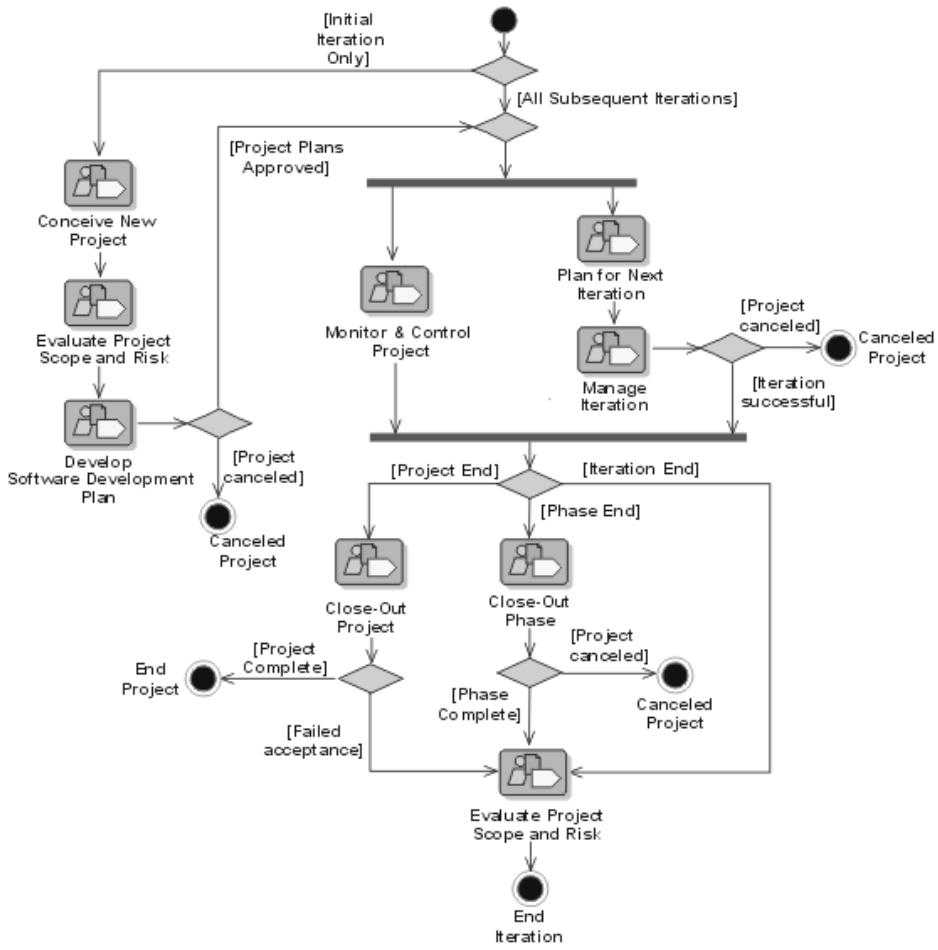
Is the project over budget?

How Well Can You Plan Ahead?

Software project management is an art of balancing competing objectives, managing risk, and overcoming constraints to successfully deliver a product which meets the needs of both customers and users.

As a project leader, you identify and manage project risks, monitor your team's progress, and plan each iteration. Your ongoing responsibilities are to assign and schedule work, and to monitor the progress of the project. Early in the development lifecycle, your team identifies, implements, and tests the most risky features and architectures. Monitoring project progress involves collecting and assessing the latest metrics or status reports from each team member. Throughout development, you analyze project data to determine how well the team is meeting its objectives. You also use this data to manage change and plan subsequent iterations.

Figure 7 Project Management Workflow



In general, the purpose of project management is to balance competing objectives, manage risk, and overcome constraints to successfully deliver a product which meets the needs of all stakeholders. The workflow in Figure 7 shows how project leaders plan an iterative project, both throughout the project and for a particular iteration, and monitor the progress of a project, as described in the Rational Unified Process.

Rational Suite: The Complete Solution for Iterative Development

Each edition of Rational Suite is optimized to support a major team role: analyst, developer, tester, and project leader. Each Suite edition also includes software that unifies teams with proven processes for managing change, ensuring quality, and improving communication from requirements to release. This tool set is called the Team Unifying Platform, and focuses on all product domains: requirements, modeling, testing, and change management. All Suite editions include the Rational Team Unifying Platform (Rational Unified Process, RequisitePro, ClearQuest, ClearCase LT, SoDA, and TestManager) that help team members obtain the information they need in order 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.

Implementing Best Practices

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 software development best practices:

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

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

Developing Complex Products Using Unified Change Management

Unified Change Management (UCM) is 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.

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.
- 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.
- 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 project report, you start with a Microsoft Word template, either one provided with SoDA or one you create yourself. This report provides the information you need to plan the next iteration.

You can use the project report to:

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

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

- Rational Suite 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.
- 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.
- Unified Software Project Management focuses on compiling information to assess the project’s status.
- SoDA allows you to extract and compile critical information so that you can articulate the project’s status to stakeholders.

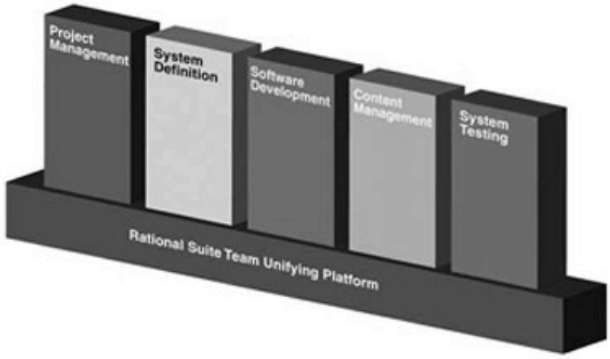
Next Steps

6

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, developer, tester, and project leader (see Figure 8).
- Rational Suite gives you the support to communicate and track changes in a repeatable and predictable manner so you can control changes to your project.
- By managing change and implementing the most risky project features in early iterations, your team can correct serious project flaws with less difficulty and at less expense.
- Within the framework of iterative development, Rational Suite promotes the most effective software engineering practices to help you solve the right problem and define the right solution.

Figure 8 Rational Products Support Every Function on Your Team



Adopting Practices and Tools

Adopting All of Rational Suite

Your team may decide to use the entire Rational tool set in your next project. The Rational Unified Process 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 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. After you've identified the next Rational tools to use, Tool Mentors and Extended Help can guide you in using them.

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

Exploring 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 *Rational Suite Tutorial* (Windows only). *Rational Suite Tutorial* 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 tools, so that you can learn more on your own.

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 onsite 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/support/contact.

Online Technical Resources

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

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

Additional Resources

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

Glossary

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

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 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 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.

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

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

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

Rational Purify. Automatically pinpoints hard-to-find run-time 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, multiuser database.

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

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

Rational 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 ContentStudio. Edition of Rational Suite optimized for companies managing content and code in complex e-business application. Contains the Team Unifying Platform –Rational Unified

Process, RequisitePro, ClearCase LT, ClearQuest, SoDA, and TestManager – as well as Rational SiteLoad and Vignette Content Management Server.

Rational Suite DevelopmentStudio. Edition of Rational Suite optimized for software development, available for Windows or UNIX. Contains tools from 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 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 testers. 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 using 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 creating or updating a Rose visual model from existing code, so that the visual model and code are synchronized. See *visual model*.

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

risk management. Consciously identifying, anticipating, and addressing project risks and devising plans for risk mitigation, as a way of ensuring the project'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 case. A set of test inputs that describe a testable and verifiable behavior in a system, the extent to which you will test an area of an application, and the results of each test.

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

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

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

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

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

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

L

M

P

Q

R

S

T

U

V

W