Introducing Rational Suite

Product Version	Rational Suite 2000.02.10
Release Date	April 2000
Part Number	800-023314-000

Rational®
the e-development company™

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

IMPORTANT NOTICE

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

Audience

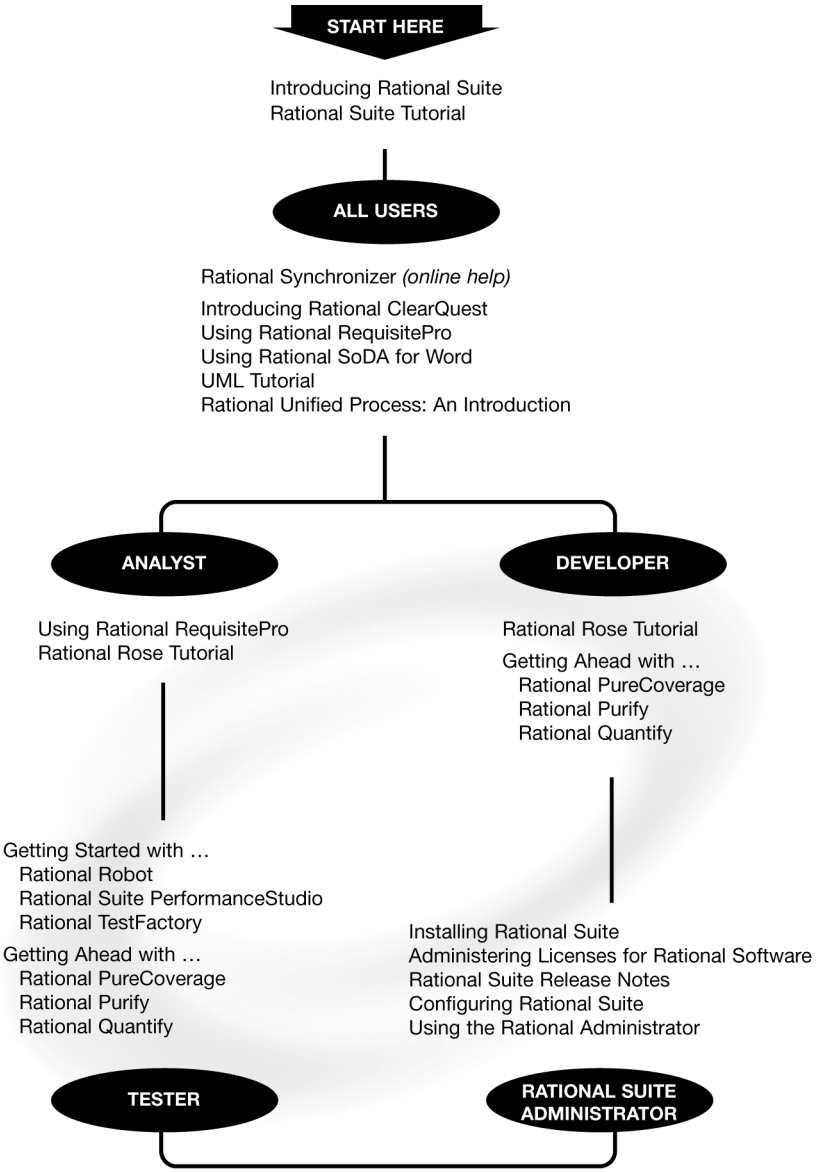
This guide is intended for all readers, including managers, project leaders, analysts, developers, and testers.

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.
- If you install Rational Suite DevelopmentStudio – RealTime Edition, PDF versions of the manuals for Rose RealTime are installed in `%ROSERT_HOME%\help`. All other manuals are available on the Rational Solutions for Windows Online Documentation CD.
- For more information on training opportunities, see the Rational University Web site: `http://www.rational.com/university`.

Rational Suite Documentation Roadmap



Contacting Rational Technical Publications

To send feedback about documentation for Rational products, please send e-mail to our technical publications department at techpubs@rational.com.

Contacting Rational Technical Support

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

Rational Technical Support Information

Location	Contact Information	Notes
U.S. and Canada	800-433-5444 781-676-2450 support@rational.com	When sending e-mail: – Specify the product name in the subject line, for example, “Rational Suite”. – For existing issues, include your case ID in the subject line.
Europe	+31 (0) 20 4546 200 support@europe.rational.com	
Asia Pacific	+61-2-9419-0111 support@apac.rational.com	

1 Welcome to Rational Suite

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:

- 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.

- If team members effectively track changing requirements and enhancements, they can easily reconstruct who changed what, when, where, and why. Effectively managing change helps the team deliver a project on time and within budget.
- Managing requirements more accurately and efficiently helps your project team deliver a product that meets customer expectations. The team builds a product that meets current reality, not reality two years ago.
- 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.
- 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 an integrated set of software development tools that supports your team throughout the development process: requirements management, design, testing, and change management. The Suite provides tools that address team communication and team productivity. Rational Suite helps you meet the challenges your team faces in designing, developing, and testing e-commerce applications, internet infrastructure software, e-devices, and other software systems and applications.

Rational Suite works with popular Integrated Development Environments (IDEs) such as Microsoft Visual Studio, 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?

Unifying the Team is Easy with Rational Suite

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. The Suite also provides project leaders with tools that help support their role in the project lifecycle.

Each edition of Rational Suite provides these core team-unifying tools that promote communication among team members.

- **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.
- **Rational ClearQuest**—Lets you track, manage, and report on product defects and change requests.
- **Rational SoDA (for Microsoft Word or Adobe FrameMaker)**—Extracts information directly from application databases and files, helping to keep project documentation current.

Optimize Your Team's Productivity

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.

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.

Rational Suite PerformanceStudio is designed for team members who verify system performance under a simulated user load and team members who want to test system architecture early in the development process.

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.

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 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 set of tools that are tested and updated together in each Suite release.

Incorporate Best Practices into Your Team's Process

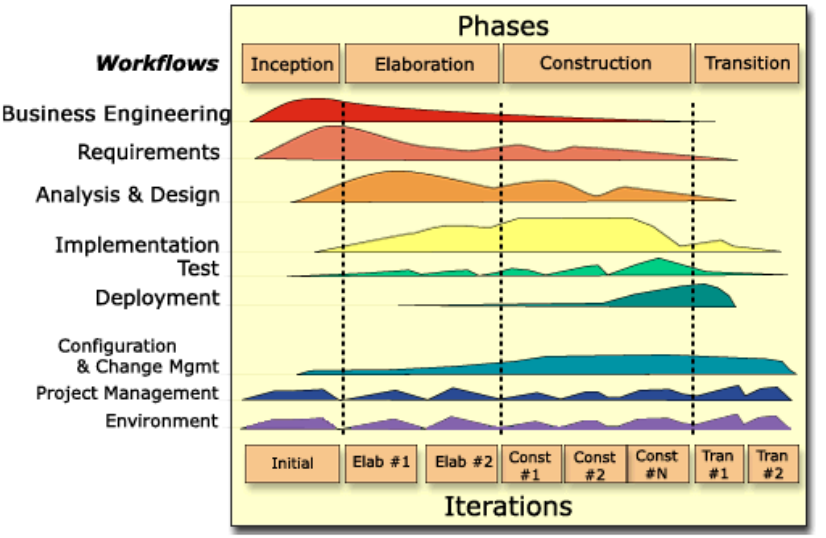
One of the first steps when solving common development problems is to evaluate your team's software development process. Rational advocates following a set of software-engineering practices based on its own experience as well as the experience of its many customers. All Rational Suite editions are designed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

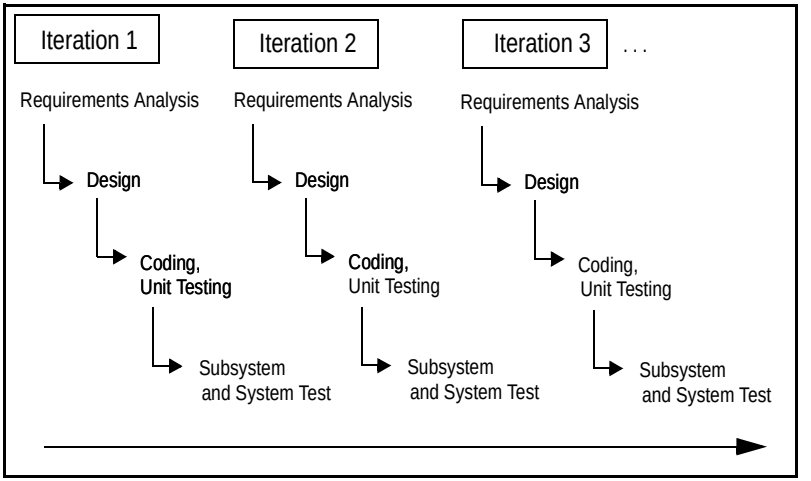
In the project lifecycle, there are multiple iterations. Each iteration focuses on one of these development phases:

- **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.



These steps 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.

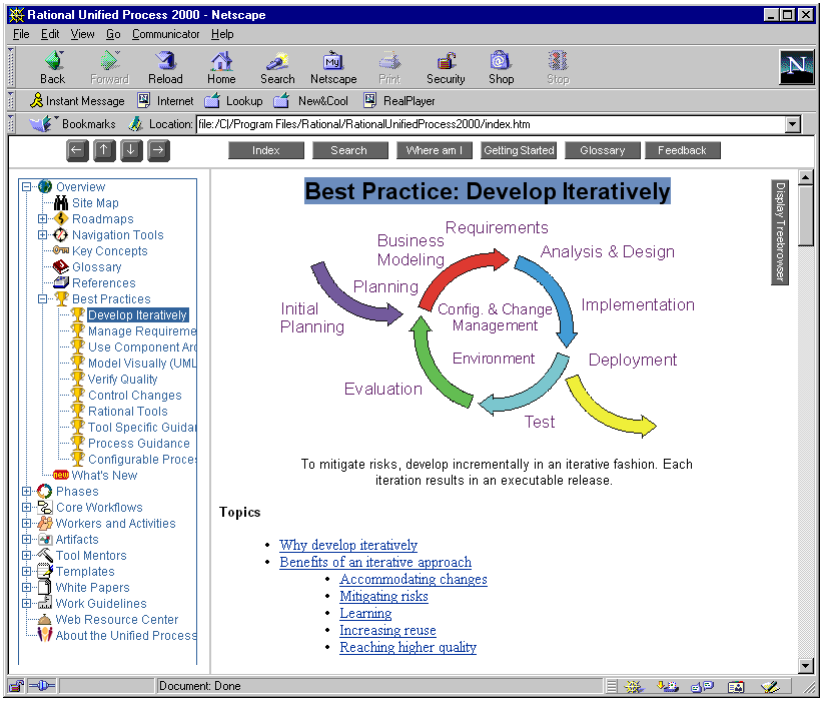
Visually model software. Visual models improve your team's ability to manage software complexity. Creating a graphical blueprint of the system's architecture conveys complex information in a common language that all members can understand. A visual model helps team members detect inconsistencies and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

Verify software 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.

Control changes to software. It is important to manage changes in a trackable, repeatable, predictable manner. Enhancement requests and defect reports help facilitate clear communication among team members. Your team should also 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 Rational tools and provides pathways to parts of 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, and subsystem and system testing.
- Directs you to more information about Rational's products, services, and the Rational Unified Process.

2 Building a Bridge: The Analyst

Are you meeting your users' changing needs?

Did feature creep delay your last release?

Is your project over budget or behind schedule?

Are your users dissatisfied with your product?

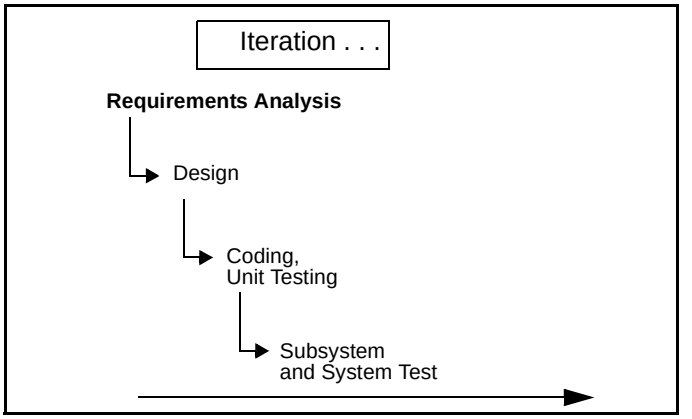
Bridging the Gap between the Customer and Your Team

As the analyst, you interpret requests from stakeholders and determine *what* the system will do. You represent the customer by collecting, managing, and communicating requirements to the rest of the development team. It is essential that all stakeholders clearly understand the system's objectives and requirements even as these objectives and requirements change during the project.

Requirements Analysis: The First Step in Iterative Development

In iterative development, your project team plans, develops, and tests one subset of system functionality. Each iteration results in either an internal or external release. In the next iteration, 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 can change and evolve. For example:

- Your competitors have released 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.

Managing 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 articulate and manage requirements in a form accessible to all stakeholders. It combines the familiar and flexible environment of Microsoft Word with the power of a relational database.

You, as the analyst, can use Microsoft Word to document project requirements. You then store and track this information in the RequisitePro database. The database allows you to specify requirement attributes, and to sort, filter, and relate requirements to other requirements. RequisitePro's comprehensive features enable you to quickly assess the impact of any changes on requirements.

Because every team member needs the same understanding of project goals and objectives, you have to keep your team up to date on requirements activities. RequisitePro makes this easy by providing **RequisiteWeb**. This Web capability enables UNIX users and other team members who do not have RequisitePro on their desktops to review and update requirements.

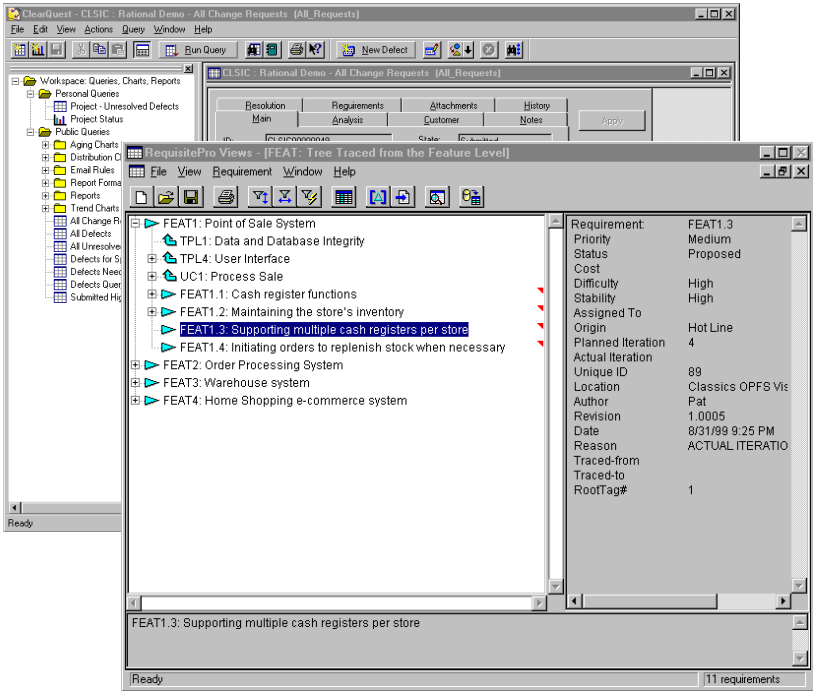
Tracking Requests for Your Project

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.

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.

Chapters 3 and 4 discuss ClearQuest's defect tracking features.

ClearQuest Change Request



Requisite Pro Feature Tree

Helping Your Team Communicate Visually

Rational Rose (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.

Rose unifies the team by helping you create models that all team members can share, test, and revise. It ensures that all members have the same understanding of the project. Analysts use Rose to describe a system at a high level. Architects continue this work by using Rose to design the system in more detail.

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.

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 instance, in Rational Suite AnalystStudio, once you've captured feature and use case requirements in the RequisitePro database, you can use the Synchronizer to generate test requirements. You can also use the Synchronizer to generate a sequence diagram in Rose.

Summary

- Rational Suite AnalystStudio provides an integrated set of tools to help you effectively manage and communicate requirements to your project group.
- Rational RequisitePro, the primary tool in AnalystStudio, helps you interpret requests from stakeholders and determine *what* the system will do. The entire team uses these requirements as a foundation for their work.
- Effectively managing requirements helps your team avoid common development pitfalls and deliver products on time and within budget.

3 Managing Complexity: The Developer

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?

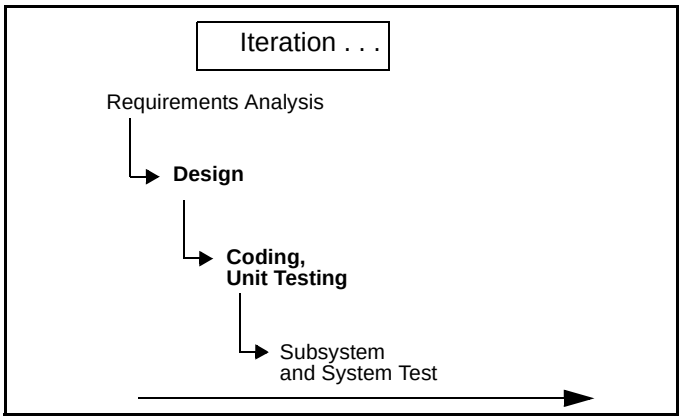
Managing a Complex Process

Architects and developers define *how* the system works. In the architect role, you design the system architecture. You must design a flexible system that can be modified as customer needs change. As the developer, you create, modify, and manage code for the system on the basis of 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 you work with component-based architectures, your project team can create new components or reuse, and even customize, existing components from previous projects or commercially available sources.



As the developer, you produce executable code that can be evaluated against the project requirements 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 Component-Based Architectures with Rational Suite DevelopmentStudio (Windows and UNIX)

Rational Suite DevelopmentStudio is the complete solution for software architects, designers, and developers. This Suite helps your team design and build the project right the first time.

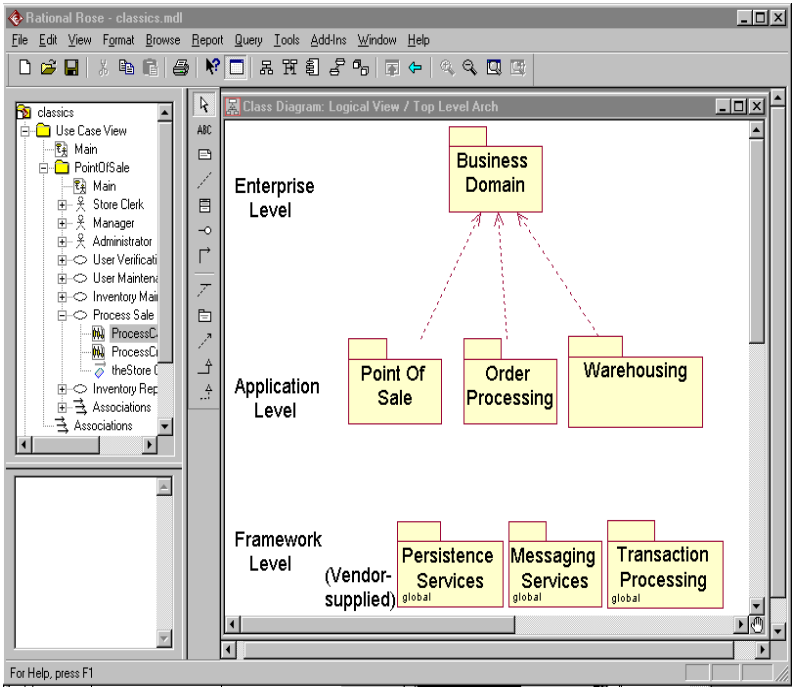
Rational Suite DevelopmentStudio – RealTime Edition is further tailored for practitioners who focus on real-time and embedded development. For more information, please see *Creating Component-Based Executable Architectures with Rational Suite DevelopmentStudio – RealTime Edition* on page 33.

Communicating Visually

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 models that all team members can share, test, and revise. It ensures that all members have the same understanding of the project. During design and code reviews, team members use project models to see the ramifications of any changes they want to make to the code.

Logical View of Architecture in Rose



Jump-Starting Code Implementation

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

Note: The Rational Rose Modeler Edition in AnalystStudio does not generate code, update code, or update models.

Keeping Code and Models Consistent

Once you change code, you can direct Rose to update the model, ensuring that your model and code remain consistent throughout the project. This process is called *reverse engineering*.

Revising your visual models helps you see the impact of changes on the architecture. It is important to assess these changes because they may violate project standards or architectural decisions. If they do, they need to be reworked. If they do not, they must be reflected in the software architecture and communicated to the entire project team.

Evaluating Requirements Visually

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 stakeholders 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.

Validating Changes to the System

Rational ClearQuest tracks new feature, enhancement, 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 feedback.

Using Rose models, you can determine the impact of these requests on the system, and then if applicable, validate the changes.

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 perform code testing while you implement code. Rational Suite provides testing tools to use as soon as you have a working program.

Rational Purify checks every active C++ 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.

Rational PureCoverage checks every component in your program for lines of code that the program fails to exercise.

Rational Quantify detects performance bottlenecks, which are 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.

Tracking Test Results

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.

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 test requirements in Rational RequisitePro that correspond to those diagrams.

The UNIX version of Rational Suite DevelopmentStudio does not include the Synchronizer.

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

Rational Suite DevelopmentStudio – RealTime Edition is tailored for practitioners who focus on real-time and embedded development. This Suite edition contains all the tools in Rational Suite DevelopmentStudio but replaces 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. Generating these applications 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 offers an integrated tool set to optimize the definition, design, application generation, and testing of your real-time, embedded development project.
- Rational Rose, the primary tool in DevelopmentStudio, helps you design your system's architecture. By visually modeling software architecture, you can discover and correct design flaws early in the development cycle.
- You change the model to respond to changing requirements and new problems. Rose can update the model once you change code, ensuring that your model and code remain consistent throughout the project.
- By using Rose models, team members can see the ramifications of any changes they want to make to the code. It is important to assess these changes because they may violate project standards or architectural decisions.
- Your team's ability to manage software complexity improves if your team models architectures for your software. Creating models promotes team communication because it provides a common (visual) language to describe the system.

4 Deciding to Release: The Tester

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?

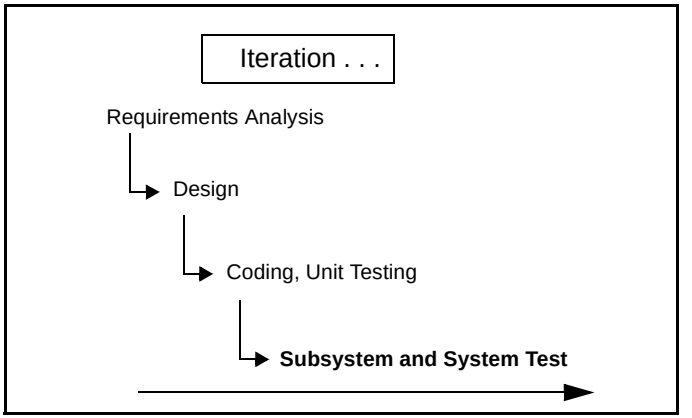
Do your users find serious bugs after you release your product?

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.



In the course of the project, you manage and track which scenarios, requirements, and code 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. Both Rational Suite TestStudio and Rational Suite PerformanceStudio (see page 39) help 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 can then create test requirements in RequisitePro to help track testing progress. Its 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.

Providing an Objective Status Report

Rational TestManager provides access to requirements and test information so your team can obtain an objective assessment of project status. TestManager helps you track how many tests have been planned, how many tests have been scripted, what tests have been run, what and how many requirements 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.

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 in Rational **LogViewer**. LogViewer reports the complete details of any failures: what test was running, what type of failure occurred, where it occurred and which verification point failed. LogViewer can also identify any requirement associated with a script.

Is Your Application Reliable?

Automated Reliability Testing—Rational TestFactory automatically generates tests that pinpoint severe defects: where the application crashes, hangs, or behaves erratically. 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 central repository that it shares with Robot and other Rational testing tools. Your team can generate coverage and progress reports from test results in this repository.

Rational Robot can 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++ 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 in Rational Purify, PureCoverage, and Quantify to ensure all reliability and application performance tests are repeatable.

Optimizing Defect Tracking

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.

Optimizing Your Work with the Rational Synchronizer

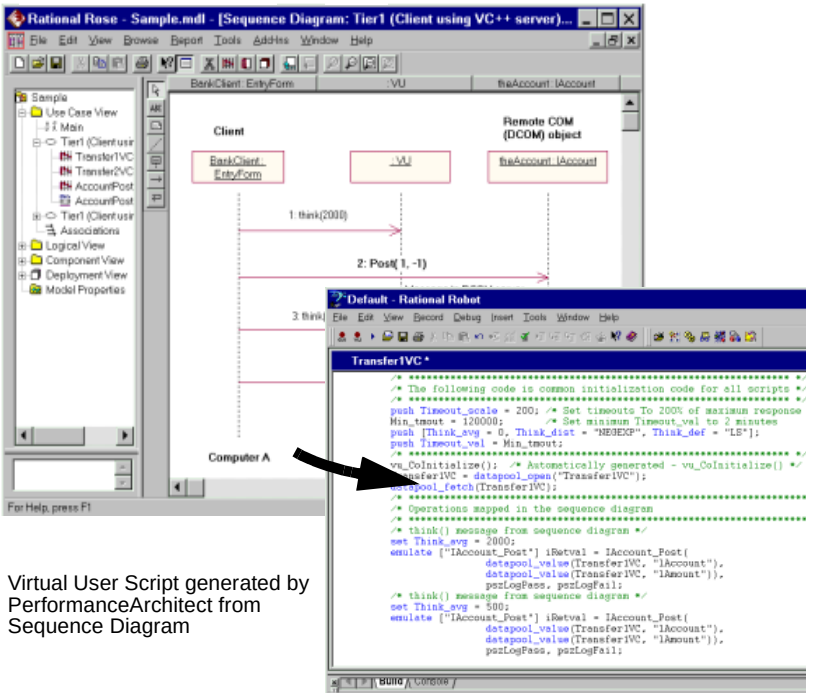
The **Rational Synchronizer** automatically creates items in your project based on the existence and status of related items. This tool helps your team manage project elements by ensuring that all related items exist at the right time and that none of them are lost.

For example, to generate scripts for test requirements that require automatic testing, use the Synchronizer to discover which test requirements do not have corresponding scripts. Then instruct the Synchronizer to create script shells for some or all of those requirements.

Verifying System Performance with PerformanceStudio

PerformanceStudio includes the tools in TestStudio, plus tools that create real-world, multi-user tests of system performance. You can use these tools to test performance as soon as your system architecture is designed.

Rose Sequence Diagram



Virtual User Script generated by PerformanceArchitect from Sequence Diagram

Does Your System Perform Under Production Load?

In addition to all the tools contained in TestStudio, PerformanceStudio includes:

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 thousands of users. As LoadTest runs these tests, it collects data that helps your team measure and predict your project's system performance.

Rational Rose (Enterprise Edition) allows your team to model your project's architecture and components using the industry-standard Unified Modeling Language. Like a blueprint, a model provides an overview of the system, which team members can share, test, and revise.

Rational PerformanceArchitect allows your team to test the system performance of different COM, DCOM, and Enterprise JavaBeans models created in Rational Rose. Using PerformanceArchitect, you convert a Rose model, which diagrams a sequence of actions, to a virtual user script. With this script, you can run system performance tests using LoadTest. Testing early helps you reduce project risk because your team can determine how a potential system architecture meets performance requirements before developing the design further.

Summary

- Rational Suite TestStudio is the complete solution for team members who verify the reliability, functionality, and application performance of software.
- Rational Suite PerformanceStudio provides the features of TestStudio, plus tools that verify system performance under production load.
- 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.

5 Managing Risk: The Project Leader

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. In the beginning of the project 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 plan subsequent iterations.

Rational Suite: The Complete Solution for the Iterative Process

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 more 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 these 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.
- **Control changes to software.** 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.

Control Changes to Software with Rational Suite

With the exception of Rational Suite Enterprise edition, each Rational Suite is customized to support a major team role: analyst, architect and developer, and tester. The Enterprise edition provides a comprehensive tool set that supports the entire team. Each edition also focuses on one or more product domains: requirements, modeling, testing, and change management. They all include team-unifying tools (Rational Unified Process, Rational RequisitePro, Rational ClearQuest, and Rational SoDA) that help project leaders obtain the information that they need to manage risk and control changes to software.

Planning Ahead

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.

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.

Meeting the Challenge of Developing Complex Products with ClearCase

To help your team and organization meet the challenges of configuration management, Rational Software offers Rational ClearCase. Although ClearCase is not included in Rational Suite, it is integrated with most Rational Suite tools and with Microsoft VisualStudio. This tool enables your organization to organize and manage versions, releases, and parallel development of multiple products.

Your project team can use ClearCase to control changes to source code and other project configuration items, such as Rational RequisitePro documents and Rational Rose model files. ClearCase tracks changes to every file and directory, maintaining histories of source code, binaries, executables, documentation, test suites, libraries, and user-defined objects.

Summary

- The Rational Suite family of products supports effective software engineering practices. As the leader of a project team, you oversee and participate in these best practices.
- Iterative development allows your team to manage risk and change early in the development process.
- In the beginning of the project lifecycle your team identifies, implements, and tests the most risky features and architectures.
- As your team implements more enhancements and features into the product, you can collect more feedback about the system. This early feedback, which includes defect tracking with ClearQuest, enables the team to change or adjust system architecture when it is easier and less expensive to do so.
- If your project team manages risk and change, you, as the project leader, can effectively plan and manage each iteration, and therefore the entire project.

6 Next Steps

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 and 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.

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

- Extended Help links the tools to 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 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 with the *Rational Suite Tutorial,* included in your documentation set. The tutorial provides a step-by-step approach that follows the Rational Unified Process through an iteration of a software development project.

Note: The UNIX version of Rational Suite DevelopmentStudio does not include a tutorial.

Using the Documentation Set

In addition to the *Rational Suite Tutorial*, each Rational Suite edition includes the following manuals:

- *Installing Rational Suite*—A guide to installing Rational Suite software.
- *Administering Licenses for Rational Suite*—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 Rational Administrator*—A guide and reference to the primary administration program, Rational Administrator (Windows only).
- 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.
- Tool 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 `%ROSERT_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 course, 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.jtmpl.

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

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

Rational Tool	Analyst Studio	Development Studio (Windows/Unix)	Development Studio - RealTime Edition	Test Studio	Performance Studio	Enterprise
Rational Unified Process	X	X	X	X	X	X
RequisitePro	X	X	X	X	X	X
ClearQuest	X	X	X	X	X	X
SoDA	X	X	X	X	X	X
Rose	Modeler Edition	Enterprise Edition	RealTime Edition		Enterprise Edition	Enterprise Edition
Robot				X	X	X
TestFactory				X	X	X
PureCoverage		X	X	X	X	X
Purify		X	X	X	X	X
Quantify		X	X	X	X	X
LoadTest					X	
Performance Architect					X	

Glossary

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.

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.

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.

Rational ClearCase

Provides comprehensive configuration management, including version control, workspace management, build 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 – bug 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

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 optimizes the productivity of analysts, developers, and testers – and unifies them, creating highly effective software development teams.

Rational Suite AnalystStudio

Edition of Rational Suite optimized for analysts. Contains the team unifying tools – Rational Unified Process, RequisitePro, ClearQuest, and SoDA – and Rational Rose (Modeler Edition).

Rational Suite DevelopmentStudio

Edition of Rational Suite optimized for system developers and designers. Contains the team-unifying tools – Rational Unified Process, RequisitePro, ClearQuest, and SoDA – 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 tools – Rational Unified Process, RequisitePro, ClearQuest, and SoDA – plus Rational Rose RealTime, Rational Purify, Rational Quantify, and Rational PureCoverage.

Rational Suite Enterprise

Edition of Rational Suite containing all Rational Suite tools except Rational LoadTest.

Rational Suite PerformanceStudio

Edition of Rational Suite optimized for test engineers who develop and run performance tests. Contains the team-unifying tools – Rational Unified Process, RequisitePro, ClearQuest, and SoDA – plus Rational Test tools, Rational Rose (Enterprise Edition), and Rational LoadTest.

Rational Suite TestStudio

Edition of Rational Suite optimized for test engineers. Contains the team unifying tools – Rational Unified Process, RequisitePro, ClearQuest, and SoDA – and Rational Test tools.

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 match. 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 do both forward and reverse engineering as often as needed.

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 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

Q

R

S

T

U

V

W