IBM Rational University

Rational. software

Essentials of IBM Rational RequisitePro v7.0

REQ370 / RR331 – October 2006
Student Manual
Part No. 800-027251-000

IBM Corporation
Rational University
REQ370 / RR331 Essentials of IBM Rational RequisitePro v7.0
Student Manual

October 2006

Printed in the United States of America.

This manual prepared by:
IBM Rational Software
18880 Homestead Road
Cupertino, CA 95014-0721
USA

Contents

Module 0: About This Course

Course objectives 0-2
Intended audience 0-3
Course agenda 0-4
Your team communication 0-8
Benefits of IBM Rational RequisitePro 0-11

Module 1: Product Introduction

Objectives: Product introduction 1-2
RequisitePro 1-3
Rational Unified Process (RUP) Overview 1-5
Project structure 1-7
Project templates apply predefined project structure 1-8
RequisitePro client interface organization 1-9
Packages 1-10
Views 1-11
RequisitePro documents 1-13
Getting started with Rational RequisitePro 1-15
Lab 1: Getting Started 1-16

Module 2: Plan Your Project

Objectives: Plan your project 2-2
RequisitePro requirement 2-3
Requirement attribute 2-4
First steps in defining your RM strategy 2-5
Requirements management plan 2-6
Create a new project 2-7
Requirement types 2-9
Document types 2-13
Additional project logistics 2-17
Lab 2: Customize Project Structure 2-20

Module 3: Gather, Organize, and Document Requirements

Objectives: Gather, organize, and document requirements 3-2
Requirement artifacts set 3-3
Requirements in RequisitePro 3-5
Requirements in a view 3-6
Edit a requirement 3-9
Import requirements from a CSV file 3-10
Delete a requirement in the Explorer 3-11
Lab 3: Gather and Enter Requirements 3-14
Requirements and associated documents 3-15
A requirements document in RequisitePro 3-17
Edit a requirement in a document 3-22
Import a document with the Import Wizard 3-23
Delete a requirement in a document 3-24
Offline authoring 3-26
Lab 4: RequisitePro Documents 3-29

Module 4: Manage Your Requirements

Objectives: Manage your requirements 4-2
Requirements traceability 4-3
Suspect links 4-10
Export a view to Word or CSV format 4-14
Requirement queries 4-15
Define filter and sort criteria for query 4-17
Lab 5: Traceability and Queries 4-18
Manage requirements using metric reports 4-19
Requirement metrics 4-20
Basic filters for query 4-23
Time-sensitive filters 4-25
Lab 6: Metrics 4-27
RequisitePro baseline 4-28
RequisitePro Baseline Manager 4-29
Comparing baselines 4-33
Generating a baseline comparison report 4-36
Lab 7: RequisitePro Baseline Manager 4-38

Module 5: Communicate Your Requirements

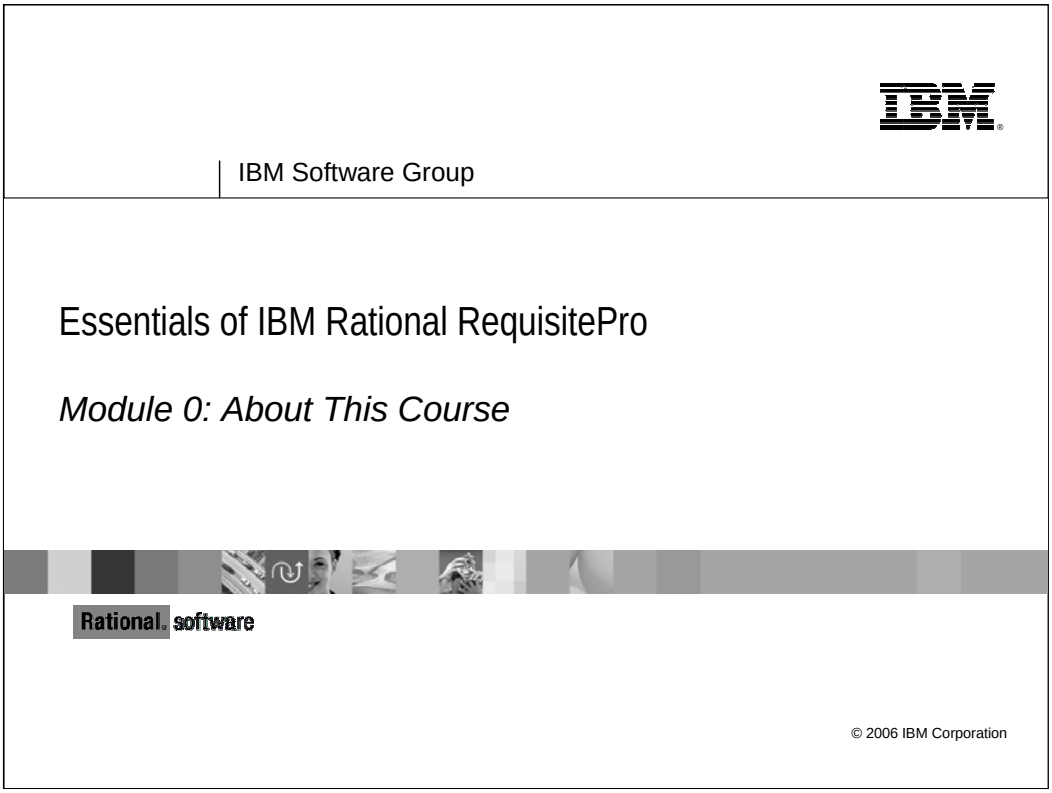
Objectives: Communicate your requirements 5-2
Requirement revision notification 5-4
Enabling revision notification 5-5
Subscribing to requirement revision notification 5-7
Team communication using discussion groups 5-9
Create discussions 5-11
Identifying discussions in Rational RequisitePro 5-12
Viewing discussions 5-13
Querying discussions 5-14
Lab 8: Group Discussions 5-17

Module 6: Summary and Tips

Summary 6-2
Qualities of software requirements sets 6-6
Tips for writing good requirements 6-8
RequisitePro tip: Renumber requirements option 6-12
RequisitePro tip: Capabilities for documents 6-13
Other sources of information 6-14

Glossary

RU e-st Artifacts



Contents

Course objectives	0-2
Intended audience	0-3
Course agenda	0-4
Your team communication	0-8
Benefits of IBM Rational RequisitePro	0-11

Course objectives

- Define the components and structure of an IBM® Rational® RequisitePro® project.
- Connect to and work in a RequisitePro project
  - Gather, organize, and document requirements
  - Import and create requirements
  - Create RequisitePro documents
- Manage requirements
  - Create packages
  - Define traceability between requirements
  - Open views, queries, and metrics
- Communicate and manage changing requirements
- Create and compare baselines using the RequisitePro Baseline Manager

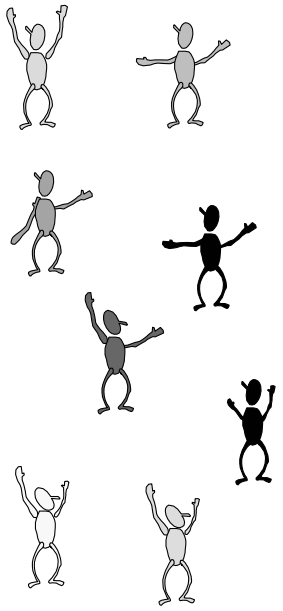
2

The student labs guide you through using Rational RequisitePro to manage a project according to the guidelines in the Requirements Management Plan. You gain an understanding of the development process using Rational RequisitePro as your requirements management tool.

- Gain skills and experience.
  - Transfer knowledge to your environment
- Apply techniques demonstrated in class
  - Use Rational RequisitePro features to complete your tasks and responsibilities
- Communicate project information to the whole team
- Manage changing requirements effectively

Intended audience

- Analysts
  - Business analysts
  - System analysts
- Engineers
  - Process engineers
  - Business engineers
- QA team
- Tech writers
- Architects



3

IBM

The primary audience for this course consists of business and requirements analysts. Additional audiences include any team member involved in creation of requirements, specification, use, and management, such as these roles:

- Technical writers
- QA
- Documentation managers
- Project leads
- Application experts
- Testers
- Designer and other software developers

Course agenda

- Morning
  - Module 0: About This Course
  - Module 1: Product Introduction
  - Module 2: Plan Your Project
  - Module 3: Gather, Organize, and Document Requirements
- Lunch break
- Afternoon
  - Module 4: Manage Your Requirements
  - Module 5: Communicate Your Requirements
  - Module 6: Summary and Tips

4

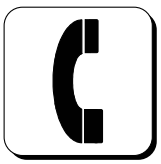
Course materials

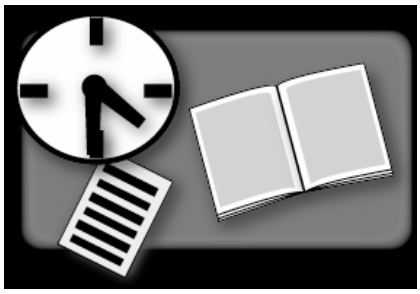
- *Essentials of IBM Rational RequisitePro* student manual
- *Essentials of IBM Rational RequisitePro* student workbook

5

Logistics







Morning
Two 15-minute breaks
Lunch
1 Hour
Afternoon
Two 15-minute breaks

6

IBM

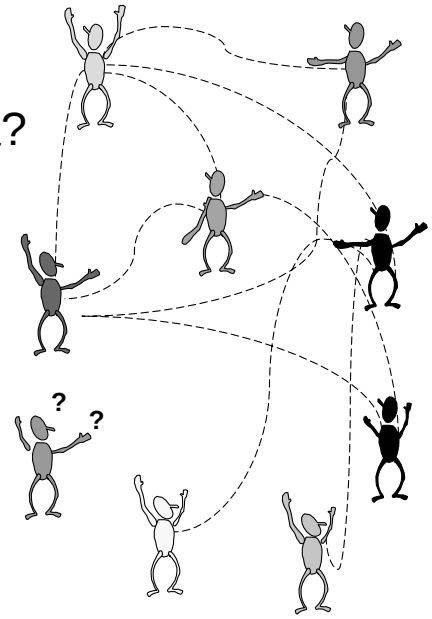
Introductions

- Your organization
- Your role
- Your background and experience
  - Software development experience
  - IBM Rational tools experience
- Course expectations

7

Your team communication

- Is there a centralized place for reviewing requirements and data?
- How are changes your communicated?
- How do you monitor project progress and status?
- How do you manage change?



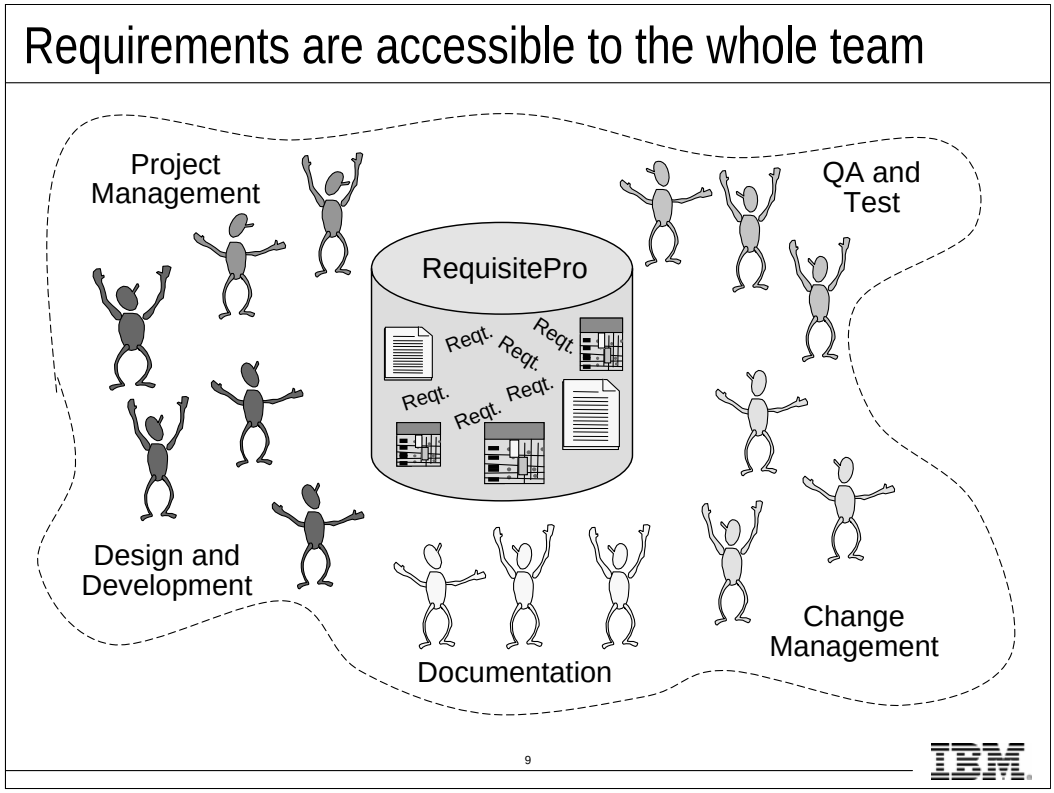
8

IBM

How does your team currently communicate project requirements? Here is a list of techniques that are commonly used across many software projects:

- E-mail messages
- Memos
- Meetings
- Spreadsheets
- Printed documents
- White boards

Depending upon the project size, any or all of these techniques may work. Most software projects today are nontrivial and require a coordinated effort to effectively manage their requirements.



Effective requirements management requires you to organize your requirements so that they are available to the whole team. It also requires you to control change and ensure that your project does not spiral out of control as changes occur. When a change occurs, it must be communicated effectively, and the impact of the change must be fully understood.

Key problems

- There is no consistent way to organize user needs.
- Developers and testers receive poorly defined requirements.
- Requirement documents are difficult to write, review, and update.
- Applications miss customer expectations.
- Feature creep causes schedule delays and cost overruns.
- There is no easy way to review feature priorities and status.
- Requirement changes cannot be traced quickly.
- Requirements are not communicated to the entire team.

10

Failing to manage requirements decreases the probability of meeting the project objectives.

Requirements management is the process of eliciting, organizing, and documenting requirements of the system. A requirements management process establishes and maintains agreement between the customer and team regarding changing requirements of the system.

Rational RequisitePro helps you manage project requirements.

Communication is a key factor to a successful project.

Benefits of IBM Rational RequisitePro

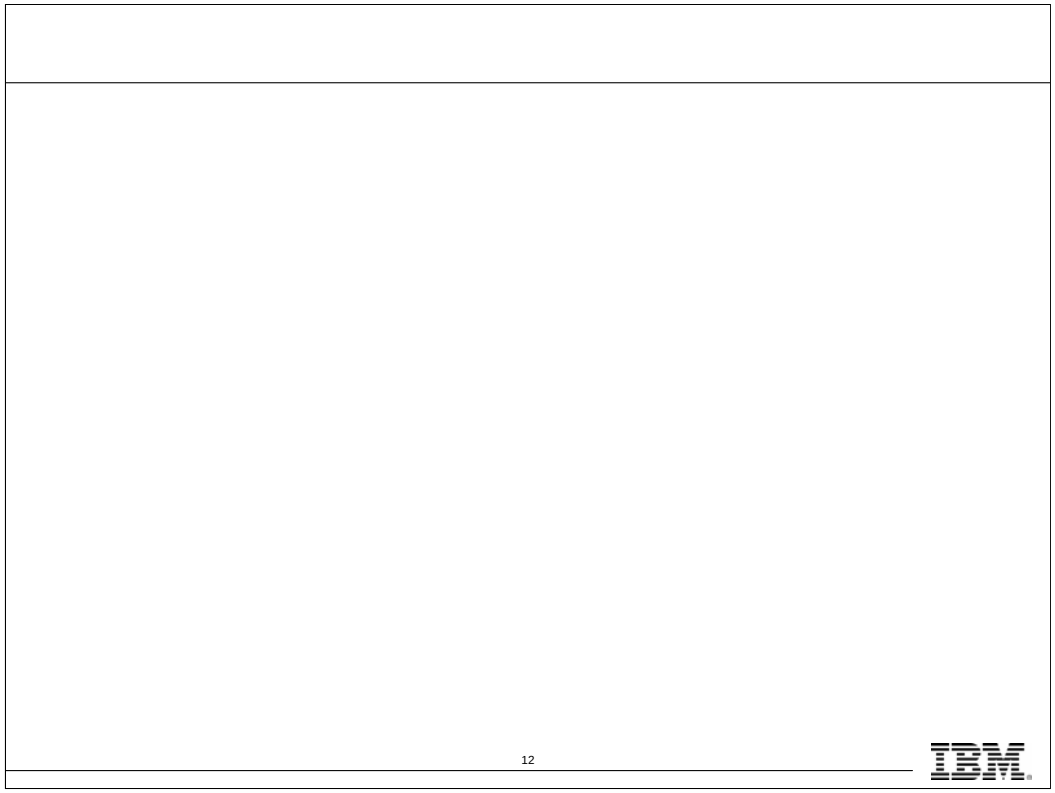
- Maintains documents with the requirements dynamically linked to a database
  - Sort and query capabilities
- Identifies the impact of change with traceability features and impact analysis queries
  - Scope management and resource allocation decisions
- Integrates requirements with other lifecycle artifacts and processes
  - Clear communication across tools and teams

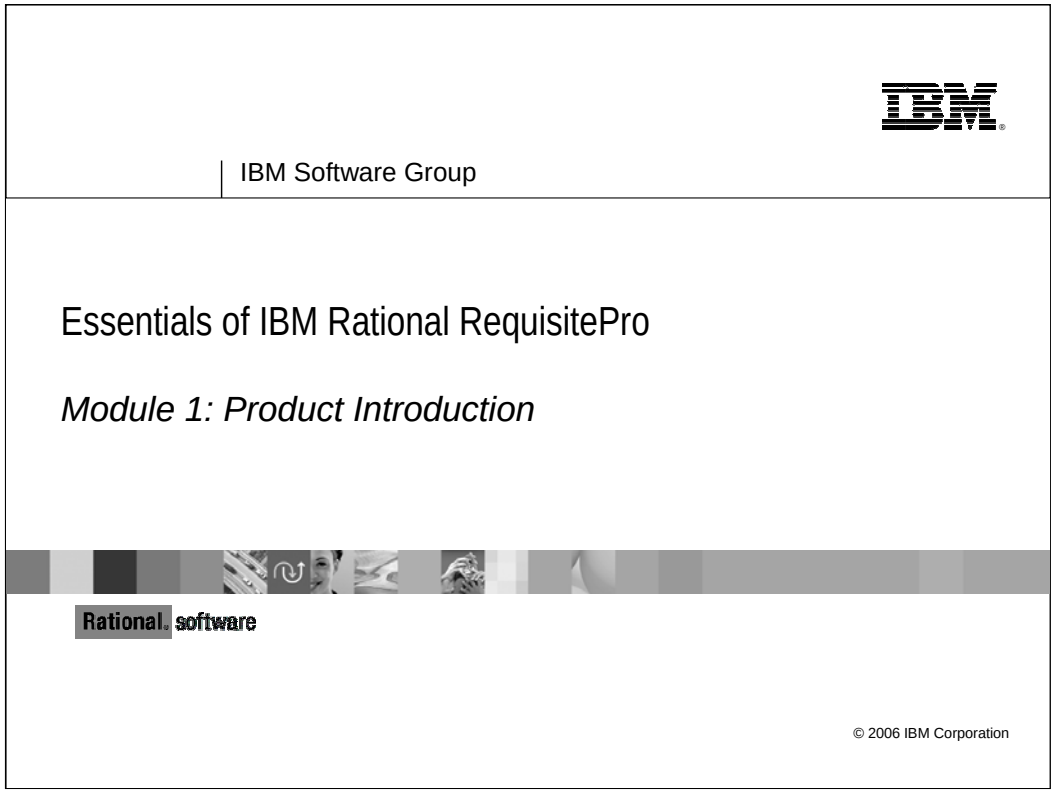
11

Rational RequisitePro gives you the power of a database and the flexibility of a word processor. You work with your requirements in a flexible environment – Microsoft Word,— yet you are able to manage your requirements more effectively because they are stored in a centralized repository. This enables you to comprehensively manage change, and it makes collaboration and communication easier.

Key benefits:

- **Ease of use:** Dynamic integration between Word and your requirements database.
- **Centralized storage:** Secure central requirements repository.
- **Change tracking:** Team is synchronized, and requirements have a version history that capture all changes to the requirements.
- **User security:** Permission privileges are defined, and you can enable LDAP (Lightweight Directory Access Protocol) for user authentication.
- **Use-defined project structure:** Requirement types, requirement attributes, and document types are easier to understand and organize.
- **Coverage analysis:** Set up and track relationships between requirements. Then you can query relationships about coverage analysis to ensure completeness.
- **Analysis of impact of change in requirements:** Trace effects between related or dependent requirements.





Contents

Objectives: product introduction	1-2
RequisitePro	1-3
Rational Unified Process (RUP) Overview	1-5
Project structure	1-7
Project templates apply predefined project structure	1-8
RequisitePro client interface organization	1-9
Packages	1-10
Views	1-11
RequisitePro documents	1-13
Getting started with Rational RequisitePro	1-15
Lab 1: Getting Started	1-16

Objectives: Product introduction

- Describe software development process.
  - How and where IBM® Rational® RequisitePro® fits in the process
- Define Rational RequisitePro interface
  - Project structure and organization
- Navigate Rational RequisitePro

2

In this module, you will gain a basic understanding of all the parts of Rational RequisitePro and how they relate to each other. Furthermore, you will briefly review the Requirements Discipline from the Rational Unified Process in which Rational RequisitePro will assist in the project development lifecycle.

RequisitePro

- Is a requirements management tool
- Enables you to track relationships between requirements
- Provides functionality to analyze the impact of changes to requirements
- Enables users to access RequisitePro project data using a browser with RequisiteWeb, which is a Web client of RequisitePro

3

IBM

RequisiteWeb uses a Web browser to provide platform-independent, thin-client access to RequisitePro project data. RequisiteWeb enables the following users to access RequisitePro requirement data:

- Distributed team members
- Teams that work in multiple-platform environments
- Linux browser users
- Reviewers

RequisiteWeb provides most of the capabilities of RequisitePro. It allows you to read, create, and modify RequisitePro project requirements and documents across a network and the Internet.

RequisiteWeb supports these browsers:

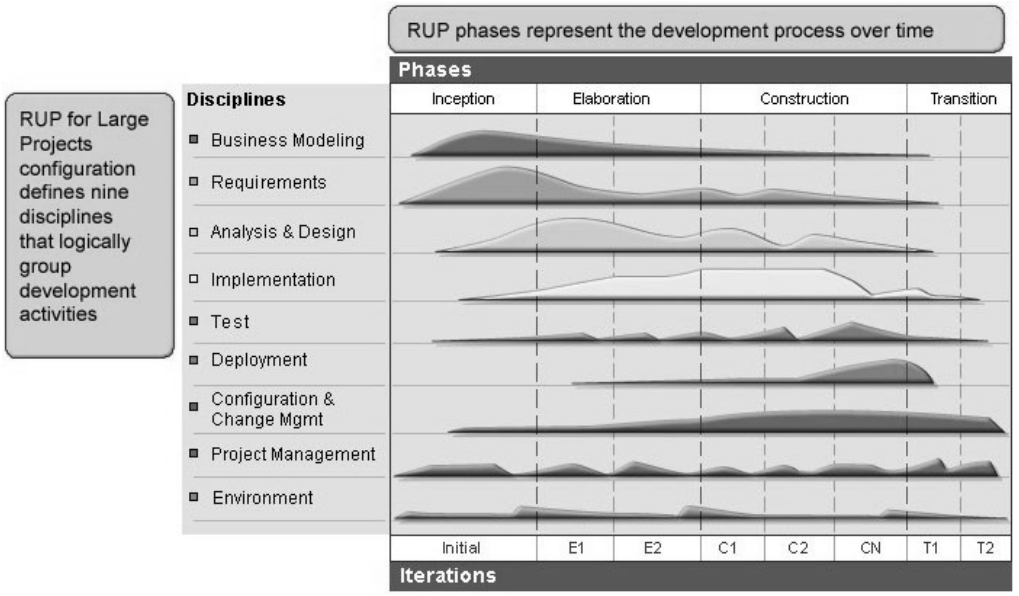
- Microsoft® Internet Explorer®
- Firefox
- Mozilla

RequisitePro (cont.)

- RequisitePro is integrated with Microsoft ® Word ® for creating document-based requirements
- Team members use RequisitePro to:
  - Plan projects by creating and editing requirements and requirements documents
  - Gather, organize, and document requirements
  - Manage requirements
  - Communicate with team members and stakeholders
  - Perform project administrative tasks

4

IBM® Rational Unified Process® (RUP®) overview

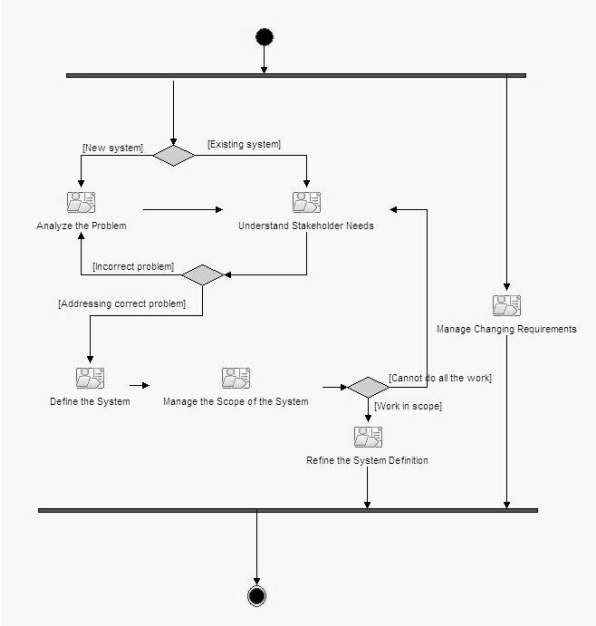


Rational RequisitePro is used throughout the development lifecycle. This *Essentials of Rational RequisitePro* course focuses on organizing a project, entering requirements, and managing those requirements.

With each discipline, the IBM Rational Unified Process (RUP) defines activities to help manage and control your development process.

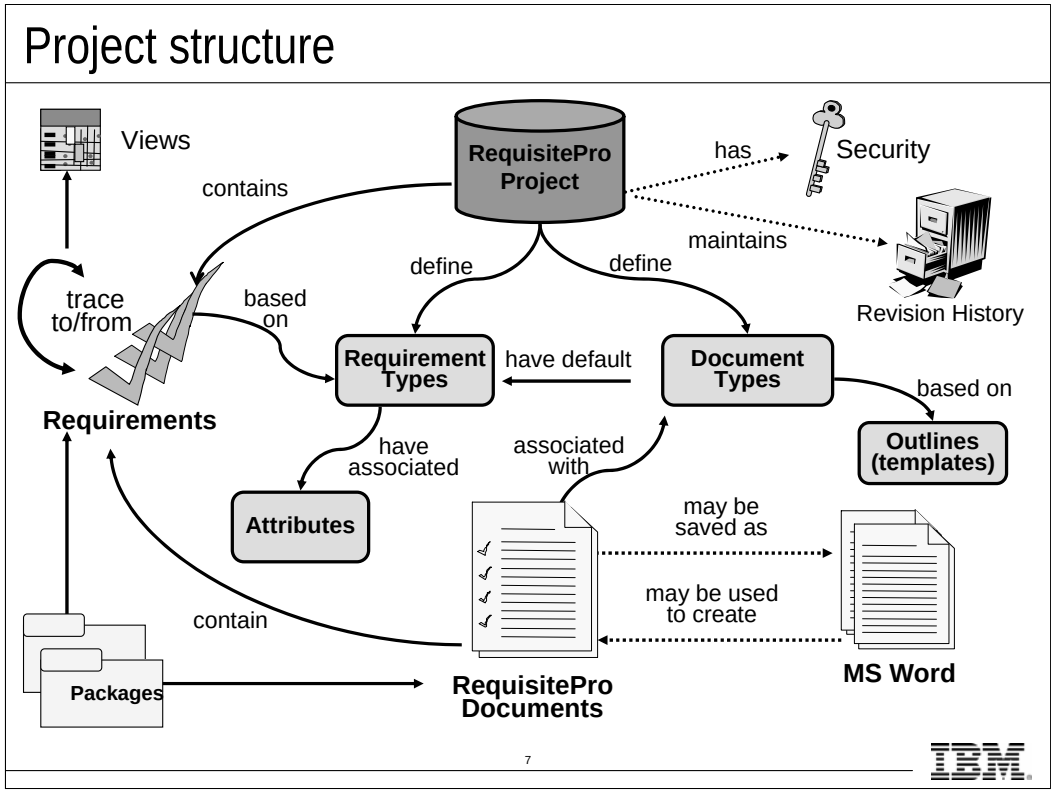
RUP requirements reference workflow

- Analyze the problem
- Understand stakeholder needs
- Define the system
- Manage the scope of the system
- Refine the system definition
- Manage changing requirements



6

Rational RequisitePro is used to manage the artifacts in each activity in the Requirements Discipline. Rational RequisitePro provides a groundwork for organizing and efficiently managing requirements and project document information.



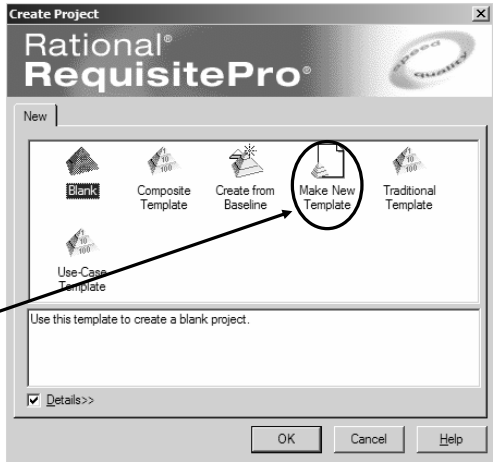
Your Requirements Management (RM) Plan dictates your project structure in RequisitePro. Your RM Plan specifies the types of requirements that you want to capture, the relationships between the requirement types, and the attributes you want to capture with each requirement.

This slide shows a high-level overview of all the components in a RequisitePro project and how they relate to each other. Requirement types, attributes, and document types define RequisitePro project structure.

Every requirement is associated with a requirement type. All requirements are maintained in the project database but can be located in documents as well. Requirements may have relationships among or dependencies upon one another. They can be traced from one requirement to another.

Project templates apply predefined project structure

- Templates include default
  - Document Types based on outlines (Word templates) with default Requirement Types.
  - Requirement Types with associated attributes.



Define a template

Templates can include artifacts, such as a glossary.

8

IBM

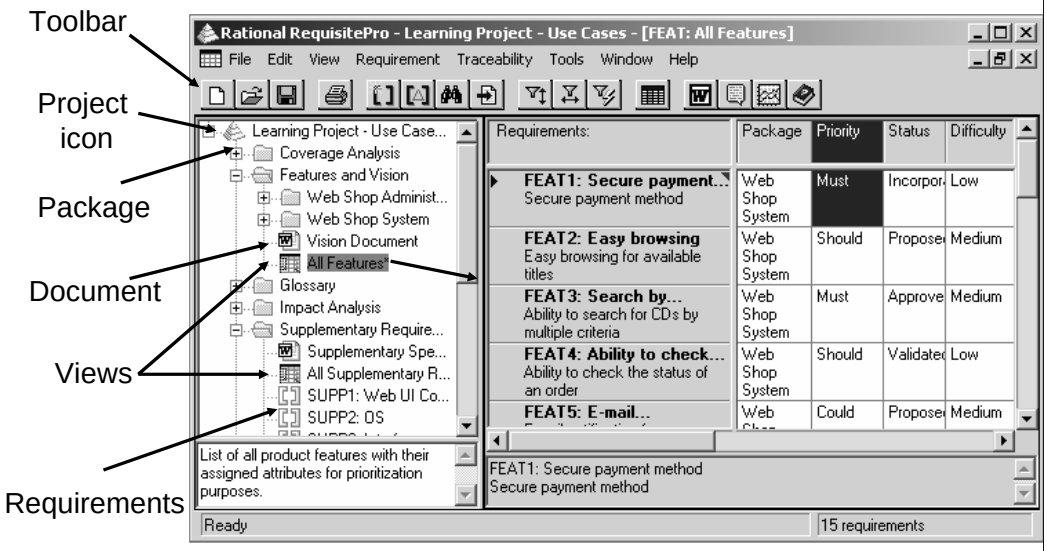
You use a project template to create a new project. You can use one of the templates packaged with Rational RequisitePro that most closely resembles the structure you determine in your RM Plan, or you can make your own template. Templates include document types, requirement types, attributes, and security settings for a project.

Packaged project templates include:

- **Blank**: Empty project
- **Composite Template**: A combination of the Use-Case and Traditional templates
- **Traditional Template**: Traditional requirements capture strategy
- **Use-Case Template**: Uses a use-case methodology

You can also create a project using **Create from Baseline** that creates a new project using an existing project baseline.

RequisitePro client interface organization



9

IBM

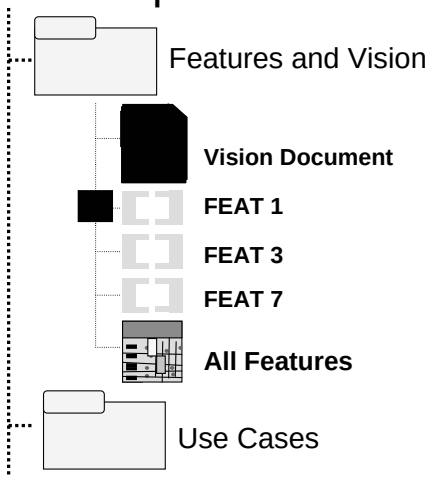
The left pane is called the Project Explorer (sometimes called just Explorer). The right window is where all views are displayed. A **view** is your window into the project requirements database displayed as matrices.

Documents are opened in a separate Word window that includes the RequisitePro menu option.

You can toggle between the Word and Rational RequisitePro to organize and prioritize your requirements, trace relationships among them, and track requirement changes.

Packages

- Visually organize your project in Microsoft® Windows® Explorer
- Contain related requirements information



10

Packages provide a way for you to visually organize your requirements, requirements artifacts, and views into related groups.

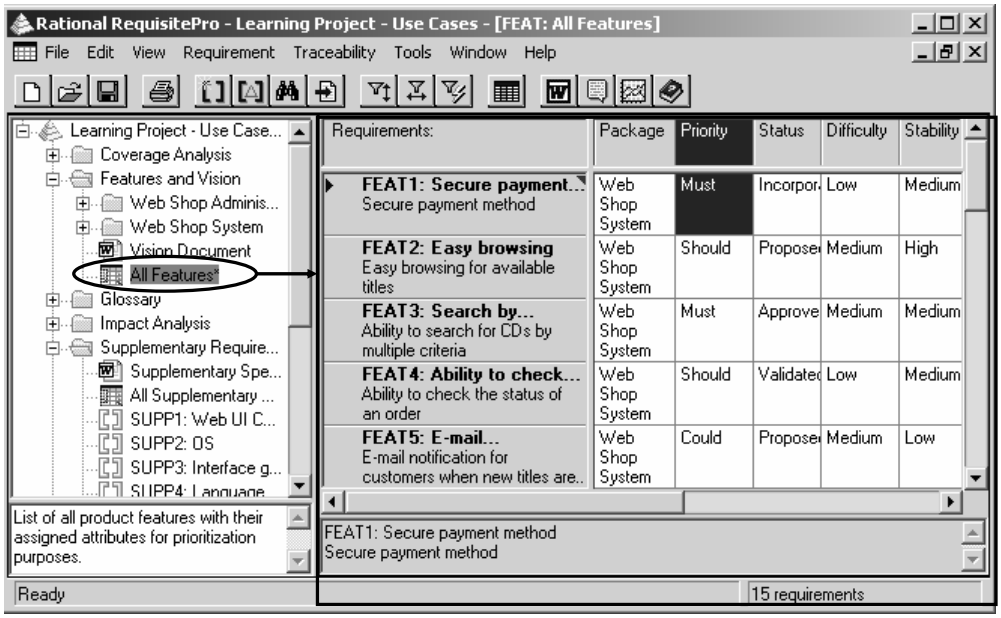
Views

- Access database information by Requirement Type.
- Keep you organized and on track.
- Analyze data.
  - Control feature creep.
  - Provide Coverage and impact analysis.
- Display data in a matrix or tree.

11

A view in Rational RequisitePro is an area where you create, analyze, and print requirements information. Views display and allow you to manage requirements in the database. The types of views are discussed in depth in Module 3.

Working in a view

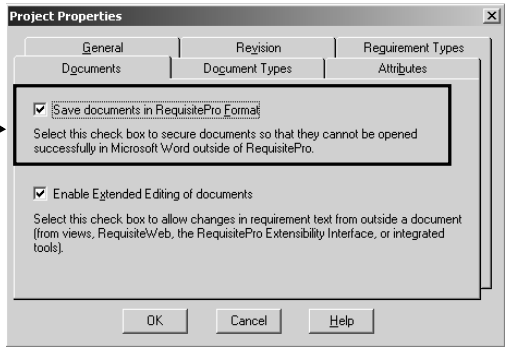


A view lets you work with the requirements directly in the database. You can modify requirement attributes, traceability, and text from within a view.

Views are created in the Explorer. You can open a view by double-clicking the view name. The view is opened in RequisitePro.

RequisitePro documents

- Are part of the RequisitePro project.
  - Requirements are maintained in database.
- Are maintained using Word within RequisitePro.
- Saved in RequisitePro format.
  - Configurable.



13

Documents are the natural format in which you capture requirements, providing context and supplementary information through a familiar Microsoft Word interface.

The costs for education and ramp-up time are significantly reduced when using Microsoft Word. Rational RequisitePro fits in your existing environment.

RequisitePro provides a robust architecture that maintains documents that “tell the story“ of requirements in context. The documents are dynamically linked to a database for powerful sort and query capabilities and effective requirements management.

RequisitePro provides outlines of standard documents for you to use in your project. The outline templates provided in RequisitePro use the use-case approach and comply with the Rational Unified Process (RUP).

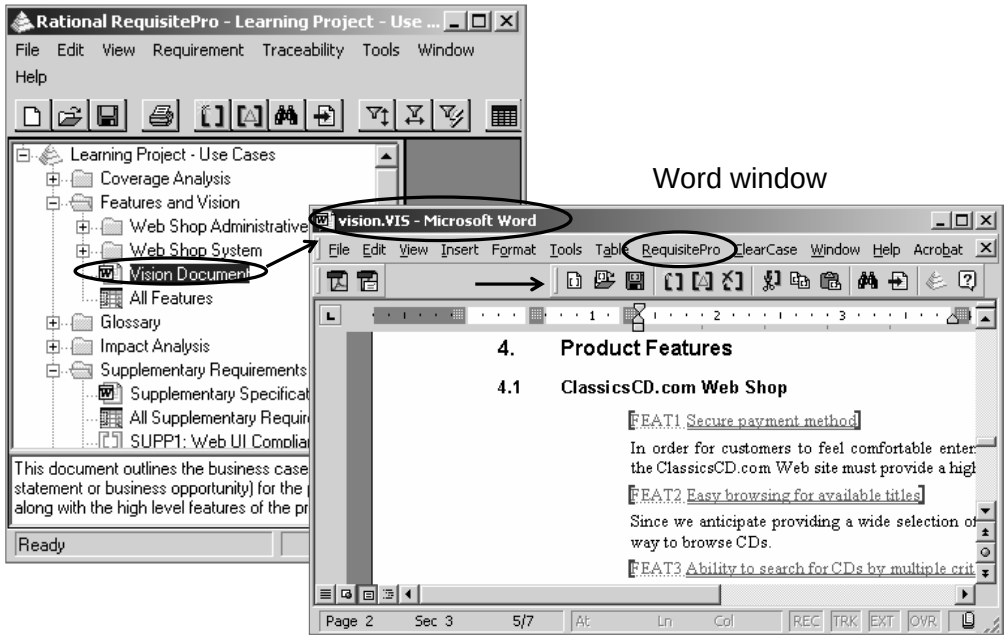
RequisitePro documents support industry standards, such as:

IEEE - Institute Electrical and Electronics Engineering

CMM - Capability Maturity Model

ISO - International Standardization Organization

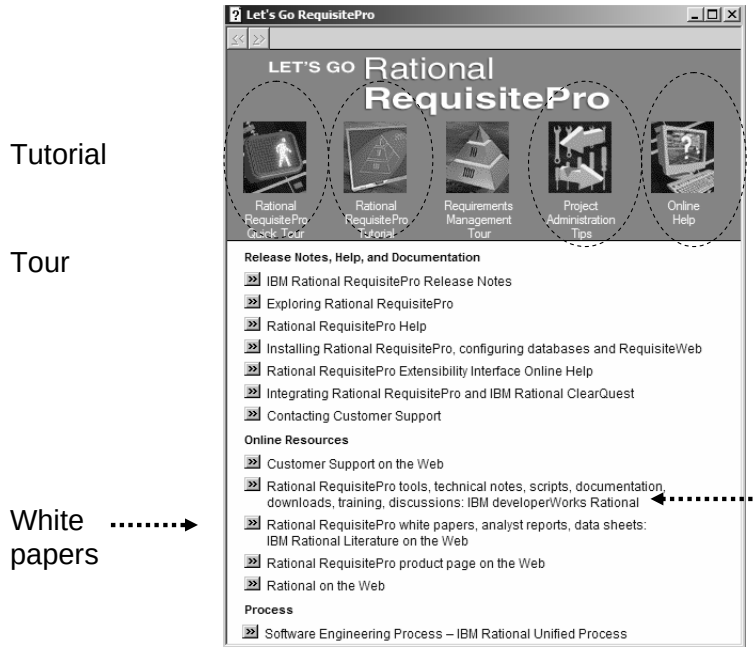
Working in a RequisitePro document



You view, create, and modify requirements documents in Word. RequisitePro uses the Word functionality to provide powerful features for viewing, editing, and formatting documents.

The Word menu bar includes a RequisitePro menu for managing RequisitePro requirements and documents from within Word.

Getting started with Rational RequisitePro



Tutorial

Tour

White papers

Admin tips

Help

developerWorks

http://www.ibm.com/developerworks/rational/

15

Let's Go RequisitePro is a Help interface that launches at startup (when configured to do so in the **Options** dialog box). It is also accessible through the Help menu.

It provides links to Help, white papers, and external resources such as the developerWorks to help you in your development activities.

Lab 1: Getting Started

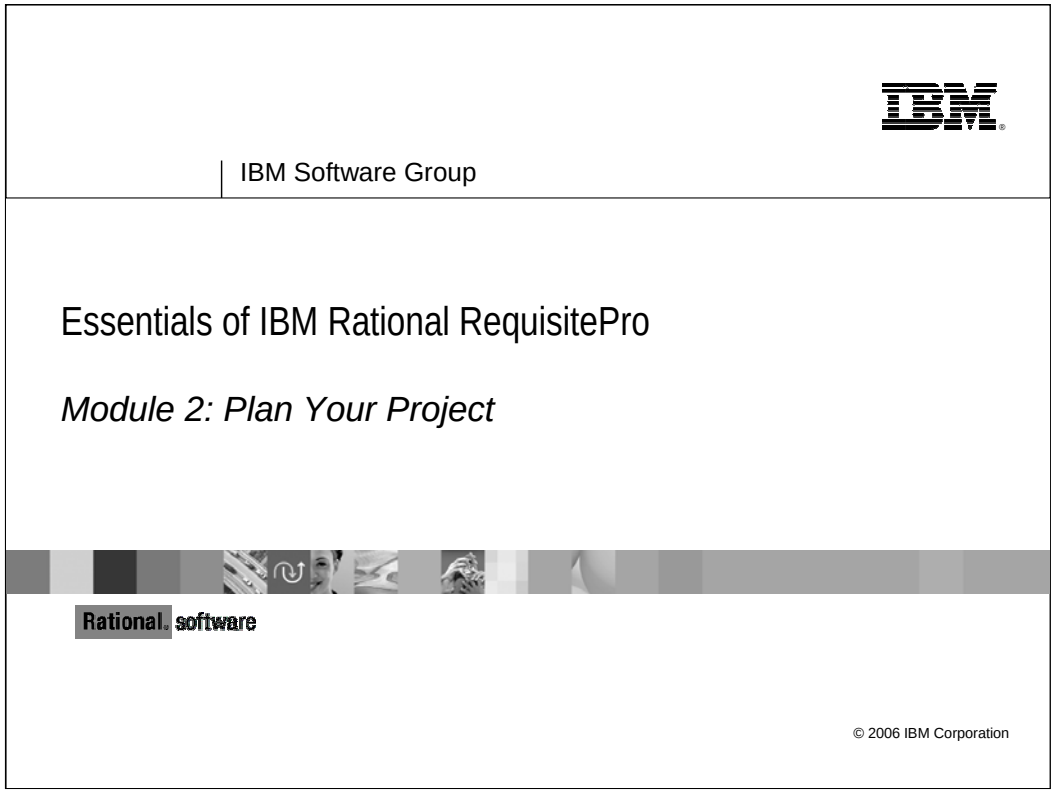
- Start Rational RequisitePro.
- Open an existing project.
  - The Learning Project – Use Cases.
- Browse project information.
- Explore Help.
- Set configuration options.

16

IBM

See Student Workbook Lab 1.

Goal: Become familiar with Rational RequisitePro and its structure by navigating through a sample project.



Contents

Objectives: Plan your project	2-2
RequisitePro requirement	2-3
Requirement attribute	2-4
First steps in defining your RM strategy	2-5
Requirements management plan	2-6
Create a new project	2-7
Requirement types	2-9
Document types	2-13
Additional project logistics	2-17
Lab 2: Customize Project Structure	2-20

Objectives: Plan your project

- Define basic terms used in RequisitePro.
- Define your RM strategy.
  - Identify project artifacts that help you plan and define the project
- Describe components for project structure.
  - Document types
  - Requirement types
  - Attributes and their values
  - Traceability criteria

2

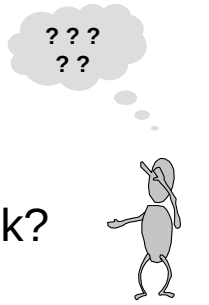
When defining the problem, you start to plan the project.

In the **Analyze the Problem** activity you establish your requirements management plan. With this plan, you are then able to establish the RequisitePro project structure.

RequisitePro helps you create and organize the basic artifacts necessary to start a project.

RequisitePro requirement

- Any tracked item
  - Inputs and outputs to the system
  - Functions of the system
  - Attributes of the system and its environment
  - Features
  - Use Cases
  - Supplementary requirements
  - Stakeholder requests



What project requirements will you track?

3

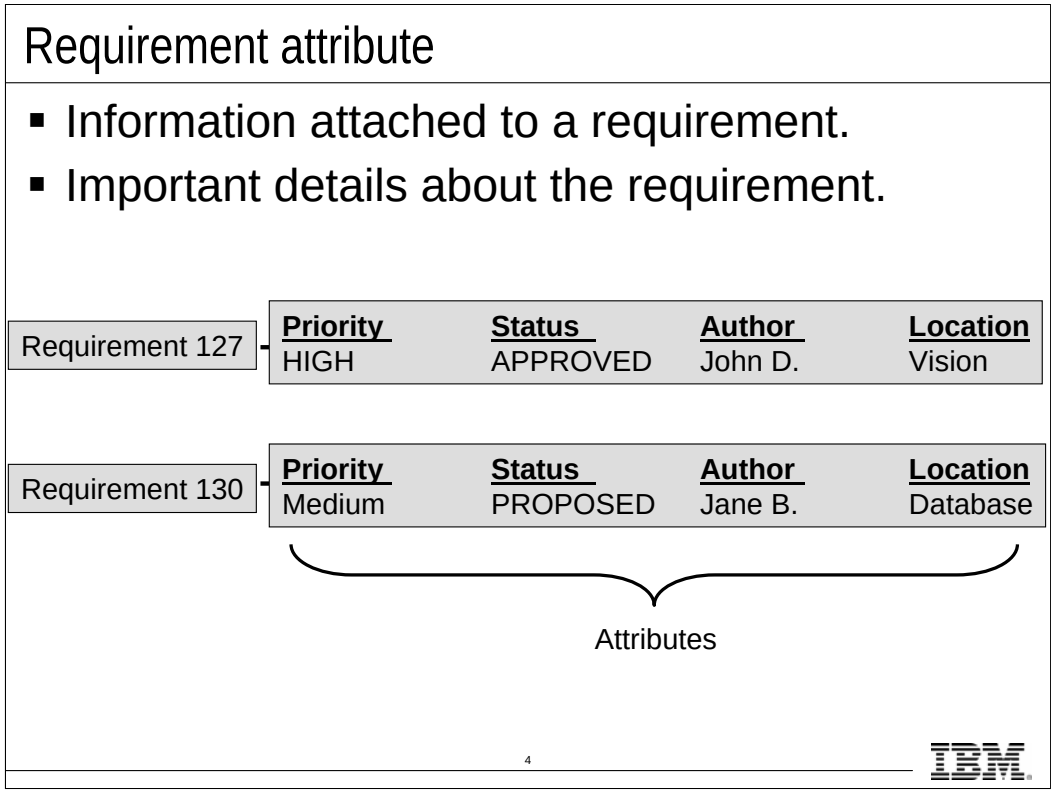
IBM

A requirement is defined as a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document.

RequisitePro is flexible enough to allow you to define any item you need to track as a requirement.

After you have created a requirement in a RequisitePro project, you can:

- Qualify the requirement by assigning attributes.
- Trace the requirement to and from other requirements.



Attributes are details about each requirement that are used to manage the requirements throughout the lifecycle of the project.

Attributes are defined per Requirement Type. Each attribute is assigned values.

RequisitePro has two types of attributes that are helpful to you.

1. **User-defined** attributes are defined by the user.

   Examples: Priority, Status, Risk, Stability

   Any of these attributes may be modified or deleted based upon the needs of your project.
2. **Read-only system** attributes are created and defined by RequisitePro.

   Examples: Location, Author, Date, Revision Number

Note: Read-only system attributes cannot be modified or removed by users.

An attribute value is information assigned to a requirement. Attribute values can be text or numbers.

For example, the attribute Priority may be assigned the values of Low, Medium, or High.

First steps in defining your RM strategy

- Name the project.
- Give a brief description of the project.
- Plan the project structure by drafting a Requirements Management Plan.

5

IBM

To define the project, give it a name and a location. Determine a location for the project repository where all team members can access the data.

If you wish you can give the project a brief description for informational purposes.

The RequisitePro default database is Microsoft® Access®. .

RequisitePro is also compatible with IBM® DB2® Universal Database, Oracle, and Microsoft SQL Server.

Choosing a database:

Use the following criteria when deciding which database to use with RequisitePro:

- Microsoft Access is recommended for use in small work groups, with fewer than 10 concurrent users.
- If your team is distributed across states or countries, use DB2, SQL Server, or Oracle. These enterprise databases provide socket-level access to remote network locations, which tend to perform better.
- Use DB2, SQL Server, or Oracle if you anticipate having more than 10 users logged on at one time. Tests have shown that the performance of Microsoft Access decreases after ten simultaneous logged-on users.
- These numbers may vary with network bandwidth. Use DB2, SQL Server, or Oracle if you will be managing large numbers of requirements (tens of thousands).

Requirements management plan

- Provides outline of project structure.
- Organizes your project and requirements.
  - Describes management strategy.

RM Plan

I. Introduction
II. Requirements Management
    Organization
III. Requirements Management Program
    Requirements Identification
    Traceability criteria
    Attributes
IV. Milestones
V. Training and Resources

6

A Requirements Management Plan records decisions about requirements and requirements management. A Requirements Management Plan is typically developed early in the project and refined throughout the project.

The decisions include structure and content of your requirements information, such as:

- The types of requirements
- The attributes associated with each Requirement Type

The management information includes:

- Information for measuring, reporting, and controlling changes to the product requirements
- Traceability criteria

The plan is organized for:

Managing requirements by type

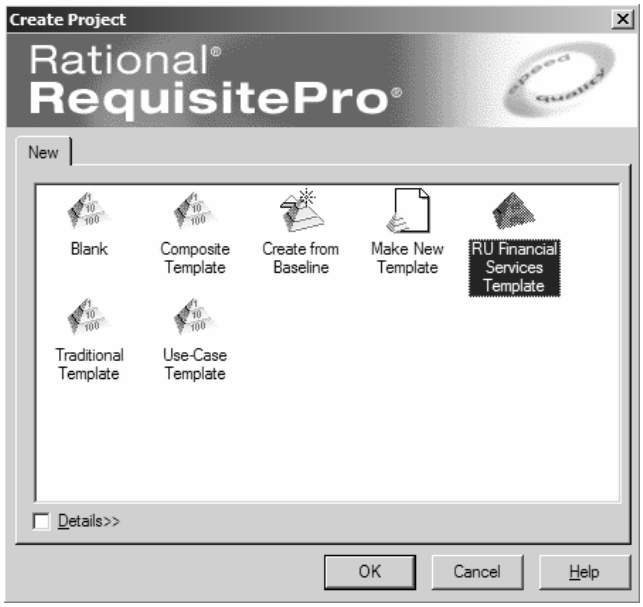
Querying project data by attribute values:

- Status
- Priorities

Viewing project requirements metrics and reports

Creating requirements traceability

Create a new project



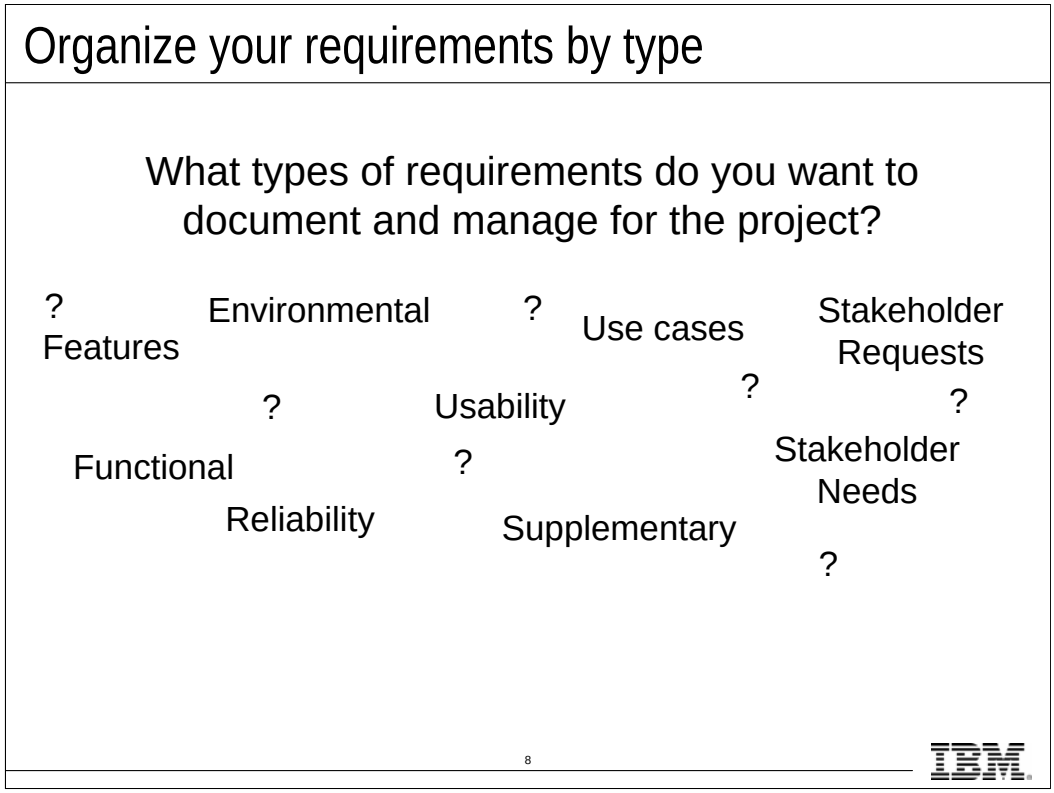
7

IBM

A new project is created from a project template.

You may find that creating a company project template is useful. The example shows an RU Financial Services Template. Each new project created for the company should use the default structure from the RU Financial Services template and then be customized as necessary.

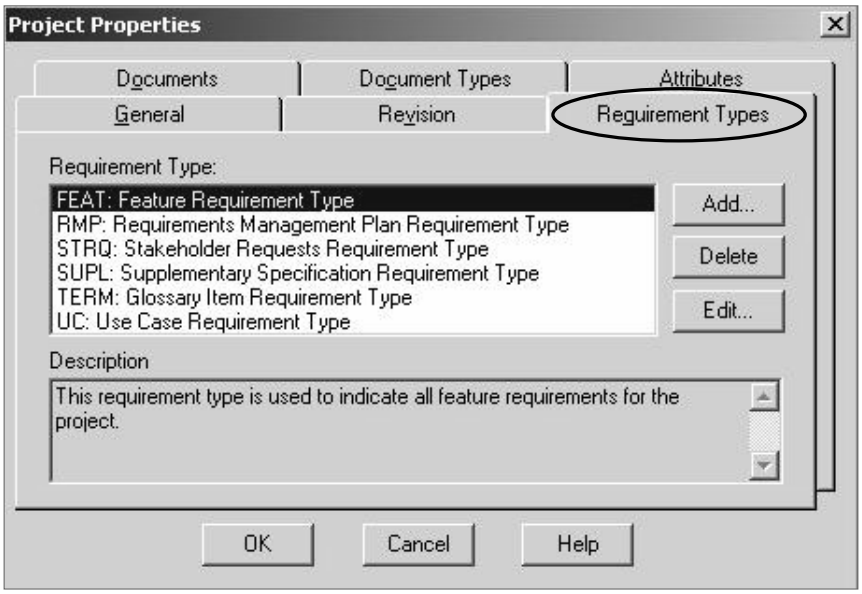
You can also create a new project from an existing project baseline. You create a RequisitePro baseline of a RequisitePro project, then use that baseline to create new RequisitePro projects. For example, you may want to base the next release of a project on a stable configuration of the previous release. You will learn more about RequisitePro baseline in **Module 4 Manage Your Requirements**.



What types of requirements do you want to document and manage in your project? Depending upon the type of system you are developing, you will choose different Requirement Types that you want to capture.

Each project has different needs. A project that involves the development of hardware and software will probably want to capture both hardware and software requirements.

Requirement Types



9

IBM

A Requirement Type:

- Is a set of descriptive and operational information associated with a requirement.
- Serves as a template for your requirement.
- Enables you to organize requirements by type at a higher level.

In defining requirements and their style, consider the following questions:

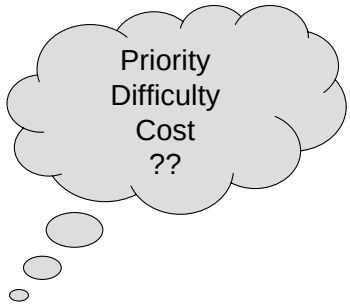
- Will they contain a specific word or format?
- How do you want your requirements identified?
- What types of requirements do you want to capture?

To set up Requirement Types select:

1. Click **File > Project Administration > Properties**.
2. Click the **Requirement Types** tab.

Empower your project with requirement attributes

- Define attributes by Requirement Type.
  - What information do you want to track?
- Use requirement attributes to:
  - Assign resources
  - Assess status
  - Calculate software metrics
  - Manage project risk
  - Estimate costs and time
  - Manage project scope
  - Prioritize requirements



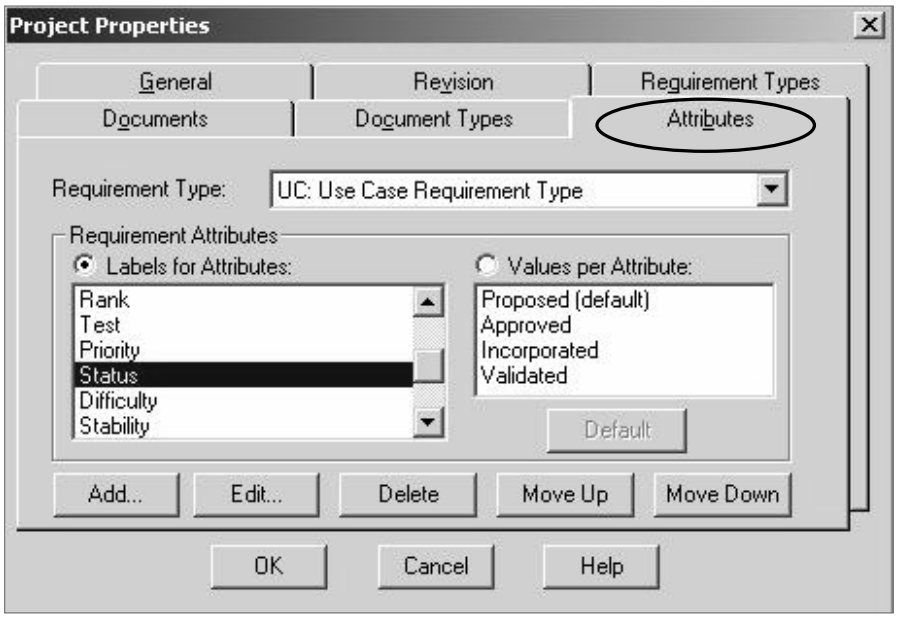
10

When setting up a RequisitePro project, you need to consider the attributes that you want to collect for each Requirement Type. Next you must document the attributes that you want to collect in your Requirements Management Plan.

Tip: A minimalist approach is best. The more attributes you choose, the more work will be required to maintain them. As soon as your attributes lack maintenance and go out of date, all of your attributes become useless. That is because you have no way of determining which attribute values are correct and which are not.

Be sure to choose your attributes so that you are able to get useful information.

Requirement attributes for each Requirement Type



11

IBM

Once your Requirement Types have been identified, determine what management information about each requirement is needed. Store information as attributes about a particular type of requirement.

Attributes are either system attributes (defined by RequisitePro) or user-defined (defined by the project owner).

A requirement attribute:

- Provides information to manage a requirement.
- Helps a team plan, communicate, and monitor the project.

To configure attributes, click **File > Project Administration > Properties**, and then click the **Attributes** tab.

Organize your project artifacts

- Define the types of documents that you want to create:
  - Glossary
  - RM Plan
  - Vision
  - Supplementary Specification
  - Use Cases
- Identify which Requirement Type will be captured in each Document Type:
  - TERM ➔ Glossary
  - RMP ➔ RM Plan
  - FEAT ➔ Vision
  - SUPL ➔ Supl. Spec.
  - UC ➔ Use Case

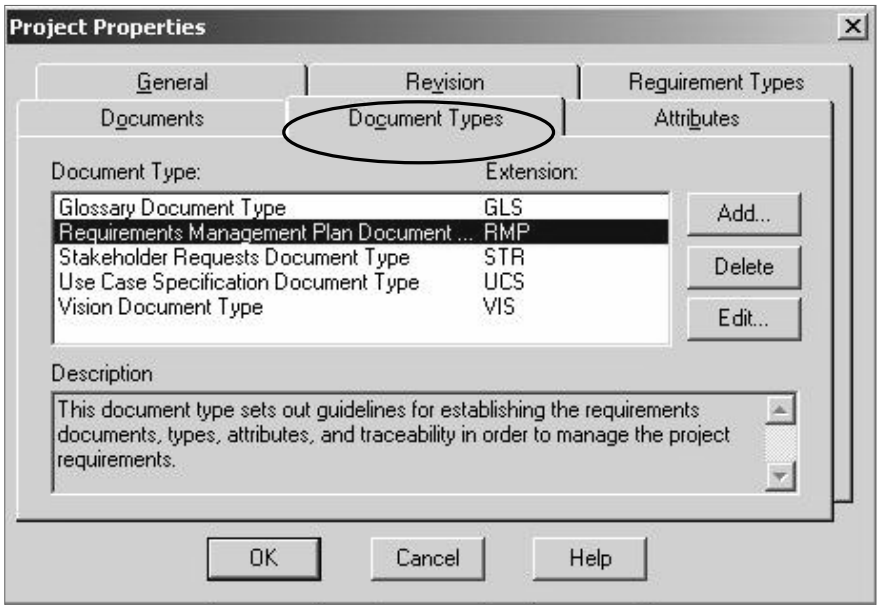
12

RequisitePro enables you to store all of your documents in the project repository. This means that your stakeholders have a single place to go when they want to locate any document related to the project.

Part of establishing your RequisitePro project requires you to predetermine the types of documents you want to store in your project.

Each different Document Type is associated with a default Requirement Type.

Document Types



13

IBM

A Document Type is a definition for creating new documents. For example, a Use Case Specification Document Type is a definition of the document for specifying a use case. A document type has a standard format and predefined text.

A Document Type is based on an outline or document template. Document templates are created in Microsoft Word.

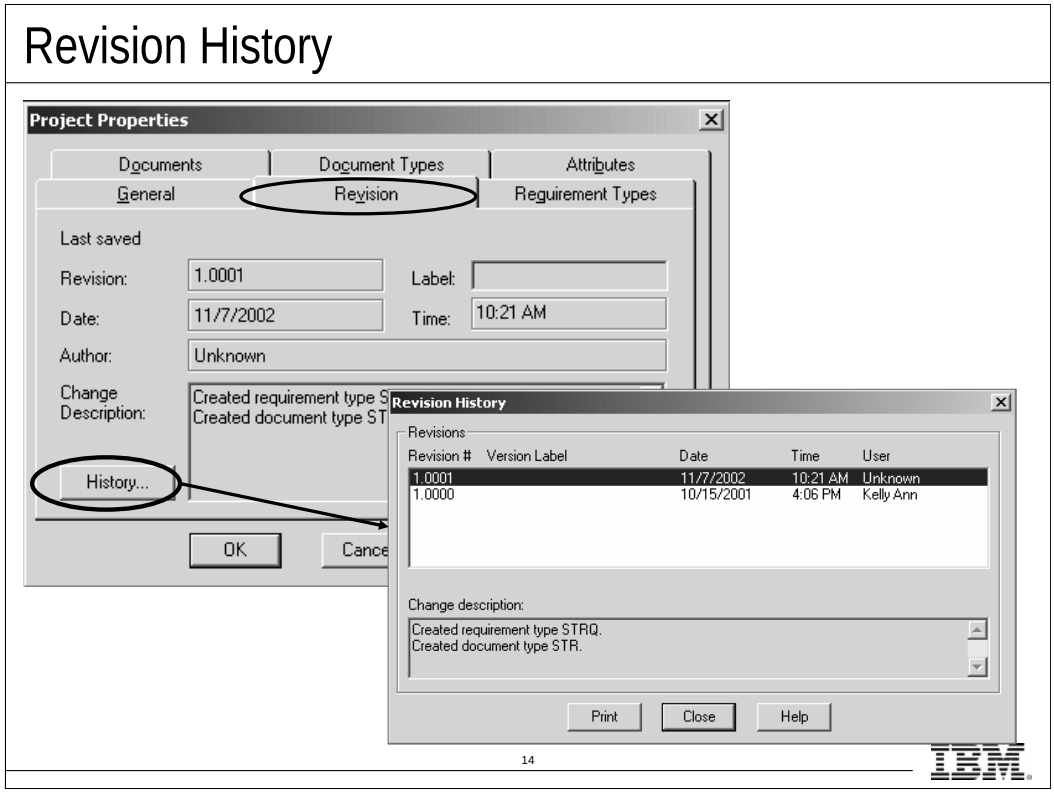
A Document Type also includes the selection of default requirement types. **Default** means that when a new requirement is created in a document, its type will be the default requirement type for that document type. A document can contain requirements of many different types. If no requirement type is chosen when a requirement is created, then each new requirement automatically becomes the default requirement type.

Benefits of standard document types include:

- They provide a good starting point.
- They apply consistent formatting to documents of the same type.

RequisitePro does not require the use of documents in a project. Requirements can be added directly into the database with no associated documentation. During project planning and assessment, the team must decide whether documents will be used in the project.

To configure document types, click **File** > **Project Administration** > **Properties**, and then click the **Document Types** tab.



RequisitePro supports change management using revision numbers and revision labels for the following: individual requirements and documents within projects, and RequisitePro projects. If you change a requirement name, text, or attributes, RequisitePro increments the requirement revision number. Also, when you modify a project or document, Rational RequisitePro creates new revision information. You can view and print the revision history for the project, documents, or requirements.

Depending on the revision history you want to view, do one of the following:

Projects

Select the project in the Explorer, and click **File > Properties**.

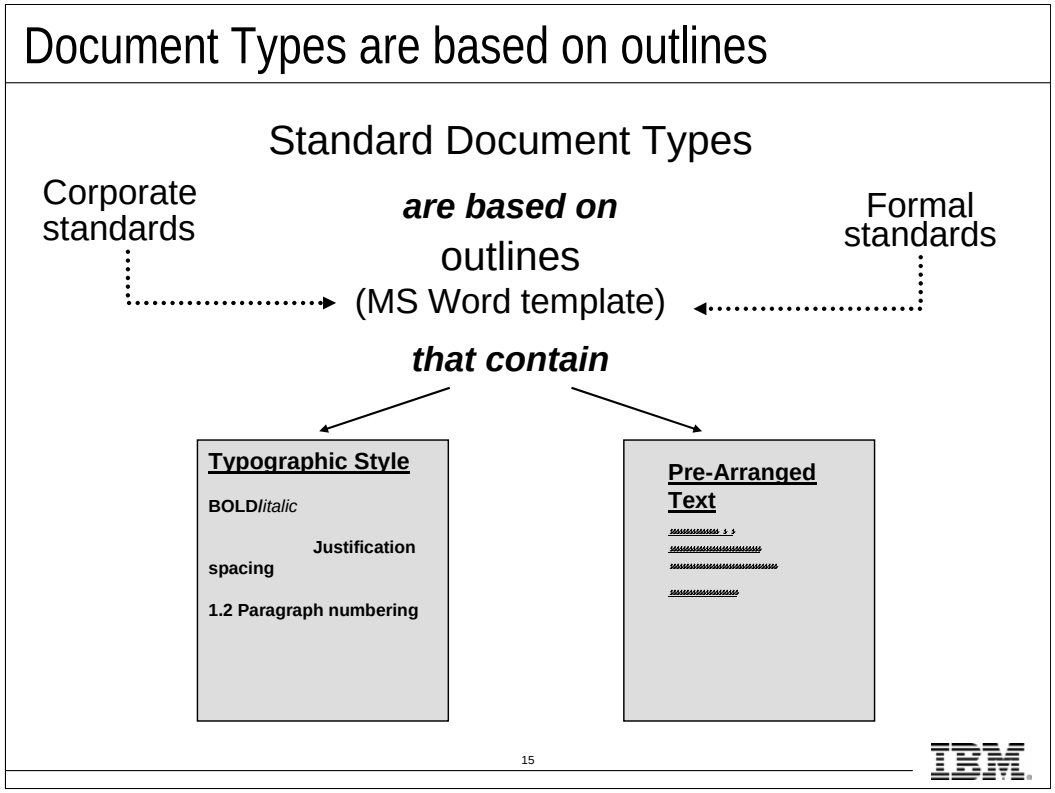
Documents

Select a document in the Explorer, and click **File > Properties** or

In Microsoft Word, click **RequisitePro > Document > Properties**.

Requirements

Select a requirement in the Explorer, and click **File > Properties**.



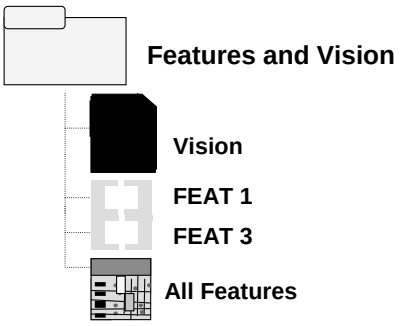
RequisitePro provides some basic document outlines to help your development process. RequisitePro includes outlines to support both the traditional or IEEE development methodology and the use-case development approach.

RequisitePro also provides the capability to capture your own corporate standard templates and use them to define RequisitePro documents.

Some companies have their own standard requirement templates that they choose to use when specifying requirements. These can easily be added to your collection of RequisitePro document templates. Packaged outlines are located in the Outlines directory in RequisitePro.

Package in RequisitePro

- Is a container that can include:
  - Requirements
  - Documents
  - Views
  - Other packages
- Organizes related artifacts visually.
- Is shared by all project users.



16

IBM

Packages provide a simple and effective way to visually organize your requirements. Organizing your requirements using packages is similar to how you organize the files on your hard disk using directories.

Packages enable you to group requirements in a convenient way. For example, you could create a package for each use case. The requirements for each use case would be grouped in a single package, thereby enabling you to quickly locate them.

To create a package, click **File** > **New** > **Package**.

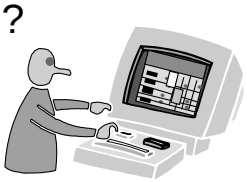
There are ways to move artifacts between packages:

- Drag and drop the artifact in the Explorer from one package to another.
- Modify the assigned package in the **Properties** dialog box.

To delete a package, make sure that it is empty, then select it and click **Edit** > **Delete**. The package is removed from the Explorer and the project database.

Additional project logistics

- User environment
  - Multi-user? Stand alone? Web-based?
- Establish the project repository.
  - Choose a database:
    - Access, IBM® DB2®, Oracle, SQL Server
    - Determine the server location
- Security and Permissions
  - Consider accessibility needs.
  - Determine the type of user authentication required
    - RequisitePro user authentication
    - Lightweight Directory Access Protocol (LDAP)



Project Administrator

17

IBM

To ensure success in the project development stage, an assessment and some initial planning are necessary. You also need to determine the current status of the work environment. In this phase, it is helpful to determine goals, review existing components, and assess resources.

Consider the following when you are initially planning the project:

Development Method

Does your project adhere to an overall governing standard such as a Use Case, IEEE, CMM, or ISO9000 standard?

User Environment

How will the project be deployed in the user environment? Stand-alone? Networked? Distributed?

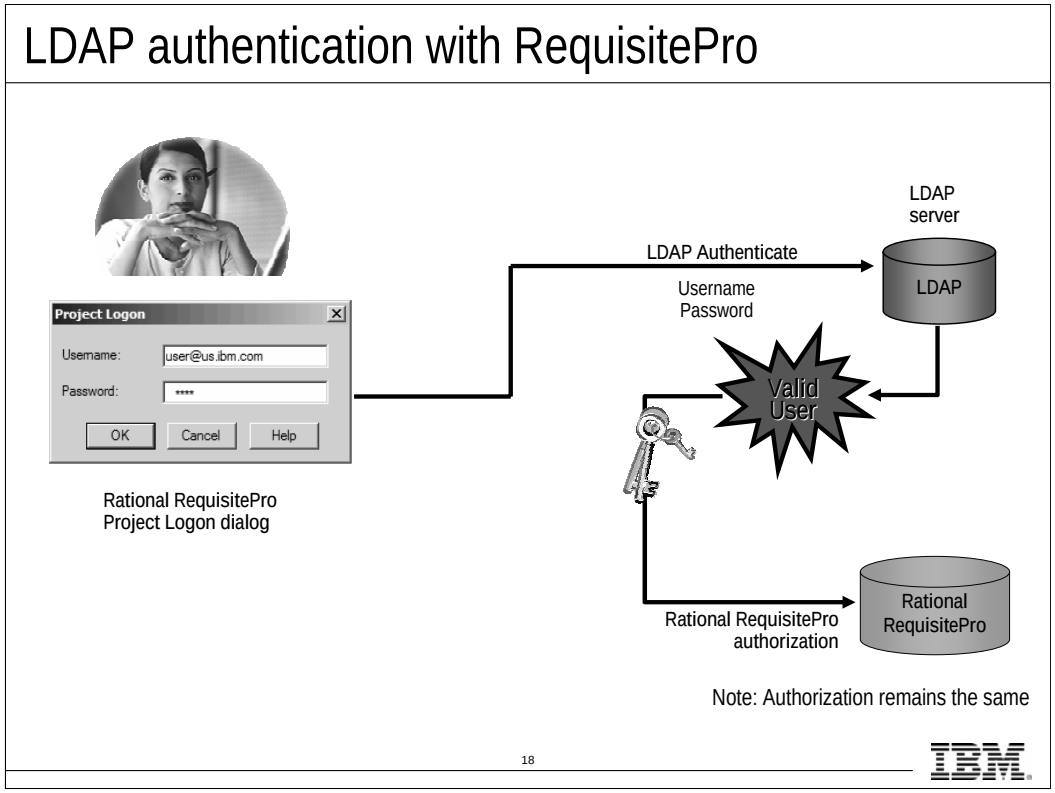
Requirements Capture Techniques

Will you capture your requirements in documents, in a database, or in both?

Security

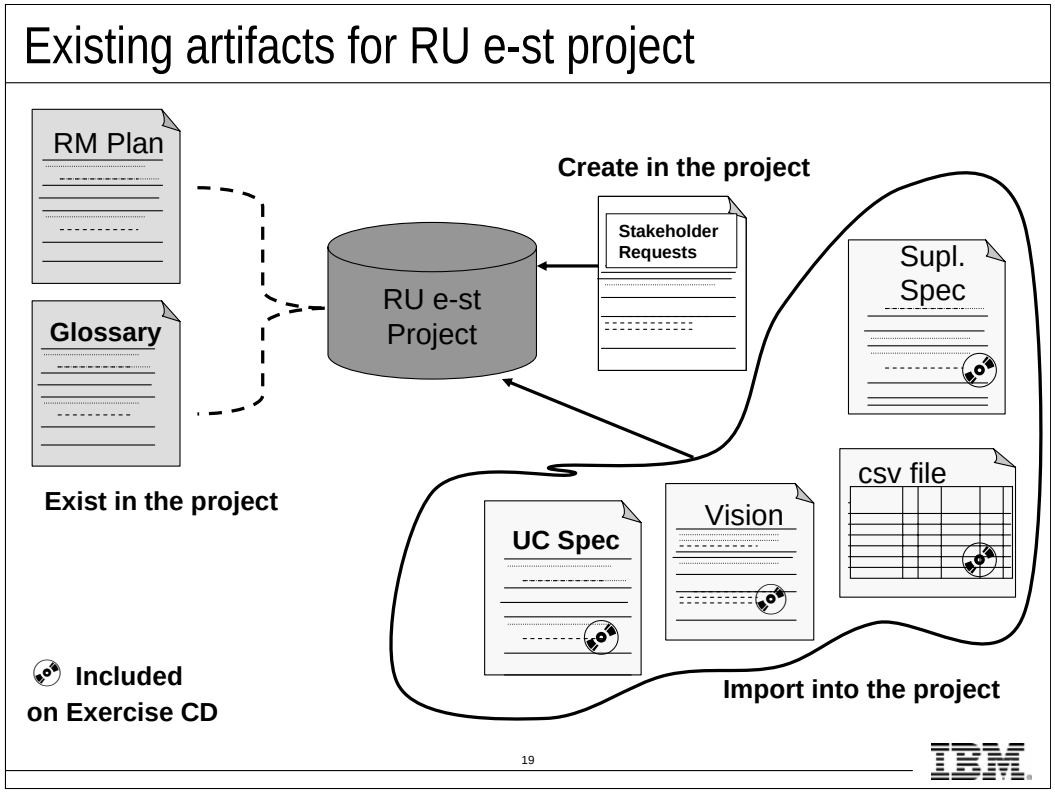
Determine specific permissions that define the kind of access users will have to the project. Security is crucial for projects with multiple users. If security is not enabled for the project, any user can open the project. If security is enabled, you must decide whether to authenticate users using RequisitePro project user records only or using LDAP authentication.

Using LDAP authentication decreases administrative costs by centralizing user records for multiple applications and reduces the number of passwords users must remember. It can also improve security by enforcing the password management policies implemented in the directory.



If you are using RequisitePro to authenticate users, the user enters a username and password. RequisitePro verifies that there is a matching username and password stored in the RequisitePro project database.

If you are using LDAP authentication, RequisitePro user information is mapped to an LDAP directory. User passwords are not stored in RequisitePro. The user enters a username and password, and then RequisitePro checks an LDAP directory for a matching user record first. If LDAP authentication succeeds, then the user name and password are authenticated against the RequisitePro project database.



- RU e-st project files on the exercise CD.

For the purpose of this class, you have been asked to build the software for an electronic stock trading system.

Each lab builds the system project as you gain more information and walk through the Requirements discipline as defined by the Rational Unified Process.

Lab 2: Customize Project Structure

- Read Initial Requests document (optional).
- Open the RequisitePro project (from CD).
- Customize project structure.
- Modify project organization.
  - Create packages.

20

IBM

See Student Workbook Lab 2.

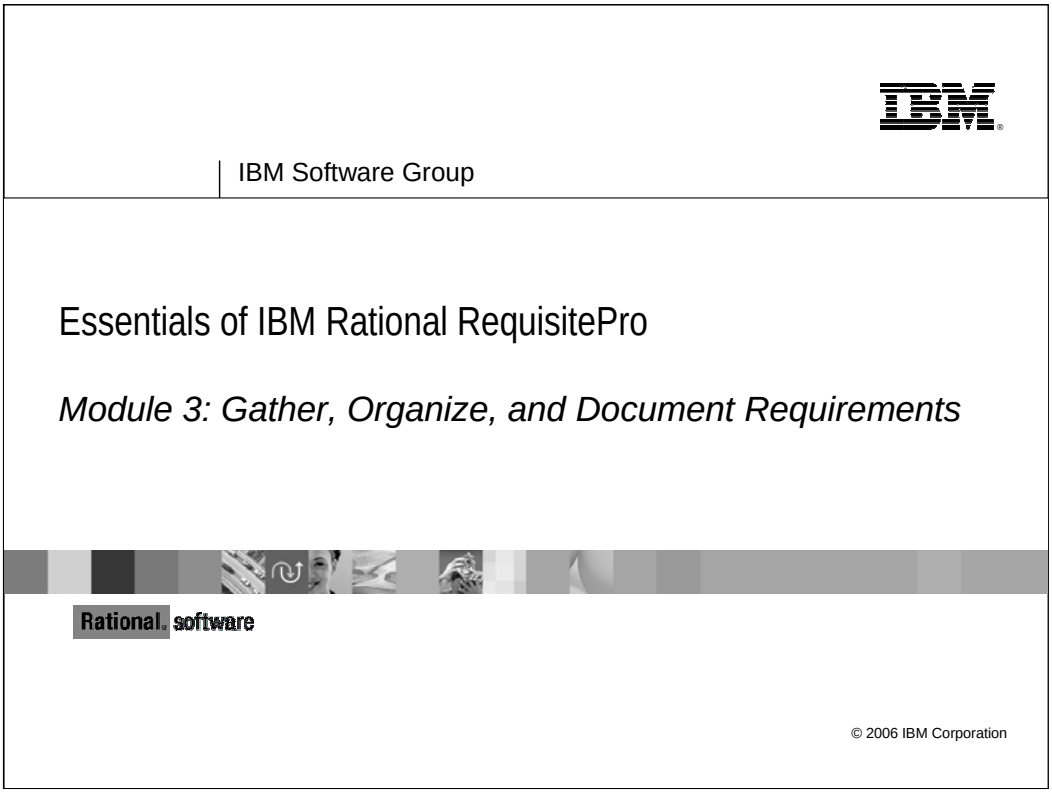
Goal: Add the RU e-st project.

- Customize project structure.
  - Requirement Types
  - Attributes
  - Document Types
    - Outlines

Questions for planning your project:

How do you want to manage the requirements?

What information is important to each team member?



Contents

Objectives: gather, organize, and document requirements	3-2
Requirement artifacts set	3-3
Requirements in RequisitePro	3-5
Requirements in a view	3-6
Edit a requirement	3-9
Import requirements from a CSV file	3-10
Delete a requirement in the Explorer	3-11
Lab 3: Gather and Enter Requirements	3-14
Requirements and associated documents	3-15
A requirements document in RequisitePro	3-17
Edit a requirement in a document	3-22
Import a document with the Import Wizard	3-23
Delete a requirement in a document	3-24
Offline authoring	3-26
Lab 4: Create and Import RequisitePro Documents	3-29

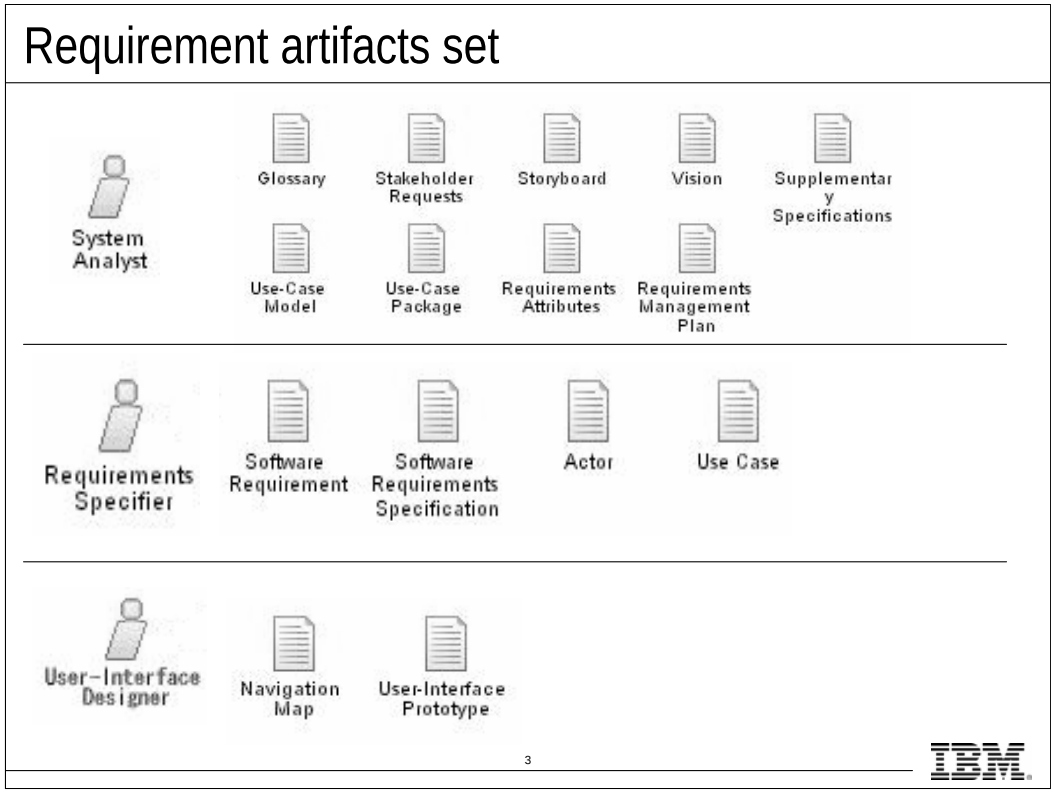
Objectives: Gather, organize, and document requirements

- Enter requirements directly into project.
- Import requirements.
- Edit and delete requirements.
- Assign attribute values.
- Recognize good requirement characteristics.
- Organize requirements.
  - Packages
  - Hierarchy

2

The first part of this module discusses requirements that reside only in the database, which are displayed and managed through views (matrices).

The second part of this module explores requirements located in documents using Word.



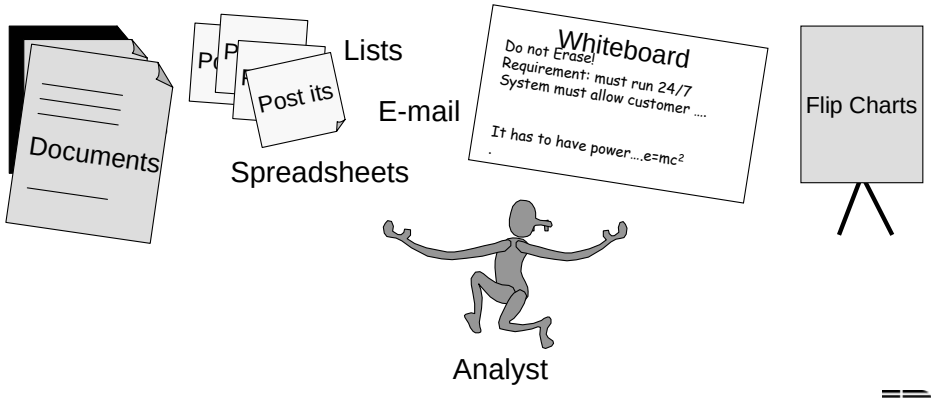
Many team members use RequisitePro to manage project artifacts in a software development project.

Each project has a strategy for organizing and managing requirements. This is one possible strategy for organizing requirements.

The examples above are from the IBM® Rational Unified Process® (RUP®).

Where are your requirements now?

- Analyze the problem to be solved.
- Understand stakeholder needs.
- Organize initial requirements created from elicitation.



4

IBM

Is there existing data outside of RequisitePro that contains requirements? If some of your requirements already exist outside RequisitePro, import them into RequisitePro.

When you plan a project, the information is gathered through meetings, discussions, brainstorming, and e-mails. From the gathered information, you start defining the project and its structure. This usually happens through draft plans (RM Plan and Vision doc).

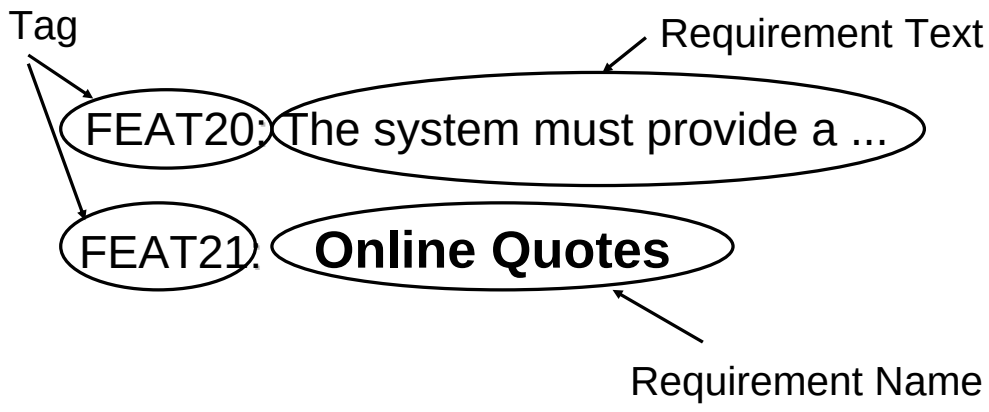
RequisitePro imports data into a project from files in Word format or Comma Separated Value format. CSV format is used by applications such as Microsoft Access and Microsoft Excel.

Where are your requirements captured?

- Post-its
- Documents
- Whiteboards
- Easel boards
- Spreadsheets
- E-mail threads
- Databases

Requirements in RequisitePro

- Must have Text or Name.
- Are uniquely identified by a Tag.
- Are organized by Requirement Type.



5

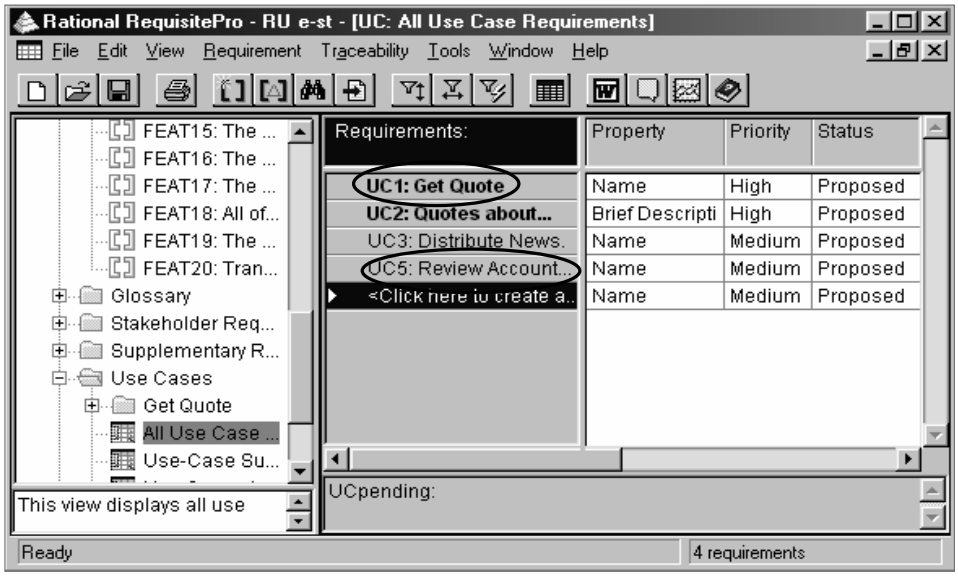
IBM

All requirements contain the **text** specification of a requirement and/or a short, descriptive requirement **name** (optional). Each requirement is uniquely identified by its requirement **tag**.

A requirement tag consists of the requirement type prefix and a unique number. For example, FEAT20 refers uniquely to the twentieth feature requirement.

A requirement name is a short descriptive name for a requirement. In a View and other locations where a requirement's tag is used as an identifier, RequisitePro will display its name. This allows users to refer to their requirements easily in conversations and find them easily in a View. If a requirement has a name, it will appear in the matrix view in **bold** type. Requirement names are optional, unlike the requirement text.

Requirements in a view

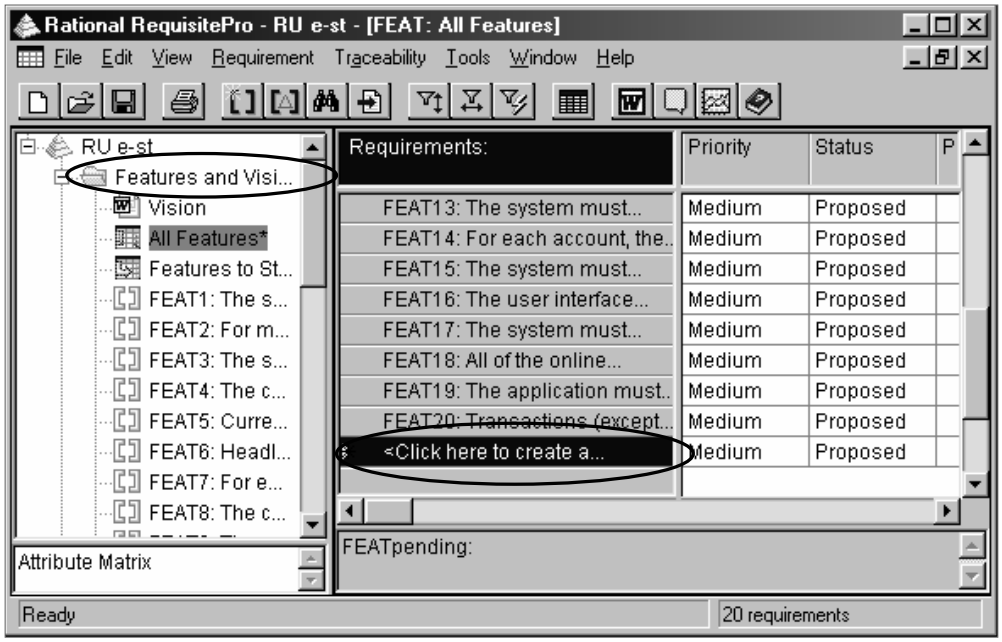


6

IBM

When working in a view, you can see the difference between a requirement text and a requirement name by the font. Requirement names are in bold. (The style is configurable.)

Create a requirement in the Explorer or in a view



There are four ways to create a requirement in a RequisitePro:

- Click the New Requirement row in a view.
- Click **Requirement > New**.
- Right-click a package in the Explorer, and then click **New > Requirement**.
- Use Word (covered later).

A requirement added directly to the database bypasses the documentation process and resides only in the database. Any changes made to a database requirement are done in a view.

In a view, find the attribute labeled “Location” to determine where the requirement resides.

When working directly with the database, you access a view in RequisitePro. This is where you perform requirement management activities.

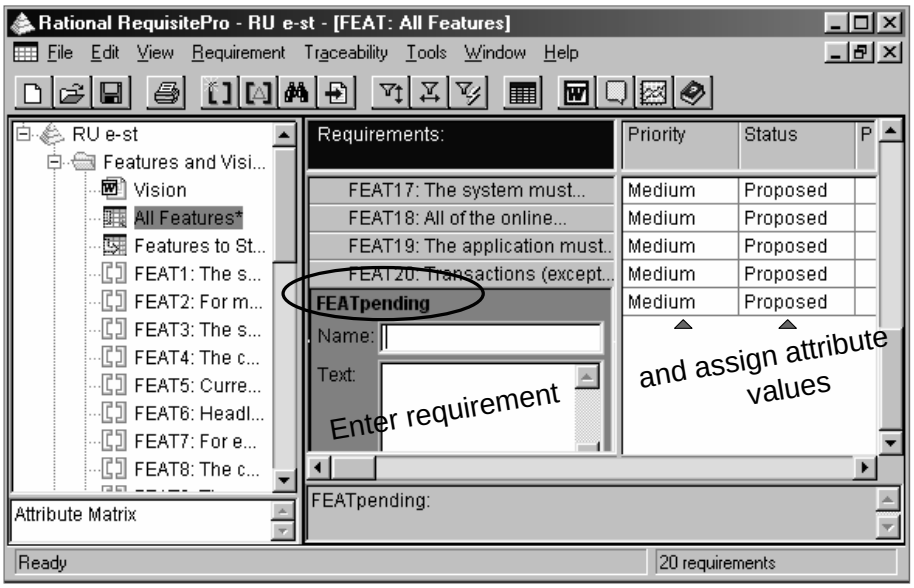
Views present information about requirements in a table (matrix) or in an outline tree.

RequisitePro has three view types:

- Attribute Matrix
- Traceability Matrix
- Traceability Trees

For now, the exploration is confined to the Attribute Matrix. This matrix displays all requirements of a selected requirement type, along with all associated attributes and values.

A newly created requirement is pending



8

IBM

The Attribute Matrix view allows you to enter a new requirement.

To create requirements in the Attribute Matrix:

1. Make sure that the **In-View requirements creation (without dialog box)** check box is selected. This check box is located at **Tools > Options**.
2. Click the last row near the statement **<Click here to create a requirement>**.
3. Click tab to modify or assign requirement attribute values.
4. Press ENTER to commit (save) the requirement. When the requirement is saved, it is assigned.

Edit a requirement

- In a view:
  - Click the requirement.

    or
  - Right-click and select **Properties**.
- In the Explorer:
  - Right-click and select **Properties**.
- From the RequisitePro menu:
  - Click **Requirement > Properties**.

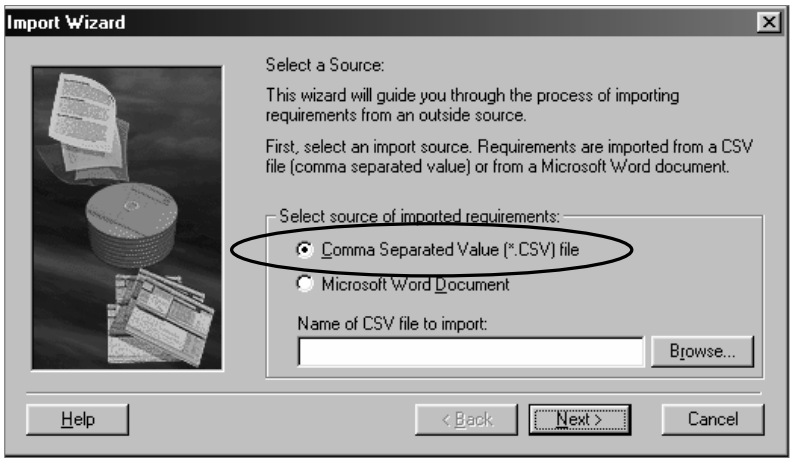
9

To modify a requirement in the database:

- If the **In-view requirement creation** option is enabled, click in the requirement. Edit the requirement directly in the matrix.
- Double-click the requirement. RequisitePro will take you to the location of the requirement to be modified.
- Click **Requirement > Properties** for the **Requirement Properties** dialog box.

Import requirements from a CSV file

- Requirement and attribute data are imported directly into the project database.



10

RequisitePro can import requirements and attributes into your project from any database that supports export of data in comma-separated value format, such as SQL Server, Oracle, Excel, or Access.

When a new requirement is added to the database, it is assigned a new root requirement number. If a requirement has a tag or a number in the external CSV file, that number is not preserved. RequisitePro recognizes a hierarchical numbering scheme and imports lower-level requirements as children of a higher-level requirement.

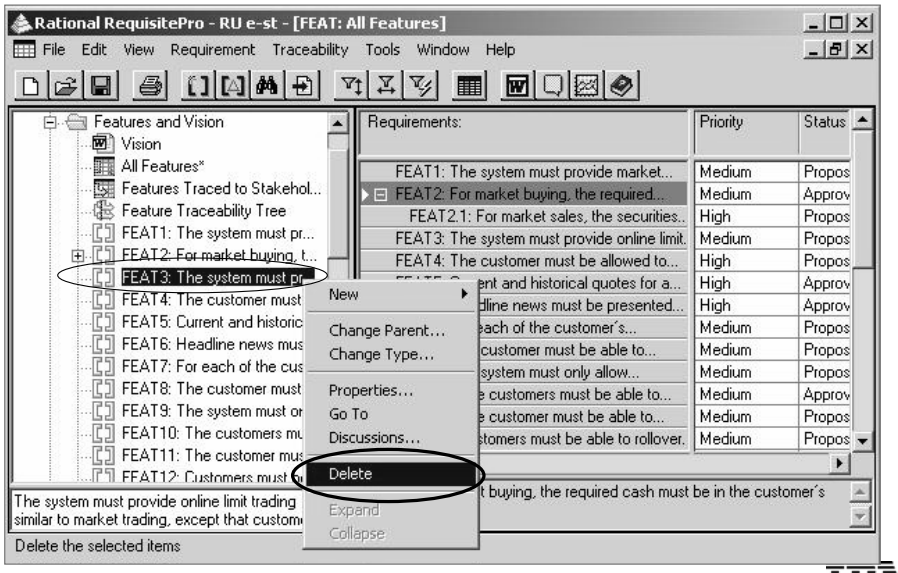
The CSV file and RequisitePro must be configured properly in order for you to import all information. Refer to RequisitePro Help for data formatting details.

To import a CSV file, click **File > Import** on the RequisitePro menu bar.

Note: The requirements are imported under the package you have selected in the Explorer.

Delete a requirement in the Explorer

- Right-click the requirement, and then click **Delete**.



11

IBM

Deleting a Requirement from the Database.

From the Explorer, right-click the requirement and select **Delete**.

A requirement that resides only in the database can be deleted from the Attribute Matrix in a view.

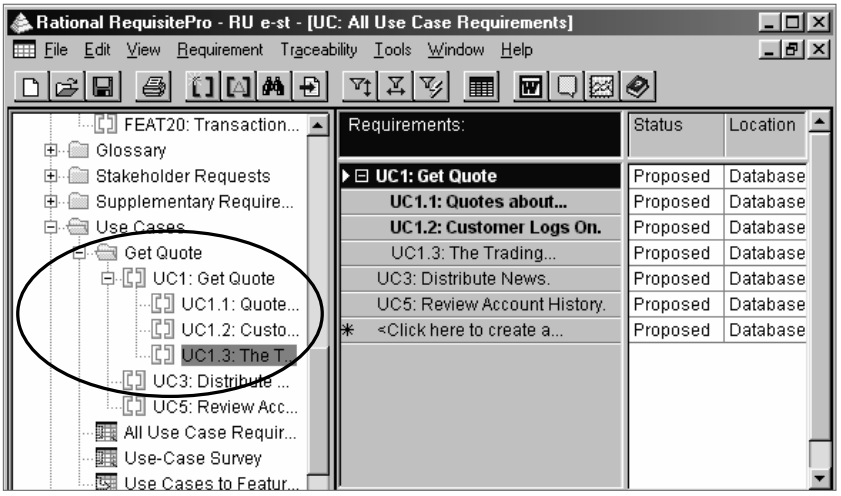
When a requirement is deleted, it is removed from the database, and its attributes, traceability relationships, and revision history are also deleted. There is no way to reverse a deletion.

Because there is no way to reverse a deletion, Rational recommends that you do not delete a requirement. Instead, assign the status attribute a value of Deleted. That way, if you ever want to reinstate it as a valid requirement, you still have its revision history and attributes in the database.

If the requirement is located in a Word document, you cannot delete it from the Explorer.

Organize requirements with hierarchy

- Subdivide a general requirement into more specific requirements.



12

IBM

Hierarchical requirements subdivide a general requirement into more specific requirements. In the example above, the requirement with the tag UC1 is a parent requirement. Its child requirements are numbered below it.

A leaf tag is found only in a hierarchical requirement and is defined as the digit(s) to the right of the final decimal point.

Hierarchical requirements can also be thought of as parent/child relationships. The hierarchical format follows a “top down” outline format. An outline contains headings and subpoints. All subpoints under the heading are assumed to be related to the heading. Because all children are subpoints of the parent, there is an implied relationship between the parent and all its children in the documentation.

Child requirements provide additional detail for their parent requirement.

It is important to understand a hierarchy in respect to cut/copy and paste. Hierarchical requirements can only be manipulated in this manner via the entire hierarchy. A child cannot be moved without its parent.

Rules for hierarchical relationships

- Parent and child: located together
  - In a document OR only in database
- Parent and child: same requirement type
  - Same root tag
  - Same attributes defined
- Parent (root) requirement may have
  - Up to 24 levels of children
  - Unlimited children at any given level

13

RequisitePro enforces certain rules when administering hierarchical requirements.

- Each hierarchy is considered to be a single unit by RequisitePro. The parent and child requirements must all be of the same requirement type.
- Hierarchies cannot exist separately from each other in RequisitePro. All cut/copy/paste operations must be performed on the entire hierarchy.
- Hierarchical requirements cannot span documents, nor can they span documents and database.
- A hierarchy is a one-to-many relationship. A child requirement can have only one parent, but a parent requirement can have an unlimited number of children. A hierarchy can contain up to 24 levels of children.
- When you delete a parent, you can choose to delete its children or assign them to another parent.
- If a parent requirement is changed, the relationships with its children becomes suspect. (Suspect links are discussed in Module 4.)

Lab 3: Gather and Enter Requirements

- Import requirement data.
  - From a CSV file located on CD-ROM.
- Create requirements directly in a view.
  - Assign attribute values.
- Organize related requirements.
- Define hierarchy.

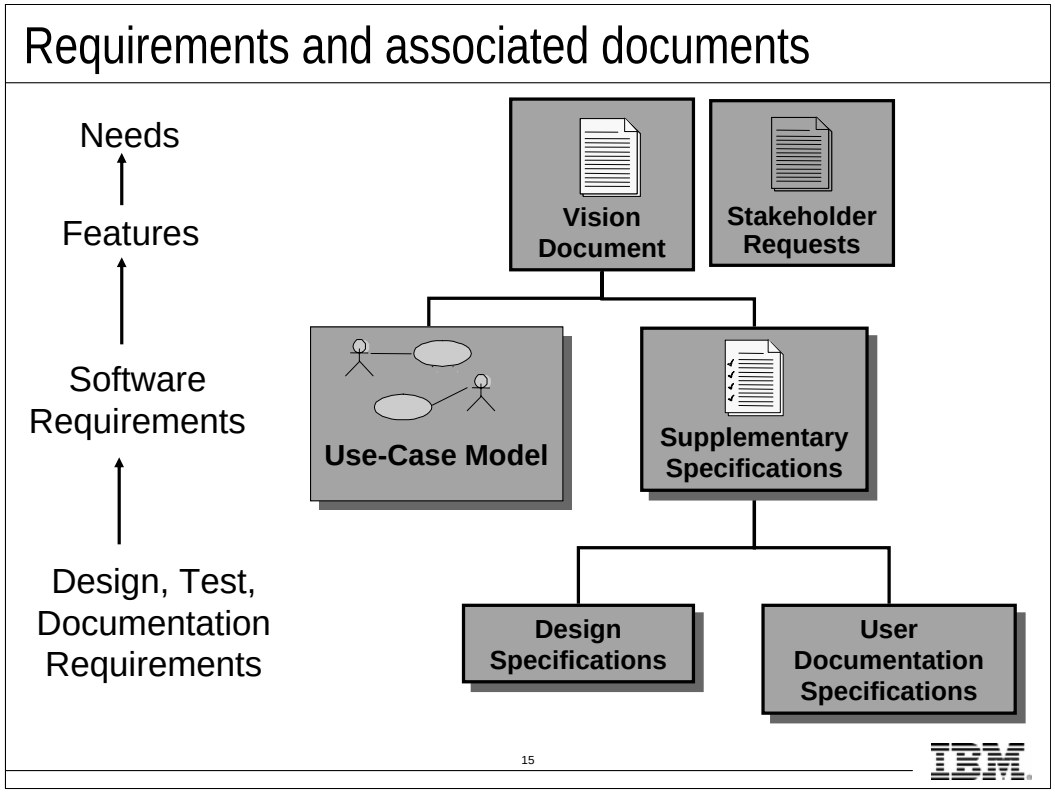
This lab focuses on entering and creating requirements directly in the project database.

14

IBM

See Student Workbook Lab 3.

Goal: Begin populating your project with requirements located only in the database. Import requirements and attributes from a CSV file. Also, enter requirements directly in the project database through a view and create a hierarchical relationship.



This diagram offers a sample documentation structure with a use-case development approach. Here you can see the documents recommended in a use-case approach and the type of requirements that are associated with each document type.

Notice the Needs and Features are typically captured in the Vision document, while the Software Requirements are documented in the Use-Case Model and the Supplementary Specifications.

Test specifications, design specifications, and user documentation specifications can also be captured in RequisitePro documents.

Use Word to write your requirements

- Create RequisitePro documents.
  - Create and edit requirements in RequisitePro documents using Word.
  - Delete requirements in documents.
  - Assign attribute values.
- Import existing documents into the project.

16

The most powerful feature of RequisitePro is the ability to maintain your requirements using Microsoft Word. RequisitePro integrates with Word to add the requirements management capability. By using Word you have all the features of a flexible word processor at your fingertips, as well the power of a database to manage your requirements.

A requirements document in RequisitePro

- Is a Word document.
  - Associated with a Document Type.
- Has a default Requirement Type.
- May be based on an outline:

Vision.VIS.
- Is based on an outline.
- Has a default Requirement Type: FEAT.

Execute Trade.UCS.
- Is not based on a outline.
- Has a default Requirement Type UC.

17

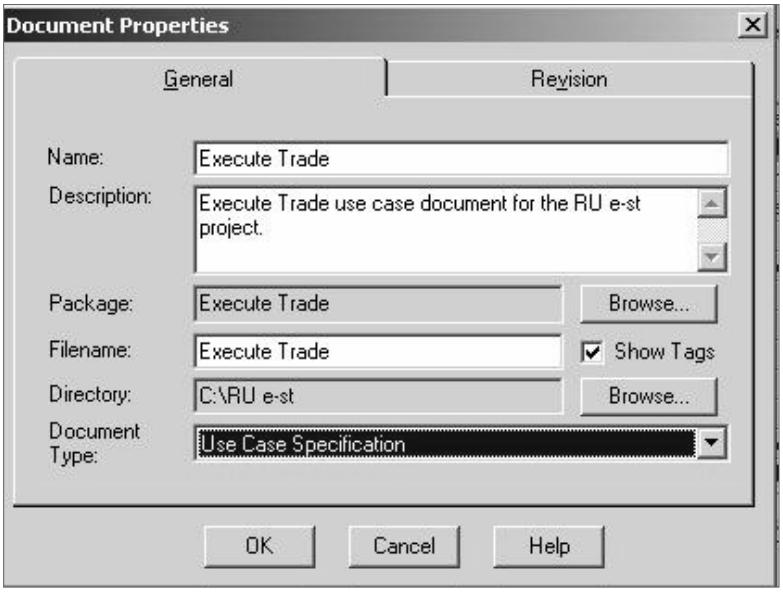
Now it is time to explore the use of requirements documents in a RequisitePro project.

Each document is associated to a Document Type. Documents created in RequisitePro are assigned file name extensions that indicate the document type. If you assign a different file name extension, you make it impossible to open the document outside of RequisitePro.

An outline is a template. This means that when you create a new document, it will be based on a template that you specify.

Requirements located in a RequisitePro document are maintained in the project database. All requirements can be viewed from a view, regardless of where they are located.

Create a requirements document in the project



18

IBM

A requirements document is created within RequisitePro, so its requirements can be maintained and managed in the database.

When creating a document, RequisitePro adds a description of the new document to the project database. The document is stored in a file. The requirements created in the document are stored in the project database and in the document.

Revision History

It is worth noting that revision history is kept at both the document level and the project level.

Revision may be kept at different levels (project, document, or requirement.) A revision is identified by a unique internal revision number, generated by Rational RequisitePro.

A document can be created in RequisitePro or Word. To create a document:

In RequisitePro:

Select a package, and then click **File > New > Document**.

Or

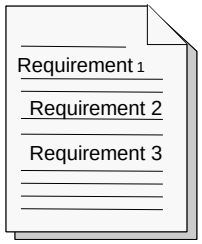
Right-click the package, and then click **New > Document**.

In Word:

Click **RequisitePro > Document > New**.

Create requirements in a document

- Enter directly into the document.
- Create from existing text.
- Import from external files:
  - Word document
  - RequisitePro documents
- Cut/Paste from documents or database.



19

IBM

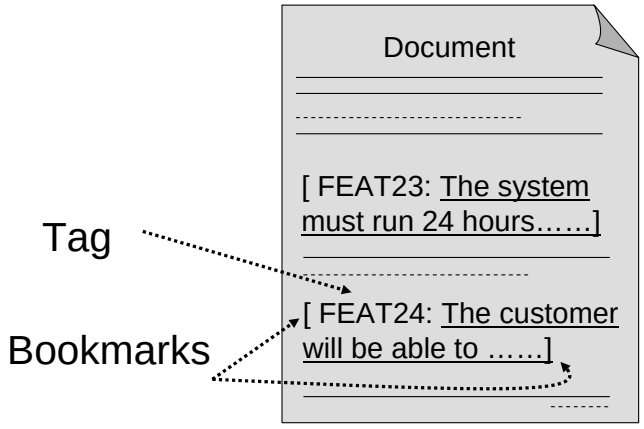
Once the document is created, begin to fill in the text and requirements within the document.

The following describes ways to create requirements in a document.

- Type them directly into the document within RequisitePro.
- Select some non-requirement text in the document and change it into a requirement.
- Cut a requirement from another location and paste it into the document at the desired location. Two cautions:
  - For requirements, use cut/paste on the RequisitePro toolbar instead of Word's cut and paste commands.
  - Paste soon after you cut a requirement. Requirements that are cut using RequisitePro's cut command are placed into a buffer only available to RequisitePro. If a requirement is in the buffer and another requirement is cut or copied, the original requirement is deleted from the database.
- Import requirements from existing Word documents and documents in other RequisitePro projects. The import feature increases reusability of previous requirement documents.

Requirements located in a document

- Maintained in the database.
- Edited in the document or a view.
- Updated when document is saved.



20

A requirement created in a document is maintained in the database. A system-generated attribute titled Location shows the name of the document in which the requirement resides.

As mentioned in Module 2, tags are RequisitePro's indication of a requirement. Because they contain hidden text, it is recommended that you do not manipulate the hidden text in the document or in the tag.

Requirements are denoted in documents by Microsoft Word bookmarks. These are defined by RequisitePro. It is recommended that you avoid the manipulation of bookmarks through Word settings and that you avoid the use of other bookmarks in your RequisitePro document.

Requirements located in a document may be edited in the Views workplace if the **Enable Extended Editing of Documents** check box is selected in the **Options** dialog box.

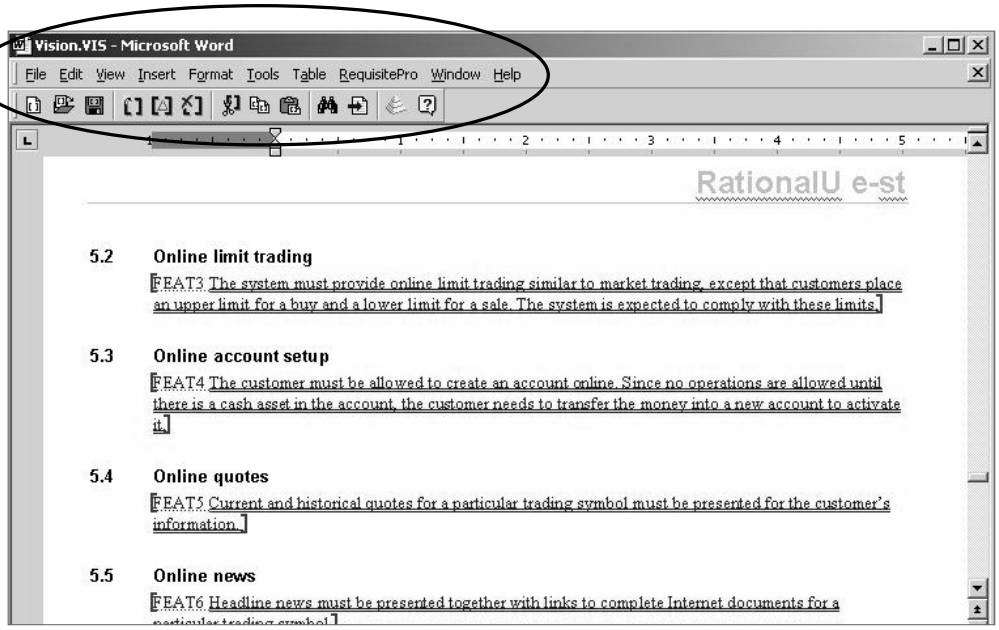
Word-Linked Files

You may include Word-linked files as part of requirement text in a document. You can link to Microsoft PowerPoint files, Excel spreadsheets, bitmap files, and other types of files. However, do not link to other RequisitePro documents.

Object Limits

A requirement can contain an object, such as a picture or spreadsheet. An object cannot be stored in the project database, so it can only be part of the requirement if that requirement is located in a document.

Maintain requirements in documents



21

IBM

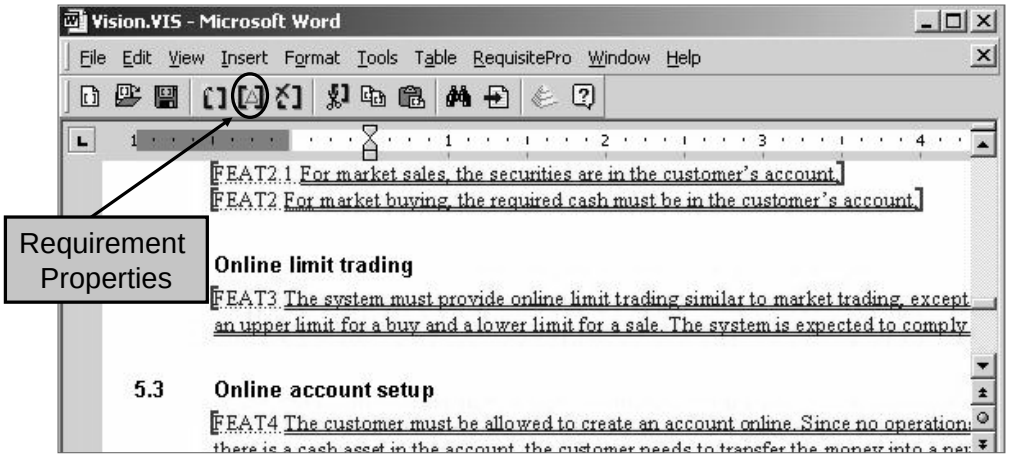
Now it is time to explore the use of requirements documents in a RequisitePro project. RequisitePro documents allow the user to add descriptive text to the requirements, print these documents and share them with Stakeholders, Project Managers, and team members.

The dynamic linking of the requirements in a document to a database keeps both environments up-to-date with the most current information. From the document, a simple right-click enables you to view or modify any of that particular requirement's attributes or traceability.

If you choose not to write documents to capture your requirements, you can maintain your requirements directly in the database via RequisitePro.

Edit a requirement in a document

- On the RequisitePro tool bar.



22

IBM

You can edit a requirement in a document simply by entering the text between the bookmarks.

You are also able to edit a requirement located in a document from RequisitePro IF you have the option enabled. The option is enabled in the **Project Properties** dialog box.

You can update requirement text without having to open the document. This allows multiple people to simultaneously edit requirements within a single document. All changes are tracked in the database.

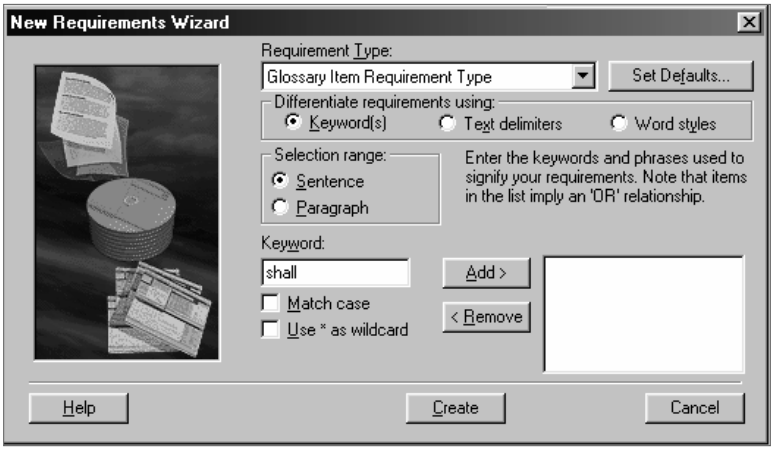
You can turn this option on and off at any time.

A requirement that has been updated in the project database will not be changed in the document until you open that document. If the document is open, the update is immediate. When you open the document after changes have been made in the database, you will receive notification of text updates.

You can modify a requirement's properties by placing the pointer somewhere between the requirement's bookmarks and clicking the **Requirement Properties** button.

Import a document with the Import Wizard

- Create multiple requirements from selected document text.
- Support wildcard (*) characters.



23

IBM

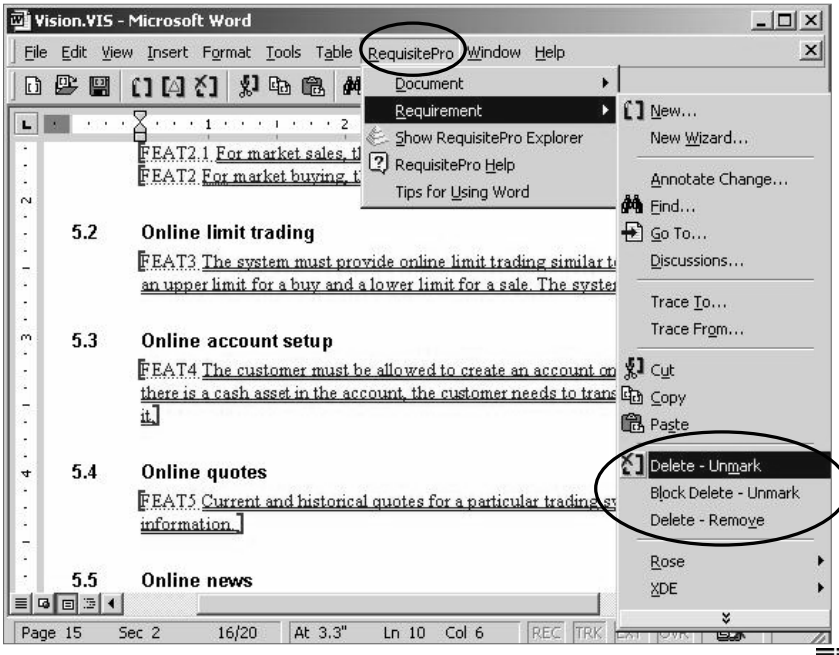
From Word, use the **Requirement > New Wizard** command to create multiple requirements from a block of selected text in a document. When you select this command, the Import wizard guides you through creating the requirements step-by-step. Specify the keywords to use for the requirements and indicate whether the requirements are in sentences or a paragraph.

RequisitePro uses three methods for distinguishing requirements in a highlighted block of text:

- Keywords such as: “must”, “shall”, “will be able”
- Text Delimiters:<>, {}, []
- Word Style: Heading 1 or Body Text

Note: When using wildcard characters, you cannot also match case. Word does not have this function/option.

Delete a requirement in a document



24

When a requirement in a document is deleted, it is removed from the database, and its attributes, traceability relationships, and revision history are also deleted. There is no way to reverse a deletion.

There are two kinds of deletion methods for requirements that reside in a document. Both ways result in the requirement no longer being a requirement.

Delete - Unmark allows the text of the requirement to remain in the document. It becomes ordinary text and no longer a requirement.

Use **Delete - Unmark** or **Block Delete - Unmark** to delete requirements from the document where they reside and from the project database without disrupting the text of the document. **Block Delete - Unmark** is the same as **Delete - Unmark**, but it deletes all requirements in a selected section of a document.

Delete - Remove removes the entire text of the requirement from the document.

The commands to delete a requirement in the document are located on the RequisitePro toolbar.

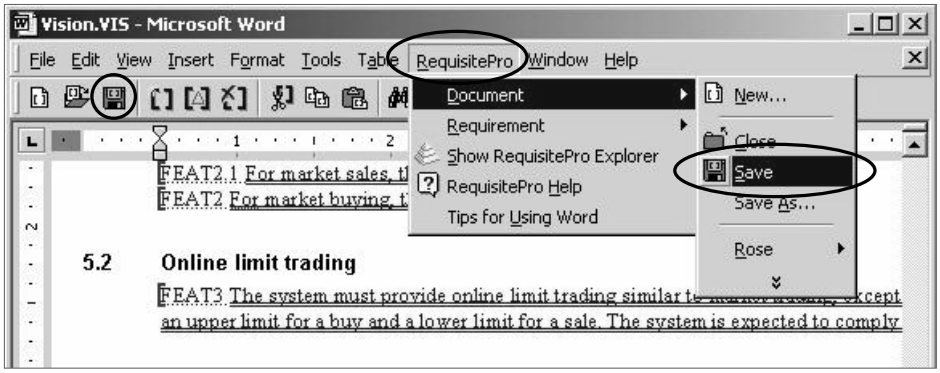
Alternative Deletion Strategies

- Assign the Status attribute value Deleted.
- Place Deleted requirements in a document.

Important: Delete requirements only when necessary, since all the requirement information (history and traceability) is removed when you perform a deletion.

Save a RequisitePro document

- From the RequisitePro tool bar.
- From the RequisitePro menu.
  - Document > Save.



25

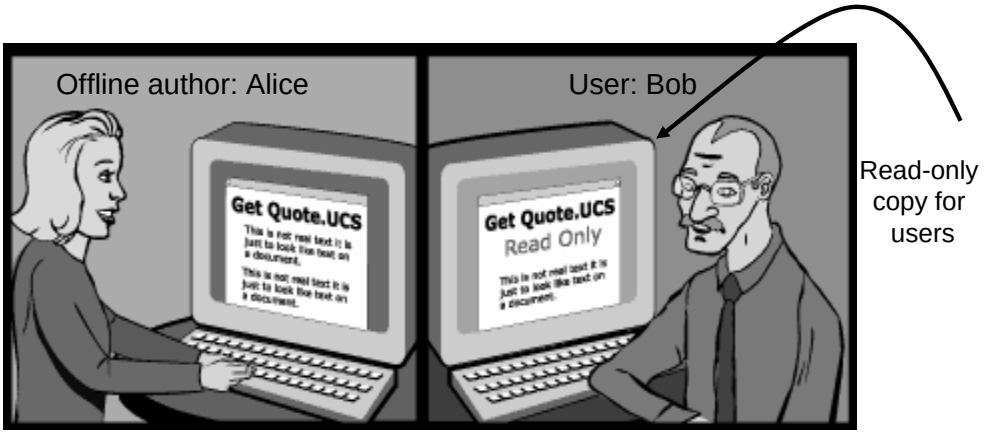
IBM

The **Save** command on the RequisitePro toolbar saves the active requirements document. This command is also available by clicking **RequisitePro > Document** in the Word workplace.

Note: Do not save a document in Microsoft Word by clicking **File > Save**. If you do so, the document will be saved, but requirements information will not be updated in the RequisitePro database until you close the document or click the RequisitePro **Save** command on the toolbar or menu.

Offline authoring

- Provides support for distributed teams.
- Suits a world of portable computers.



26

IBM

Offline authoring makes it possible for portable computer users to keep an individual document on their computers instead of the whole project. While the user is traveling, the document can be taken offline and edited without tying up the entire project. Other team members can work in the project while the document is offline.

RequisitePro places no limit on the number of documents that can be offline at one time. This is a great benefit to distributed teams, because access to the RequisitePro database is not necessary to capture requirements in documents and modify existing requirement documents.

This feature is also helpful in the review process when customers or consultants need to view the document but do not have a copy of RequisitePro.

When the document is taken offline, the author is prompted by RequisitePro for information regarding it. This information enables team members to determine at any time who has a document offline and why.

There is a clear communication channel among the team members. Access to current data is critical. At the same time, you want to be sure that changes are tracked and integrated into the database while maintaining a level of version control. Offline authoring helps with these issues.

Use offline authoring in addition to the Merge and Revise function. At review time, the document is taken offline and distributed to team members. Reviewers turn on the Track Changes feature in Microsoft Word, edit the document, and return it to the team member who took the document offline. The reviewer merges all changes into the offline document and returns it to RequisitePro, where the database is updated.

An offline authoring scenario

- Take a document offline to update.
  - Leaves a read-only copy behind for people to view.
  - Take document home to work on it.
- Work on the document at home.
  - Macros in the document allow you to add, modify, or delete requirements.
- Bring the document back online.
  - Replaces the read-only copy.
  - Incorporates edits and updates database.

27

Offline Authoring provides the ability for document authors to take a document out of the RequisitePro project, edit it in Microsoft Word outside of RequisitePro, and return the document to the project.

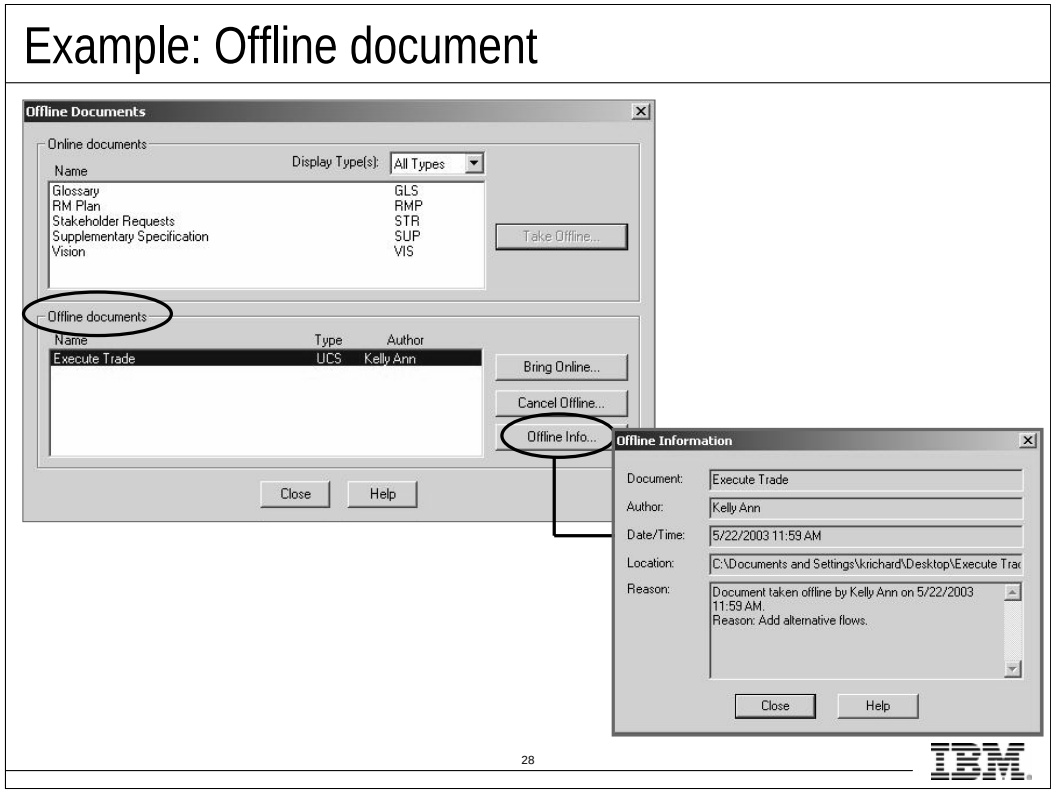
When you take a document offline, RequisitePro includes Microsoft Word macros in the document. These macros allow you to add and delete requirements in the offline document.

While the document is offline, the author can edit the textual content of the document, mark new requirements, and delete (unmark) existing requirements. A read-only copy of the document is left in the project for review by other team members during the offline process.

Upon return to the project, the offline changes are detected by RequisitePro and added to the project database.

Team members must have security privileges of Read/Write to manipulate a document or take it offline.

All changes are incorporated unless the document author cancels the offline operation. In this case, the original “read-only” copy is restored to the online project.



When you take a document offline, you can add and delete requirements in the document and edit document text. No one else can make changes to that document while it is offline.

Since only one person can have a RequisitePro document offline, offline authoring provides a level of control. The original is still stored in RequisitePro, but it is changed to a read-only document. Other users can view the document in RequisitePro, but they cannot edit it until the the author brings the offline copy back online.

Reminder:

The **Save As** command differs from offline authoring, because **Save As** saves a copy of a RequisitePro document as a .doc file. The Word document can be distributed to non-RequisitePro users to review the documents. Documents created with the **Save As** command contain information that was current at the time that the document was saved.

To take a document offline, do one of the following:

In RequisitePro:

Click **Tools > Offline Documents**.

In the Word workplace:

Click **RequisitePro > Documents > Offline Documents**.

Lab 4: RequisitePro Documents

- Create a RequisitePro document.
- Import documents and requirements into the project.
- Create and edit requirements in a RequisitePro document.
- Take a RequisitePro document offline to edit.

This lab focuses on creating and modifying requirements in documents, importing documents, and creating documents in the project.

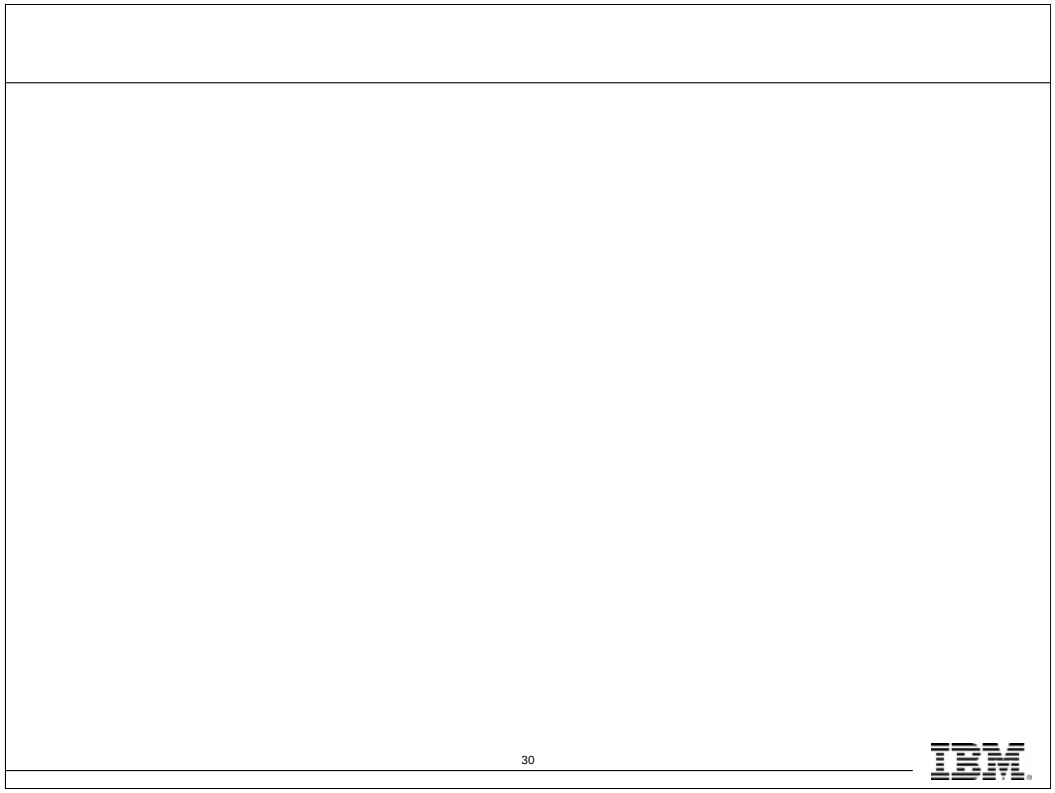
29

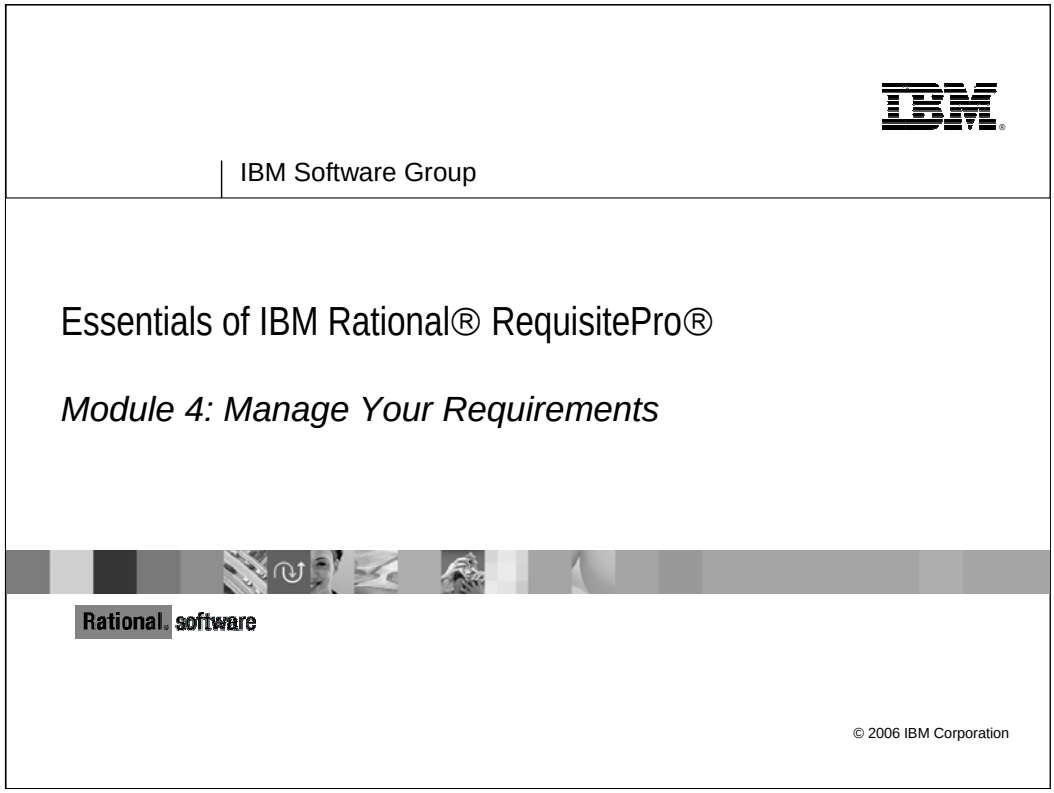
IBM

See Student Workbook Lab 4.

Goal: Import and create documents into the RU e-st project.

- Import a Word document (Vision.doc) and create requirements using the Import Wizard.
- Import a RequisitePro document (Supplementary_Specification.SUPL) that contains requirements already defined.
- Import a Word document (Execute_Trade.doc) and manually create requirements from text in the document.
- Create a document from an outline (Stakeholder_Requests.doc), insert text, and create requirements.
- Take a document offline. Edit the document offline and bring the document back into RequisitePro.





Contents

Objectives: manage your requirements	4-2
Requirements traceability	4-3
Suspect links	4-10
Export a view to Word or CSV format	4-14
Requirement queries	4-15
Define filter and sort criteria for query	4-17
Lab 5: Traceability and Requirement Queries	4-18
Manage requirements using metric reports	4-19
Basic filters for query	4-23
Time-sensitive filters	4-25
Lab 6: Metric Reports	4-27
RequisitePro baseline	4-28
RequisitePro Baseline Manager	4-29
Comparing baselines	4-33
Generating a baseline comparison report	4-36
Lab 7: Creating and Comparing Baselines	4-38

Objectives: Manage your requirements

- Define traceability relationships.
- Query requirements data.
- Create requirement statistics (metrics).
- Recognize the functionality of the IBM® Rational® RequisitePro® Baseline Manager
- Create and compare baselines using the RequisitePro Baseline Manager.

2

This module shows you how Rational RequisitePro can help you manage your requirements. It concentrates on defining traceability and using traceability and attributes for managing queries.

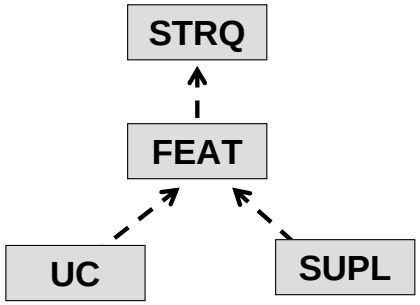
When you create traceability relationships between requirements, Rational RequisitePro visually indicates a dependency between two requirements. With this information you can perform real-time impact analysis and make informed decisions for project scope management and resource allocation, for example.

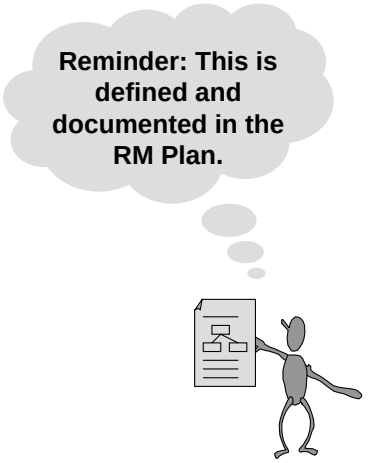
Rational RequisitePro helps you understand the impact of change with powerful traceability features that let you link related requirements. As change occurs, you are immediately aware of its impact within a project.

The RequisitePro Baseline Manager enables you to create and compare baselines of requirements contained in documents, Attribute Matrix views, and packages, or an entire RequisitePro project. Baselines can help you manage changing requirements, mitigate risks, and manage project scope more effectively.

Requirements traceability

- A relationship between two requirements.





3

IBM

Traceability is a methodical approach to manage change by linking requirements that are related to each other.

Traceability provides a link between two associated requirements.

An example of a traceability strategy:

- Product features are directly traced to stakeholder requests.
- Use cases and supplementary requirements are traced to features.
- Use cases are indirectly traced to stakeholder requests because they trace to features, which in turn trace to stakeholder requests.

Traceability benefits for requirements management include:

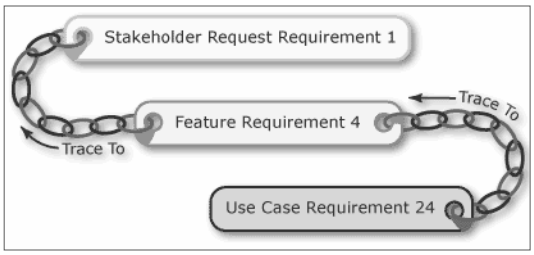
- It is an efficient method of estimating the impact of a change to a requirement, because the requirements related to the changed requirement can be easily found.
- It is used to show that one requirement is derived from another or that one requirement is dependent on another.
- It helps ensure that customer needs are reflected in the final product, because the links can be followed from a customer need to the related features, to the related software requirements, to the design objects, and so on.

Rational recommends that you use only one direction of traceability and that you start with basic necessary traceability. Traceability is powerful and helpful, but it also takes a lot of work to maintain.

Rational RequisitePro prevents you from making circular relationships. For example, you cannot trace a use case to a feature to a stakeholder request, and the stakeholder request back to the use case.

Traceability relationships

- Link two requirements to each other.
- Help manage change.
- Display in views:
  - Traceability Matrix
  - Traceability Tree



Views are created by Requirement Type.

4

IBM

Rational RequisitePro helps track changes in requirements throughout the development lifecycle. With traceability relationships, change is more visible because you can see that other requirements are related to a changed one.

There is only one traceability relationship between any two requirements. The difference between calling that relationship a "trace to" or "trace from" one is a matter of perspective.

For example, if FEAT2 is traced to UC6, both statements below are true:

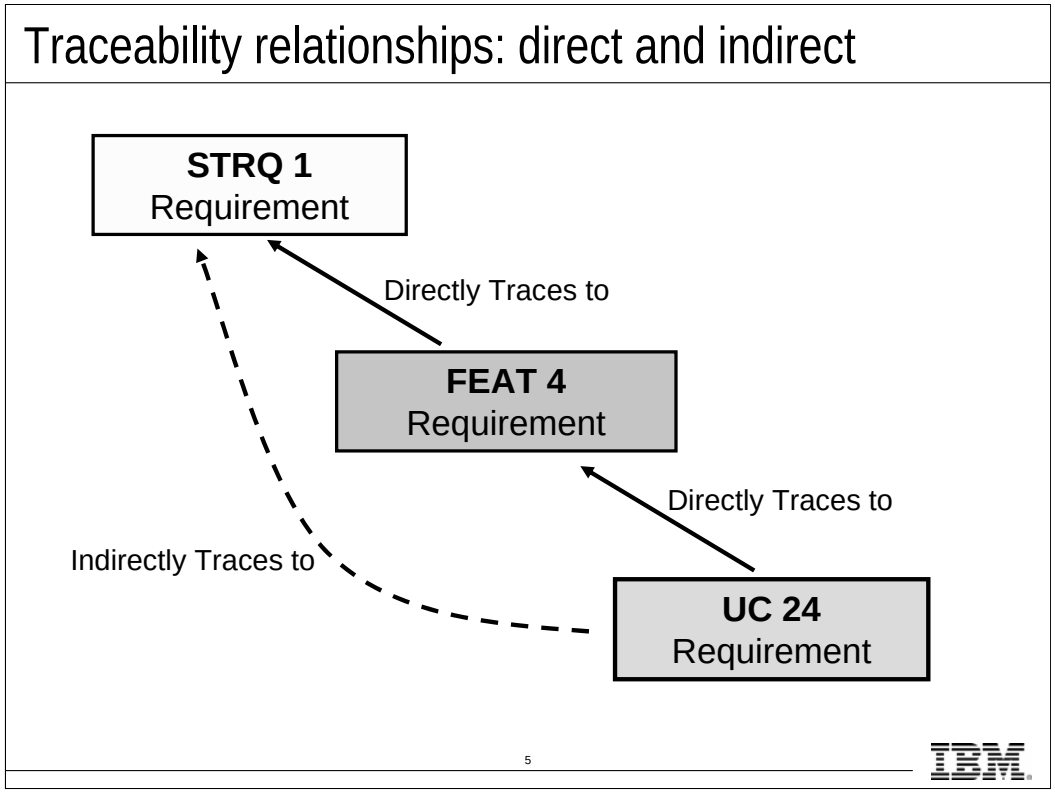
- Requirement FEAT2 is traced to UC6.
- Requirement UC6 is traced from Requirement FEAT2.

Rational recommends you use only one direction of traceability. For example, if you decide to trace *from* use cases *to* features, then consistently set all your links from a use case to a feature.

Traceability is displayed in the Views workplace in either a matrix or a tree view. The Traceability Matrix displays traceability relationships between two indicated Requirement Types. The two traceability trees (Traced out of…. Traced in to…) display the entire Traceability Tree relating to the specified root Requirement Type.

Traceability Views tip:

- "Trace to" Row to Column: The requirements appearing in the rows trace to the requirements appearing in the columns.
- "Trace from" Column to Row: The requirements appearing in the columns are traced from the Requirement Type appearing in the rows.

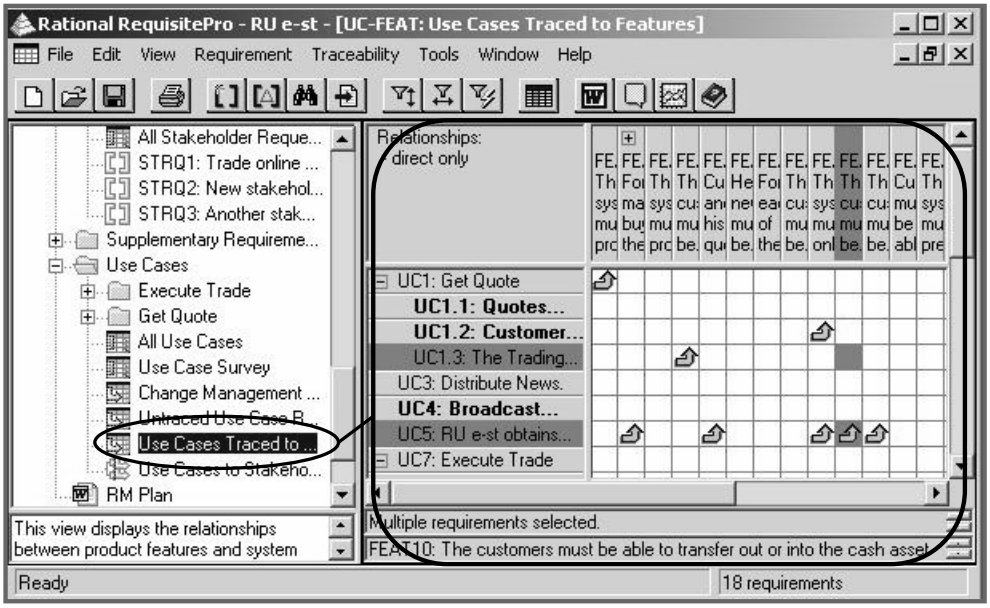


A traceability relationship is indirect if a requirement traces to an intermediate requirement (FEAT 4 Requirement), which in turn traces to a third requirement. For example, UC24 Requirement is indirectly traced to STRQ1 Requirement. The relationship between UC24 and STRQ1 is indirect.

Indirect traceability relationships are maintained by Rational RequisitePro; you cannot modify them. An arrow outlined by a dotted line, lighter in color than a direct relationship arrow, indicates an indirect traceability relationship in the Traceability Matrix and Traceability Tree views.

A Traceability Matrix always displays the direct traceability relationships. If you want indirect relationships also to be shown on a Traceability Matrix, click **View > Properties > Show Indirect**.

Traceability Matrix



6

IBM

This view helps you see the relationships between requirements and understand how a change affects other requirements and the overall project. With this information, you can make decisions about how to best manage change.

The Traceability Matrix displays the relationships between two types of requirements. The arrow indicates a relationship between two requirements. This matrix can be used to create, modify, and delete traceability relationships.

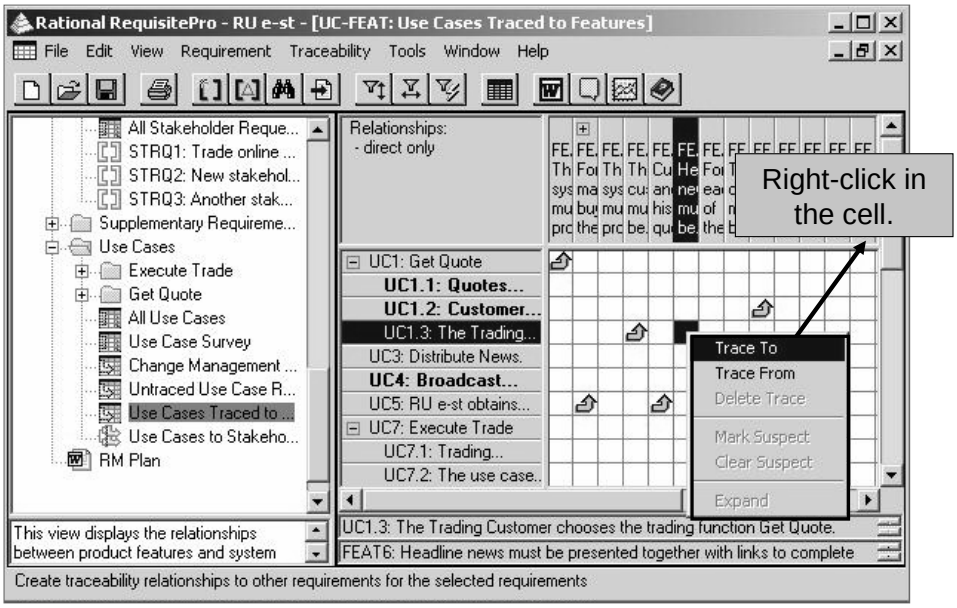
Direct traceability relationships are ones that are specified between two particular requirements. For example, in the matrix shown above, Requirement UC1.2 is directly linked to Requirement FEAT9.

How will you use this information to manage the scope of the system?

The project requirements information that has been gathered and entered into the RU e-st project allows you to:

- See the relationships between requirements, which can determine how change impacts related requirements.
- Prioritize requirements to determine which requirements should be implemented for each iteration of development.
- Create queries based on requirement attributes to review project status and progress.
- Assure quality and verify that all implementation fulfills defined requirements.

Set traceability links



You can easily set Traceability links within a Traceability Matrix by right-clicking the row and column you want to link.

Indirect links are created by Rational RequisitePro and are visible in the matrix if you have selected **Show Indirect** in the view's **Properties** dialog box.

To create a trace link:

From the Views workplace:

Right-click a cell, and then click **Trace To** or **Trace From**.

From the Word workplace:

Click **RequisitePro > Requirement > Trace To**.

From a requirement's **Properties** dialog box:

Click the **Traceability** tab, and select **Add** in the **To** or **From** list box. Next select the requirement to trace to or from and click **OK**.

To delete traceability links:

Right-click the link in the view, and then select **Delete Trace**.

Traceability links: Trace To or Trace From

- Represent a bidirectional dependency relationship between two requirements.
  - **Trace from** relationships and **trace to** relationships help you understand how changing requirements can impact related requirements.
- Be consistent.
  - Use only one direction between two Requirement Types.

8

IBM

When you create traceability relationships in Rational RequisitePro, you can "trace from" one requirement to another, or "trace to" one requirement from another. The difference between calling the "trace to" or "trace from" relationship is one of perspective.
The terms "trace to/trace from relationship," traceability relationship," and "link" and are used interchangeably.
When using traceability in Rational RequisitePro, the links are automatically maintained, making it is easy to assess their impact of change. Traceability helps mitigate risk and manage the scope of the system.

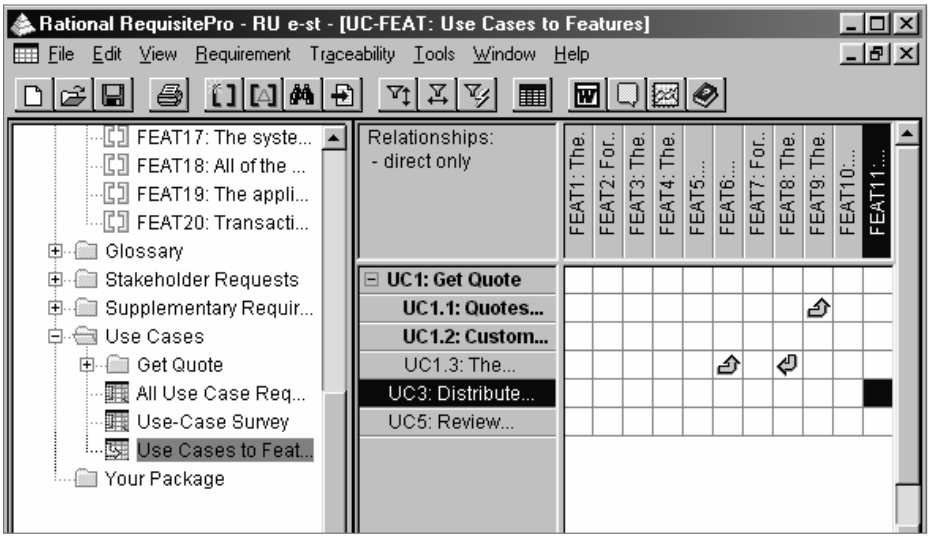
For example, both of these statements are true:

UC1 is traced to FEAT1; FEAT1 is traced from UC1.
STRQ2 is traced to FEAT2; FEAT2 is traced from STRQ2.

Trace from relationships and Trace to relationships help you to understand how changing requirements can impact related requirements.

Important reminder:
The traceability structure (directional flow and dependencies) should be decided before you start the project, defined in the RM Plan, and must be consistently used throughout the project lifecycle.

Example of inadequate traceability

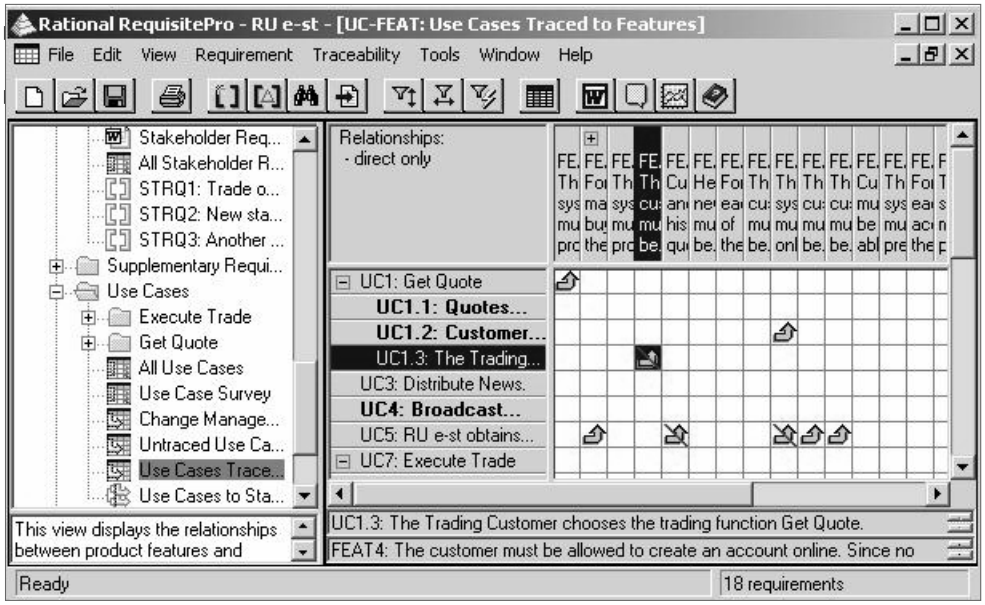


What's wrong with this picture?

9

All of your traced relationships in the matrix should be in the same direction (have the same flow, bottom-up or top-down). If they are not in the same direction, then querying the database to obtain useful information becomes very difficult – if not impossible. The traced relationships should be based on the project's RM Plan.

Suspect links



10

One of the exciting features of Rational RequisitePro is the ability to track suspect traceability links. When a change is made to a requirement, Rational RequisitePro automatically marks its relationships as suspect.

A visual indicator (a red diagonal line) is placed on all traceability links directly connected to the changed requirement. In addition, a link is marked as suspect if a change is made to an attribute value of a requirement (if that attribute has been defined to “affect suspect”).

A suspect link indicates a change to a requirement. Team members must review each such change.

Traceability views can be queried to display only those requirements that have links that are suspect. This is an easy way to find the requirements that need to be reviewed.

You may manually mark a link as suspect even if neither requirement has changed. You might do this to remind yourself that some aspect of the linked requirements must be reviewed.

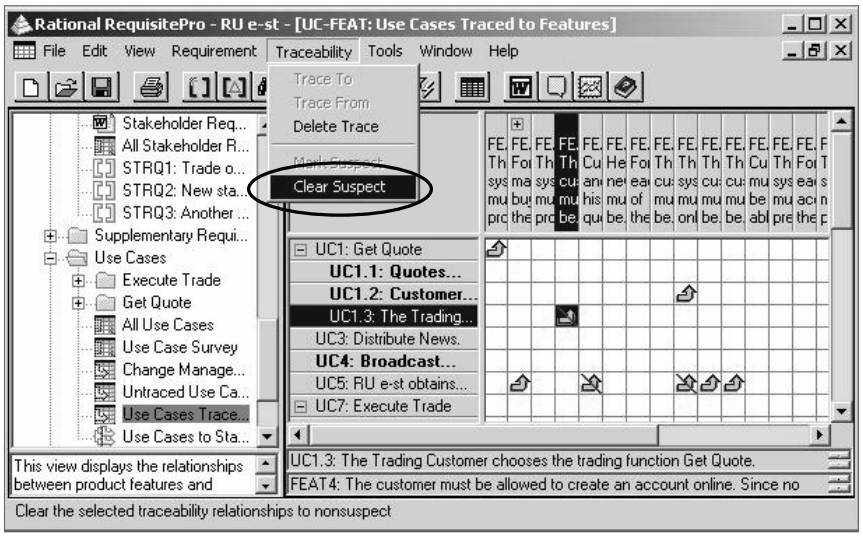
Why do requirements change?

- The team did not ask the right people the right questions.
- There was a change in the business problem being solved.
- The users changed their minds or their perceptions.
- The external environment changed.
- The team failed to create a process to help manage change.
- The team’s understanding of the problem improved.

Rational RequisitePro marks related requirement links as suspect. When a link becomes suspect, the dependent requirements must be checked to determine if or how they are affected by the change.

Suspect links (continued)

- Must be manually cleared.



11

IBM

Since a suspect link indicates a change to a requirement, team members need to review that change.

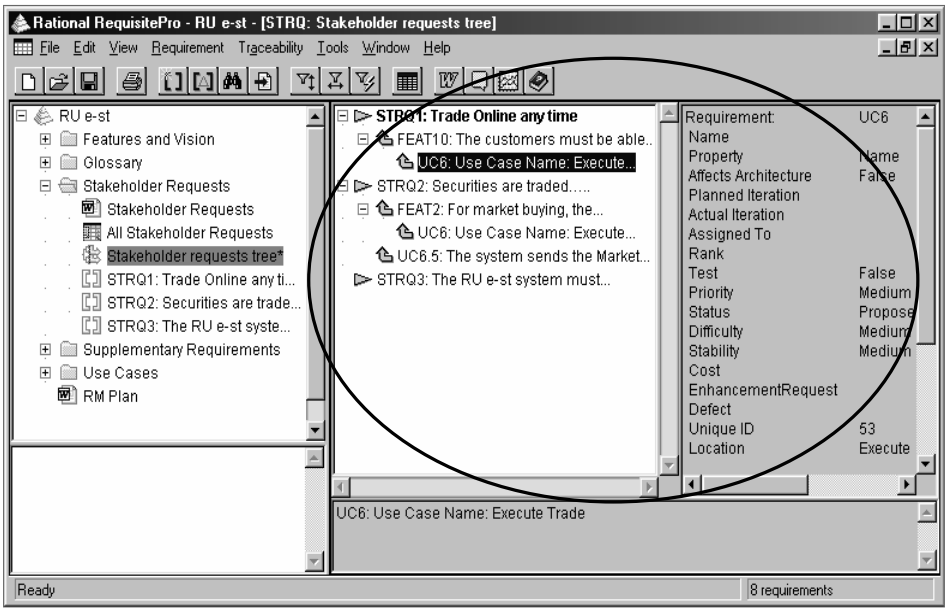
A team member determines if any updates must be made to the linked requirements. After the review of requirements in question, the suspect link must be "cleared" (erased) manually by a user.

You may manually mark a link as suspect, even if nothing has changed, as a reminder to review the link or as a reminder to review the link to ensure that the dependency relationship is valid.

You can turn off the **Auto Suspect** option—for example, during a spell check. If you do this, the option remains off until you turn it back on. So, if you turn **Auto Suspect** off for spell check, do not forget to turn it on again when the spell check is complete.

On the RequisitePro menu bar, click **Tools > Auto Suspect**.

Traceability Tree view



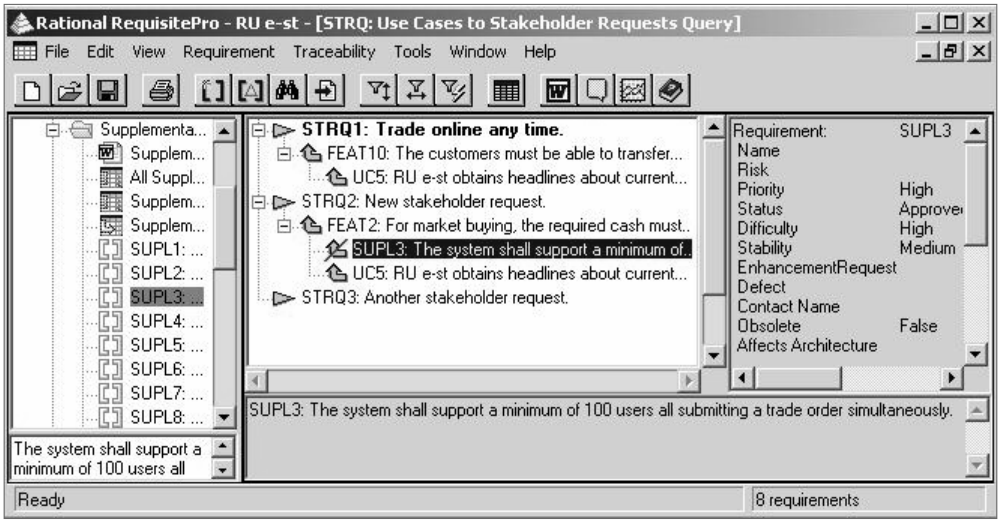
12

A Traceability Tree displays the entire chain of relationships based on a root requirement of one Requirement Type and all the traceability links and parent-child hierarchies associated with the root requirement. A root requirement is a requirement at the uppermost level of the requirements hierarchy.

A Traceability Tree is a view that displays all internal and external requirements traced to or from a requirement (depending on the direction of the tree).

Here is an example of a Traceability Tree with STRQ as the root requirement.

Suspect links with hierarchical relationships



13

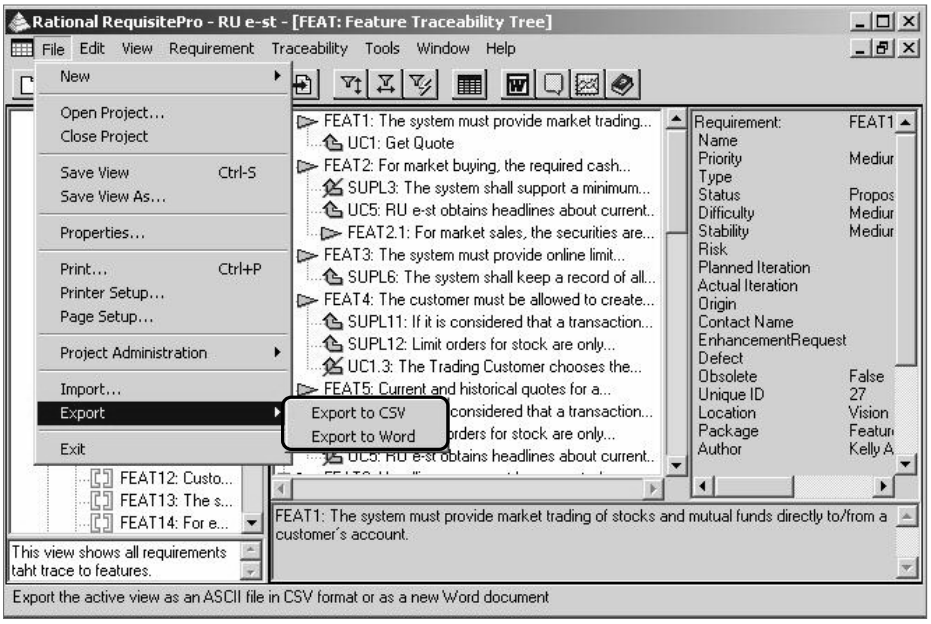
Rational RequisitePro marks suspect links between parent and child requirements. When you modify a parent requirement, Rational RequisitePro marks the relationship between the parent and all its immediate children as suspect. Changes include modifications to the requirement name, requirement text, Requirement Type, or attributes.

Suspect links on hierarchical requirements are usually viewed in Tree views. There is a special symbol for suspect links in a hierarchy: a triangle with a red line through it.

Hierarchical suspect links can also be seen in a Traceability Matrix from a Requirement Type to itself.

Hierarchical relationships are not the same as traceability relationships. Hierarchical relationships cannot be queried for suspect links. Queries are covered in the next section.

Export a view to Word or CSV format



14

Rational RequisitePro allows you to save the view results to an external file. This allows you to take the information away with you for analysis at a later time.

Views can be exported to non-RequisitePro users for review purposes. You can export a view to a CSV file or to Word format.

To export views:

Click **File > Export > Export to CSV**.

or

Click **File > Export > Export to Word**.

Requirement queries

- Query on attribute value or traceability links.
- May be saved and rerun at any time.

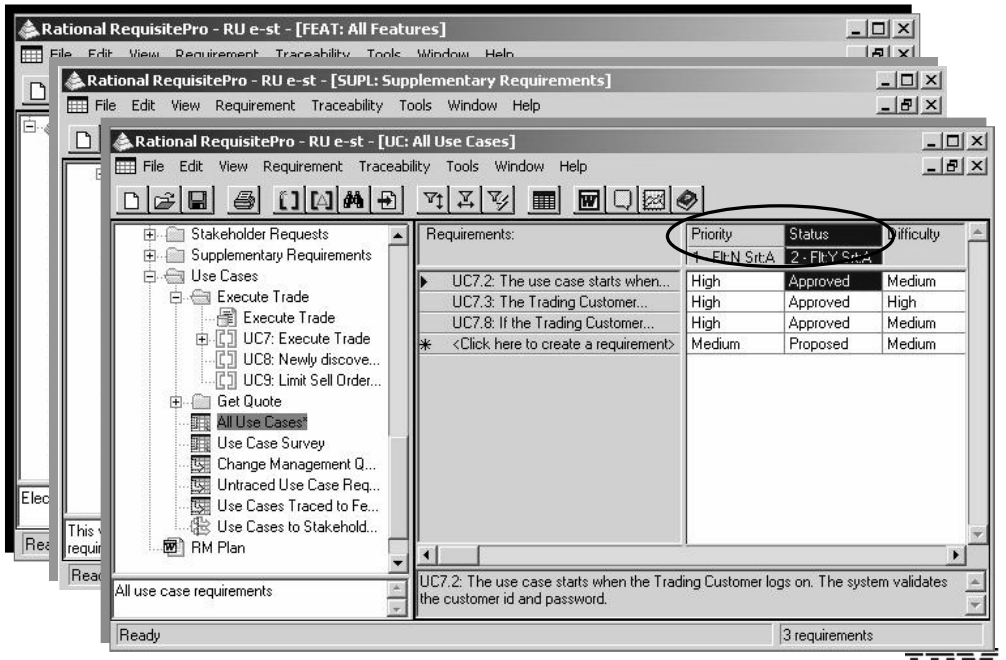
- Questions
  - Which features are assigned to this iteration?
  - Which features are high customer priority?
  - Which use cases are completed?
  - Which use cases have been changed?

15

In order to decide on the scope of the system to be built, a manager needs information about the status, cost, and difficulty of the requirements. Rational RequisitePro provides requirements management capabilities to obtain requirement information. For example, you can:

- Use attributes and traceability to manage project progress.
- Use queries to obtain answers to a variety of management questions.
- Query your requirements database to obtain answers to a wide variety of questions about the requirements.
- Save your queried views in RequisitePro as project-wide or personal views. You can rerun these saved views throughout the lifetime of the project.

Query examples



Filter requirements in a view by limiting the value of one or more attributes or by limiting traceability.

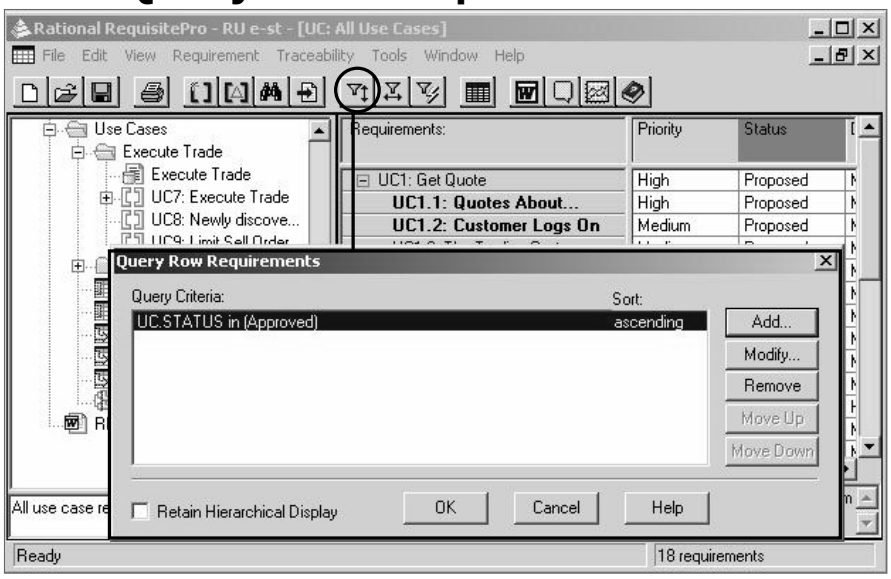
Filtering restricts the information being displayed.

Sorting determines the order in which information is displayed.

You can query row (Attribute and Traceability Matrices), column (Traceability Matrix), or root requirements.

Define filter and sort criteria for query

- Click **Query Row Requirements** button.



17

Determine the filtering and sorting criteria you need to apply to generate the desired view. After you click **Query Row** or **Query Column**, Rational RequisitePro displays the dialog boxes in which you can determine your sort criteria.

For example, John needs to review all of the use case requirements with Priority value of High and Status value of Approved. To do this, he will create the Attribute Matrix, click **Query Row**, and select the criteria in the dialog box.

Once John submits his criteria, Rational RequisitePro creates the query results. The results show him the number of filters used to create the query and whether they are in ascending or descending order.

The results of John's query are static; that is, they reflect the point in time in the project at which the query was run. All query results can be saved as a view. If any changes are made to the database after John saves the view, he must update the view to see the latest information.

To update the view, click **View > Refresh** on the RequisitePro menu bar or click the **Refresh the View** button on the RequisitePro toolbar.

Lab 5: Traceability and Queries

- Set traceability links.
- Create suspect links.
- Query requirements in a view.
- Export requirement views.

18

IBM

See Student Workbook Lab 5.

Goal: In this lab, you will create and modify traceability links, review suspect links, create traceability views, create and modify management queries, and export views.

Manage requirements using metric reports

- Scope management
  - Resources and budget
  - Time
  - Priorities
- Change management
  - Impact analysis
- Requirement relationships
- Project status
  - Priorities
  - Tested
  - Progress

19

IBM

The following queries are examples common to most projects. They can be helpful in managing requirements.

Scope Management (Attribute Matrix)

- Query all requirements that are High Priority.
- Add Low Difficulty to a query.
- Query the Status value of Incorporated or Approved.
- Query use cases in the current iteration.

Project Status (Attribute Matrix, Traceability Trees)

- Query all feature requirements that are Approved.
- Query all requirements of a type that do not have traceability links.

Requirement Relationships

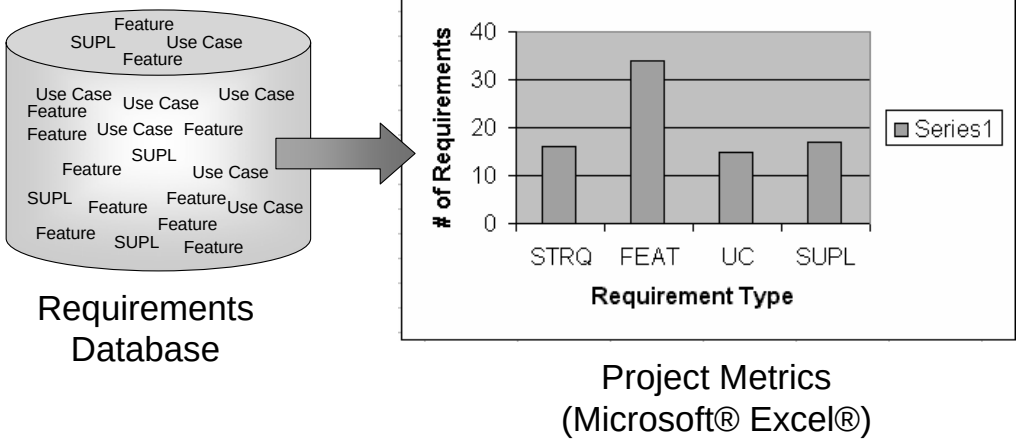
- Query use cases related to feature 1 (Traceability Matrix).
- Query all requirements of any type related to feature 1 (Traceability Tree).

Change Management (Traceability Matrix)

- Query traceability links that are marked suspect.

The results of your queries can be printed out and distributed to team members for work assignments.

Requirement metrics



Requirements Database

Project Metrics (Microsoft® Excel®)

20

Now let's discuss RequisitePro metrics. Rational RequisitePro metrics provide you with reporting capabilities on the project data.

Use requirement metrics to retrieve information that is vital for evaluating the progress of project priorities, workloads, and deadlines. The reports are displayed in Excel.

Requirement metrics provide project managers and product analysts with statistics concerning project requirement attributes, relationships, and revisions.

Requirement metrics reporting

- Provide statistical report capability.
- Apply filter(s) to requirement data.
  - Combine one or more filters to produce a query.
  - Apply one or more queries to produce a report.
  - Display reports in Microsoft Excel.

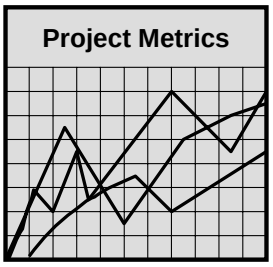
21

IBM

The metrics feature is used for compiling statistics on requirement name, text, attributes, relationships, and revisions. These report results are displayed in Microsoft Excel and can be manipulated using Excel's charting capabilities.

Metric report types

- Static report.
  - Uses basic filters.
  - Provides a snapshot view of project.
- Trend analysis report.
  - Uses time-sensitive filters.
  - Analyzes changes over time.



22

IBM

A requirement metrics filter, query, or report is considered static if it retrieves information that provides a “snapshot” view of a project at the present time.

A trend analysis report shows how the project changes over time. Each column represents one filter in a query, and each row tallies the number of revisions that meet the criteria of the filter for the specified time period.

Benefits of Measuring your requirements data

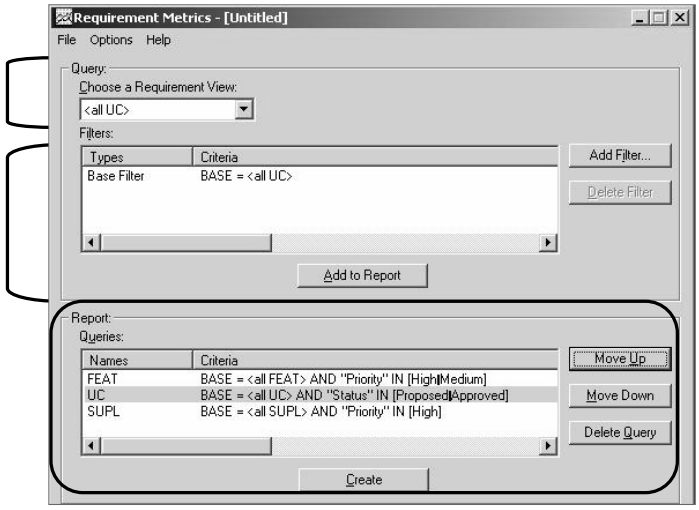
- Trend analyses
- Completeness and coverage reports
- Project status reports

Project managers and system analysts extract information from a Rational RequisitePro project and use the output to convey project status, progress, and expected performance.

Meaningful data is dispersed to team members, management, and customers.

Basic filters for query

- Create criteria for retrieving requirement data.



23

Begin by creating one or more filters. A filter creates criteria for retrieving requirement information. For example, you would use an Attribute Count Filter to determine how many requirements in the project have a Priority value of High. You would then combine one or more filters to produce a query. A query combines the criteria from multiple filters to analyze requirements. The filters that compose a query are joined with the AND statement. Finally, you combine one or more queries to produce a report.

All data in the project that is available for creating a view is available for metrics analysis, including:

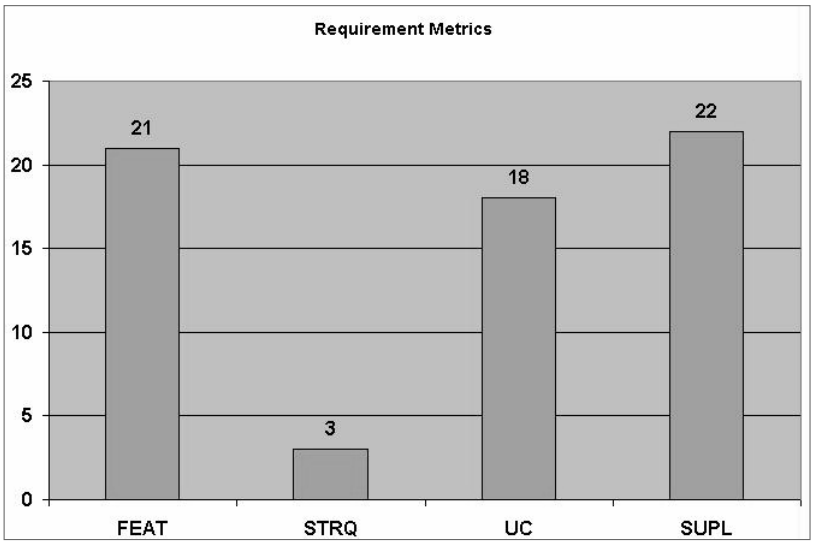
- Requirement text
- Attributes
- Traceability changes
- Hierarchical relationships
- Revision history

Once you have finished creating your filter, click the **Add to Report** button, and your filter becomes a query for the report. Multiple queries can be added until your statistical criteria are complete.

Finish by clicking **Create**. Your report is generated in Excel, which automatically opens on your desktop.

View reports in Excel

- As a snapshot view of project statistics.



24

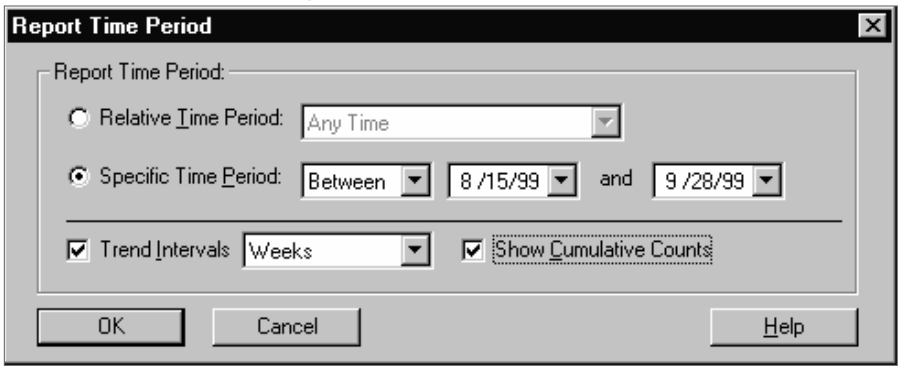
Once you have perfected your queries and filters for a report, the metric criteria can be saved and rerun throughout the project's lifecycle. You can edit the saved report at any time.

If a change is made to the requirement attributes it may make a saved report obsolete. Should this happen, Rational RequisitePro displays an error message when the report is requested.

When you save the report (metric criteria), it is saved with the file extension *.rqm.

Time-sensitive filters

- Analyze changes over time.
- Provide report time period options.
  - Capture data from a specified date range.
  - Provide trend intervals for days, weeks, months.
  - Accumulate requirements for each time interval.



25

IBM

Time-sensitive filters are used for trend analysis reports. A report is configured by the user to capture data during a specific time frame of development. Additionally, the report data can display a variety of time intervals and return a cumulative number of requirements for each time interval selected.

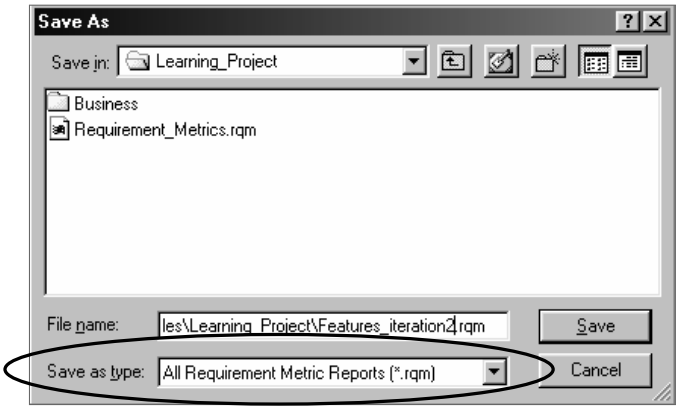
In addition to setting the **Report Time Period** options, you can request the output to detail the requirements. In this manner, you can have both the chart output of the queried report and the requirement text.

The Excel report is created in one worksheet. A summary detail worksheet is also created, which lists all requirements and text that apply to the query.

Trend analysis reports are very helpful in the management of project status. A trend analysis report is not created in the labs of this course because the class project is only one day old.

Save reports for reuse

- Create a report, save, and rerun.
  - Save a report
    - Click **File > Save As**
      - Report, queries, and settings saved to *.rqm file



26

IBM

After you have perfected your filters and queries for a report, you can save and rerun them throughout the lifetime of the project.

To save a report, click **File** > **Save Report** in the **Requirements Metrics** dialog box.

Saved reports are edited and modified as easily as they are created. Be careful when editing a saved report. If any changes to the requirement types or attributes are made, the saved report may become obsolete. An error message appears if an obsolete report is requested.

Lab 6: Metrics

- Create a static analysis report.
- Apply filters and queries.
- Format report output in Microsoft Excel.
- Create a Dynamic Query (Optional).

27

IBM

See Student Workbook Lab 6.

Goal: Create some filtered queries to see project status, and show project data via Excel.

RequisitePro baseline

- A RequisitePro baseline is a snapshot of project requirements at a particular point in the development life cycle.
- Use baselines to:
  - Determine what has changed over time.
  - Generate reports about what has changed.
  - Create RequisitePro projects based on full project baselines.

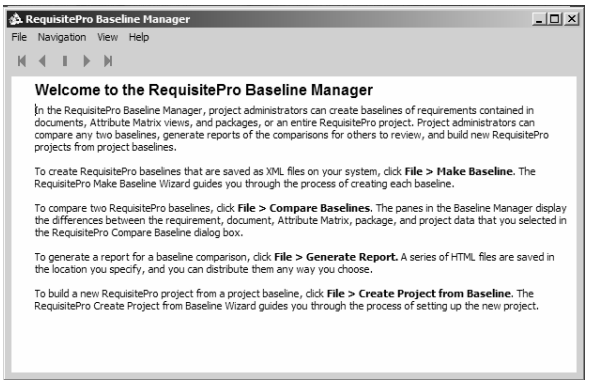
28

In a typical project, requirements often change. Baselines allow you to capture a snapshot of requirements at major project milestones.

The baseline captures information for artifacts up to the point in time in which the baseline was created. For a project baseline, historical references of all revisions up to the time the baseline is created are included in the baseline.

RequisitePro Baseline Manager

- RequisitePro Baseline Manager is a tool in RequisitePro for project administrators to:
  - Make baselines
  - Compare baselines
  - Generate reports
  - Create a new RequisitePro project from an existing project baseline
- Launch the RequisitePro Baseline Manager from the:
  - RequisitePro Tools menu
  - Windows Start menu
  - Command line



29

There are three ways to launch the RequisitePro Baseline Manager:

1. In RequisitePro, click **Tools** > **RequisitePro Baseline Manager**.
2. In Windows, click **Start** > **Programs** > **Rational Software** > **Rational RequisitePro** > **RequisitePro Baseline Manager**.
3. From a Command Prompt window type:
   ```
   <cmd-context> \Program Files
   \Rational\RequisitePro\bin\BaseComp.exe
   ```

To learn more about the RequisitePro Baseline Manager, see the Web-based training course called *REQ315 Essentials of Using IBM Rational RequisitePro Baseline Manager*.

Types of baselines

Baseline type	Description
Project baseline	A snapshot of all the project data
Package baseline	A baseline of requirements, views, and documents in one or more packages
Document baseline	A baseline of the requirements in one or more documents
Attribute Matrix view baseline	A baseline of the requirements of the query results for one or more Attribute Matrix views

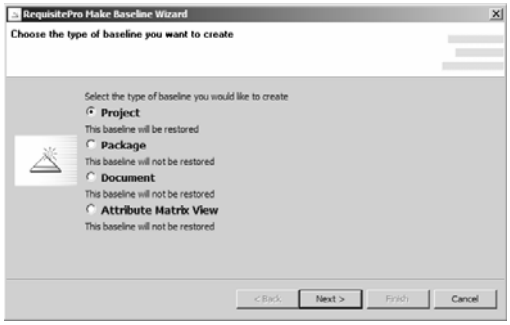
30

When you make a project baseline, the RequisitePro Baseline Manager:

- Converts a project's requirements data to XML files
- Saves requirement documents in a RequisitePro project (if any) with .DOC extension

After creating a RequisitePro baseline, you can add the baseline directory to a source control system, if desired.

Making a baseline



Click **File > Make Baseline**

- Use the RequisitePro Make Baseline Wizard to create a baseline.
  - Select the type of baseline that you want to create and which project to use.
  - Select the type of information to include in the baseline such as traceability, history, and discussions.
  - Create XML files in a subdirectory named with the label you specify.

31

IBM

Before making a baseline ensure that:

- You or the user who will make the baseline belongs to the administrator group.
- The RequisitePro project is closed. No one, including the person who is creating the baseline, can have the project open

Consider making a baseline:

- After requirements are reviewed and approved.
- At the end of each iteration.
- When requirements are frozen going into a major product release.
- When you want to create a new project based on an existing project baseline.

Baseline options:

- Traceability

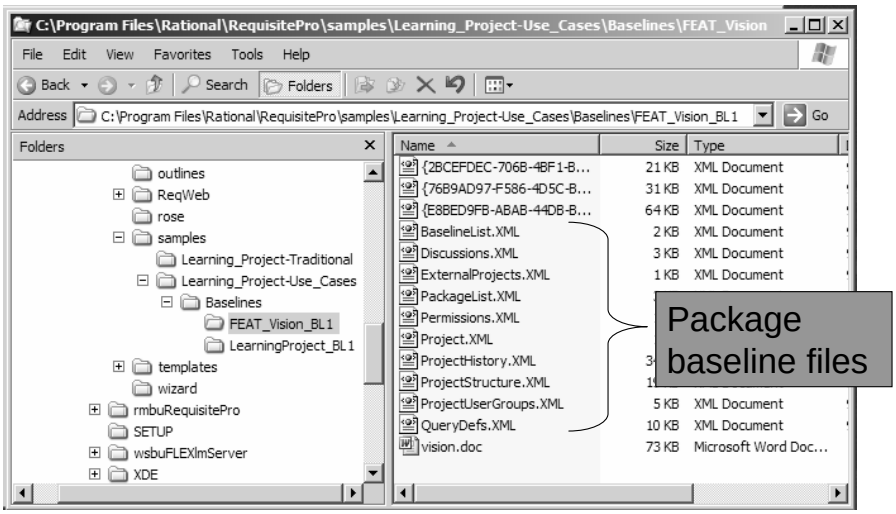
  This option is checked by default for project baselines. Include this option in other baseline types when you want to identify traceability changes between requirements.
- Cross-Project Traceability (CPT)

  Include this option when you want to compare changes in cross-project traceability. Cross-project traceability allows you to establish a relationship between two requirements that reside in different projects.
- History

  Include this option when you want to find out who has made a change, when, and why.
- Discussions

  Include this option when you want to find out if there are any new discussions or new replies to existing discussions.

Example of baseline directories and files



Before creating baselines, plan a baseline directory structure and labels that will help you organize the baselines and identify the contents of each subdirectory.

Labels

Use baseline labels to:

- Name a directory for the XML files that are created when making a baseline.
- Identify baselines when comparing baselines.
- Identify which artifacts are included in the baseline.

Labels help you identify which baselines are older and which are newer. This information is important when comparing baselines.

Before making baselines, choose labels that clearly represent the information for that particular baseline. Apply consistent naming conventions for all your labels.

Directory Structure

The default location for storing baseline files is `/<RequisitePro project location>/Baselines`. Or, you can choose to store baseline files in another location, such as a shared drive. Example of a baseline directory structure:

```
c:/<RequisitePro project location>/Baselines
    /AttributeMatrixViewBaselines
                /AllFeaturesBaselines1
                /AllFeaturesBaselines2
    /DocumentBaselines
                /AllDocs03072006
                /AllDocs06092006
    /PackageBaselines
                /UseCasesPkgInceptionPhase
                /UseCasesPkgElaborationPhase
    /ProjectBaselines
                /LearningProjectIteration_1
                /LearningProjectIteration_2
```

Comparing baselines

- It is important to communicate requirement changes to the appropriate stakeholders, so that you can get their input on how the changes may affect the project.
- By comparing baselines, you avoid the risk of missing requirement changes.

33

IBM

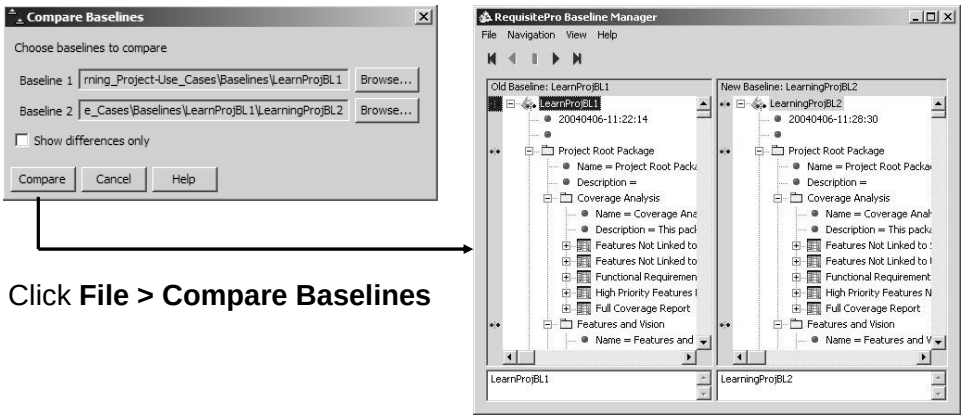
Requirements change. It is important to communicate changes to the appropriate stakeholders and get their input on how these changes may affect the project. Unmanaged changes may turn into project risks that can cause scope creep and jeopardize your project.

Comparing baselines identifies requirements changes made between two snapshots of your RequisitePro requirements documents in Word, Attribute Matrix views, packages, or projects. Comparing baselines helps you avoid the risk of missing requirement changes, which helps you:

- Effectively communicate changes to stakeholders
- Avoid project scope creep
- Mitigate project risks

Comparing baselines (cont.)

- The two baselines you choose to compare must be:
  - From the same project
  - Of the same baseline type
- The Compare Baselines results display in side-by-side panes.



Click **File > Compare Baselines**

34

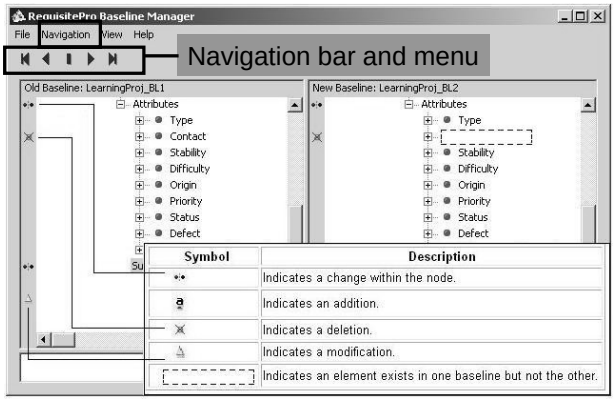
The two baselines that you choose to compare must be from the same project and be of the same type, such as two document baselines, two project baselines, two package baselines, or two Attribute Matrix views baselines. You cannot compare baselines of different types. For example, you cannot compare a project baseline to a package baseline.

Project structure, History, and Discussions are compared automatically. Packages, views, and documents are compared when you expand them in the compare tree.

The Project Root Package is the container and the starting point for all user-defined packages. When you compare project baselines, Project Root Package is not compared automatically. You need to initiate the comparison of the Project Root Package by expanding the package.

Icons in the compare panes

- Icons in the margins identify elements that have changed:
  - Additions
  - Deletions
  - Modifications
  - Moves (identified as a deletion in one location and an addition to another location)
- Use the Navigation bar or the Navigation menu to move to the differences identified by the icons.



35

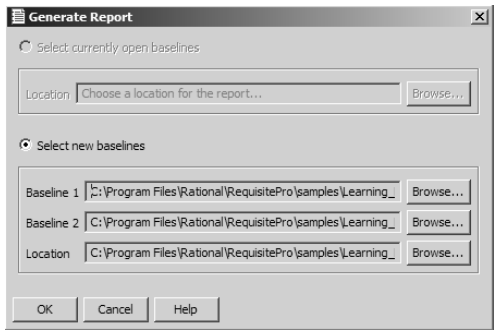
You can use the Navigation bar or the Navigation menu to move to the differences identified by Compare Baselines. The Navigation bar and Navigation menu are disabled until a difference between the two baselines is identified. You can also navigate manually through differences in the compare tree.

Actions of the Navigation bar buttons (from left to right):

- Jumps to the first difference (of what has been compared)
- Jumps to the closest difference preceding the selected node
- Scrolls to and expands the currently selected node
- Jumps to the closest difference following the selected node
- Jumps to the last difference

Generating a baseline comparison report

- A baseline comparison report enables you to:
  - Keep a record of your baseline comparison results.
  - Share comparison results with stakeholders who may not have access to RequisitePro Baseline Manager.



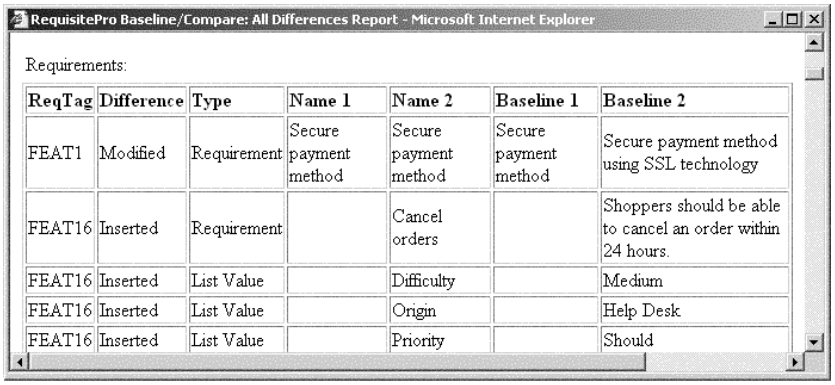
Click **File > Generate Report**

36

You can generate a comparison report by clicking **File > Generate Report** in the RequisitePro Baseline Manager. You can also run reports in batch mode.

Generating a baseline comparison report (cont.)

- The baselines that you select are compared.
- A series of HTML files are created; the files vary depending on the type of baselines compared.
- The report identifies only requirement artifacts that have been modified.



37

When you generate a baseline comparison report, Baseline Manager creates a series of HTML files. The files generated vary depending on which baselines are compared.

In the report example above, notice the changes between baseline 1 and baseline 2. Existing requirements were modified and new requirements were added.

Lab 7: RequisitePro Baseline Manager

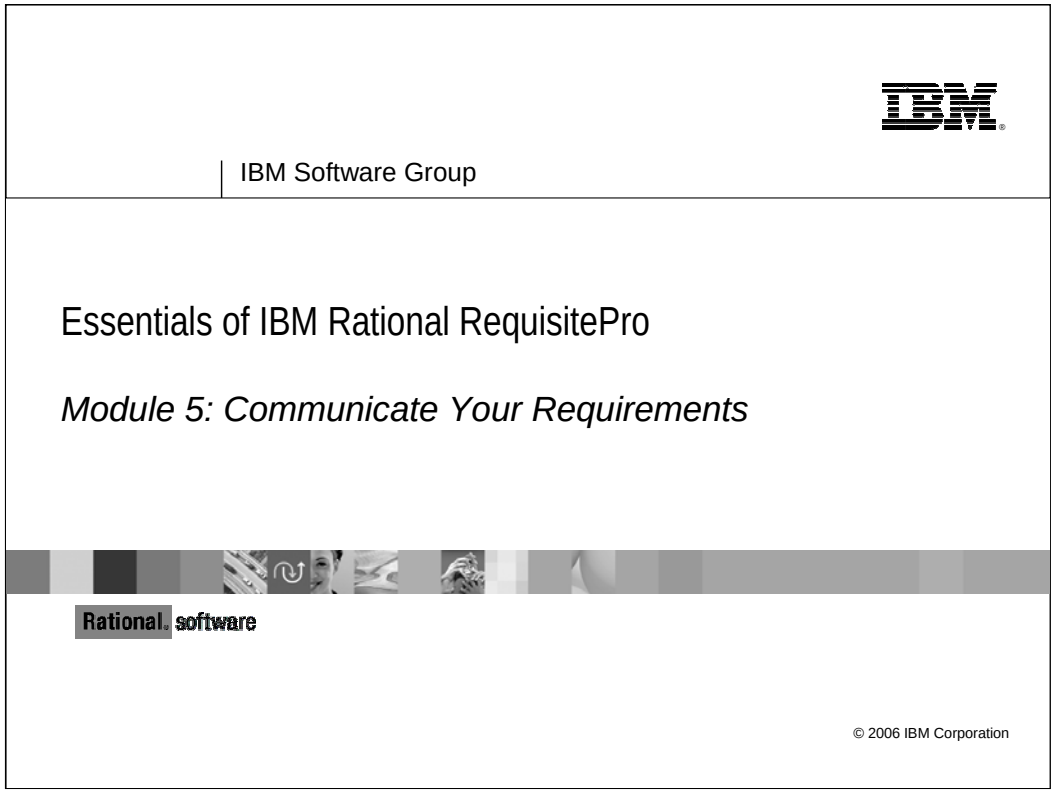
- Create baselines.
- Compare baselines.
  - After completing this lab you will review the differences that you find with your instructor.

38

IBM

See Student Workbook Lab 7.

Goal: Create project, package, attribute matrix view, and document baselines. You will make changes to the project and create new baselines. You will then use the Compare Baseline tool to identify the changes in the baselines.



Contents

Objectives: communicate your requirements	5-2
Requirement revision notification	5-4
Enabling revision notification	5-5
Subscribing to requirement revision notification	5-7
Team communication using discussion groups	5-9
Create discussions	5-11
Identifying discussions in Rational RequisitePro	5-12
Viewing discussions	5-13
Querying discussions	5-14
Lab 8: Group Discussions	5-17

Objectives: Communicate your requirements

- Explain what revision notification is.
- Know how to register to a requirement and get revision notification by e-mail.
- Participate in discussion groups.

2

In this module, you will learn how to use features of RequisitePro to manage changing requirements and to facilitate team communication. You will learn how to enable revision notification so that you can be alerted when requirements you are interested in change. You also will learn how to use the Discussion Groups feature to facilitate communication within the project team.

Team communication

- Requirements define what the system must do; therefore, it is important that you involve different stakeholders in requirements management, and keep them informed of changes made to the project and requirements.
- RequisitePro includes these mechanisms for communicating with team members and stakeholders:
  - Revision notification
  - Discussion groups

3

IBM

Keeping the project team and business stakeholders involved in requirements management can be a time consuming task. RequisitePro provides revision notification and discussion groups to aid team communication.

Revision notification enables users to subscribe to requirements and receive e-mail notification when changes occur to them. Discussion groups enable you to address comments, issues, and questions to a group of discussion participants. Discussions can be associated with one or more specific requirements, or they can refer to the project in general.

By understanding the impact of change, you are much better equipped to control changes that affect your project.

Requirement revision notification

- RequisitePro provides e-mail notification when changes occur to the requirements that you subscribe to
  - Project administrators enable revision notification
  - Users subscribe to requirements to receive change notifications by e-mail
  - RequisitePro monitors projects for revisions at specified times or intervals
  - When RequisitePro detects revisions that have occurred since the last notification, RequisitePro sends an e-mail message containing the change information to subscribed users of the requirements that have changed

4

Enabling revision notification

- The project administrator sets up revision notification
  - Which projects RequisitePro should monitor for changes
  - E-mail protocol to be used
  - E-mail address for each project
  - E-mail alert frequency

5

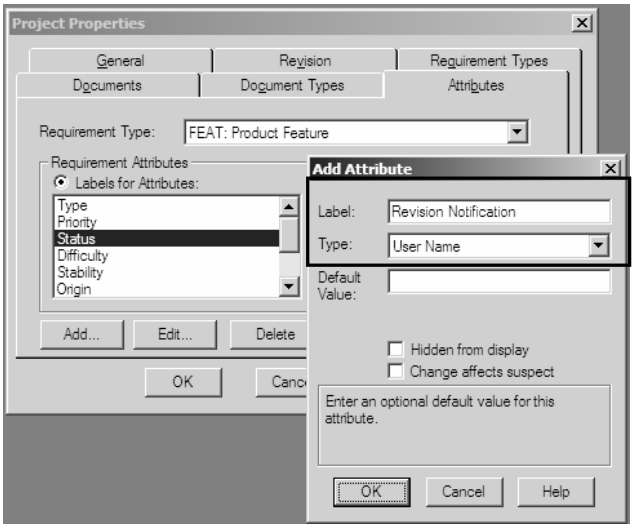
IBM

You can access this tool by double-clicking the Revision Notification executable file in the default location of:
C:\Program Files\Rational\RequisitePro\bin\RqReqEmailConfig.exe

For SMTP protocol, the administrator specifies an e-mail address to identify the sender of the e-mail notifications. The administrator can create an e-mail account for each RequisitePro project to indicate the project as the sender.

Enabling revision notification (cont.)

- The project administrator adds an attribute of type User Name with a label of Revision Notification to each requirement type in a project

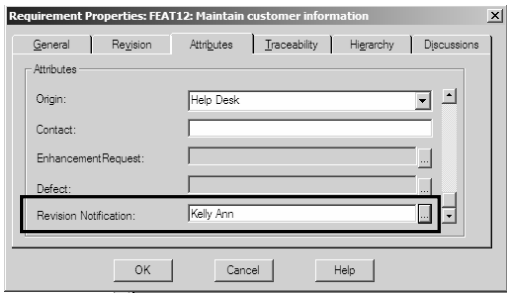


6

This step is required to enable users to subscribe to a specific requirement for revision notification.

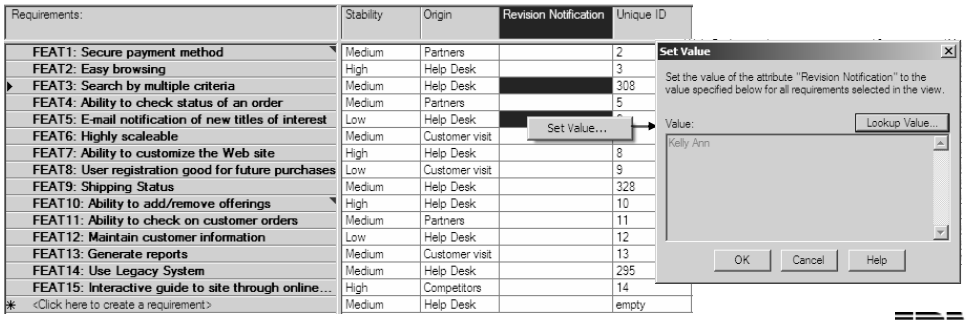
In a RequisitePro project, click **File > Project Administration > Properties > Attributes**.

Subscribing to requirement revision notification



You can subscribe to a requirement through its Requirement Properties dialog.

You can subscribe to multiple requirements in an Attribute Matrix view.



7

Anyone who has an interest in a specific requirement should consider subscribing to it. The subscriber could be an architect, a designer, a developer, a tester, a project manager, a user, or a business stakeholder.

The user name you enter should be the name listed in the Security dialog, **File > Project Administration > Security**.

Use the **Properties** box to subscribe to a single requirement. Subscribing in an **Attribute Matrix** view enables you to subscribe to multiple requirements at once.

Subscription is based on the user who is logged in at the time. If a stakeholder wants to add another account to the subscription list, the stakeholder types the user name of the person who needs to be subscribed. For multiple users, each user name must be separated by a comma.

Sample revision notification e-mail message

Subject: Requirement revision - Learning Project - Use Cases: SUPP1
SUPP1: Interface guidelines

Revision2
Revision #: 1.0018
Version Label:
Date Time: 2004-07-01 13:00:04
Author: Kelly Ann
Change Description: Created trace relationship to FEAT11.
Created trace relationship from FEAT3.

Revision1
Revision #: 1.0017
Version Label:
Date Time: 2004-07-01 12:59:48
Author: Pat
Change Description: DIFFICULTY: Medium - Low.
Requirement Text Changed.
REQTEXT: The system shall follow standard interface guidelines.

Tag, Name
SUPP1: Interface guidelines

Text, Package, Location
Text: The system shall follow standard interface guidelines.
Package: Project Root Package\Supplementary Requirements
Location: C:\Program Files\Rational\RequisitePro\samples\Learning_Project-Use_Cases\Supplementary Specification.SUP

Attributes
Priority:
Status: Validated
Difficulty: Low
Stability: Low
Revision Notification: Nabil, Mario

Traceability
Traces from: FEAT3
Traces to: FEAT11

Hierarchy
Parent: None
Children: None

8

Revision notification e-mail messages contain information such as:

- Project name and e-mail address
- Requirement tag
- Requirement name
- Revision number
- Version label
- Date and time of the changes
- Author who made the revision
- Change description
- Current values for all properties of the requirement

The top section of the e-mail message lists all new revisions for the requirement. The last revision is listed first.

The bottom section (below the dotted line in the example) lists requirement properties: General, Attributes, Traceability, and Hierarchy.

Team communication using discussion groups

- Engages team in dialogs.
  - Related to:
    - One or more requirements
    - Whole project
  - Tree-like display of comments and replies
- Occurs within Rational RequisitePro or through e-mail.

9

Tracking issues and questions related to requirements can be a difficult task. Discussion groups provide the ability to attach discussions to the project as a whole, or to one or more specific requirements. Once a discussion is defined, it is added to the project database and is graphically represented in Rational RequisitePro as a tree-like display. The discussion can then be viewed in Rational RequisitePro by all team members.

Additionally, all discussions can be queried to assist you in managing your discussions and isolating only those discussions meeting your defined query criterion.

Communicate requirements

- Support:
  - Large teams
    - Simplify the gathering of information
    - Optimize the input process
  - Multi-user environments
  - Change proposal process

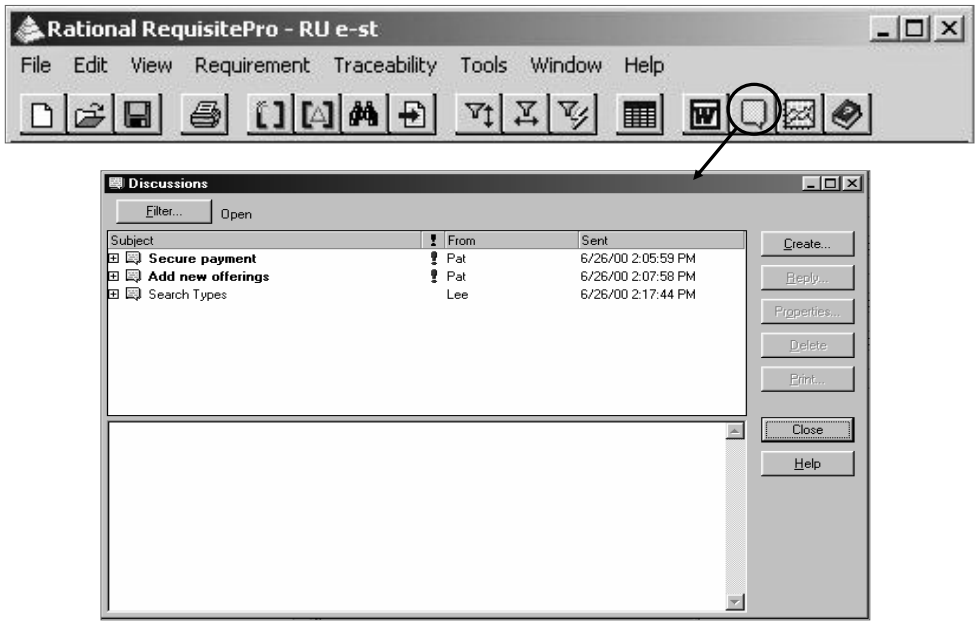
10

IBM

Discussion groups allow the managed exchange of critical information or notification of proposed changes to a project or requirement.

With e-mail enabled, specified team members are informed of proposed changes or issues regardless of their access to Rational RequisitePro. This provides a solution for development teams in a distributed environment and also allows traveling team members to stay current on important facets of the project. Similarly, key customers who do not have a copy of Rational RequisitePro can be added to these discussion groups to ensure that the development stays on target with the stakeholder needs.

Create discussions



11

IBM

You can create a discussion by clicking **Show All Discussions** > **Create**.

The **Discussion Property** dialog box appears with fields in which you can type a discussion title and text. You can also assign priority, participants, and associate the requirements to the discussion.

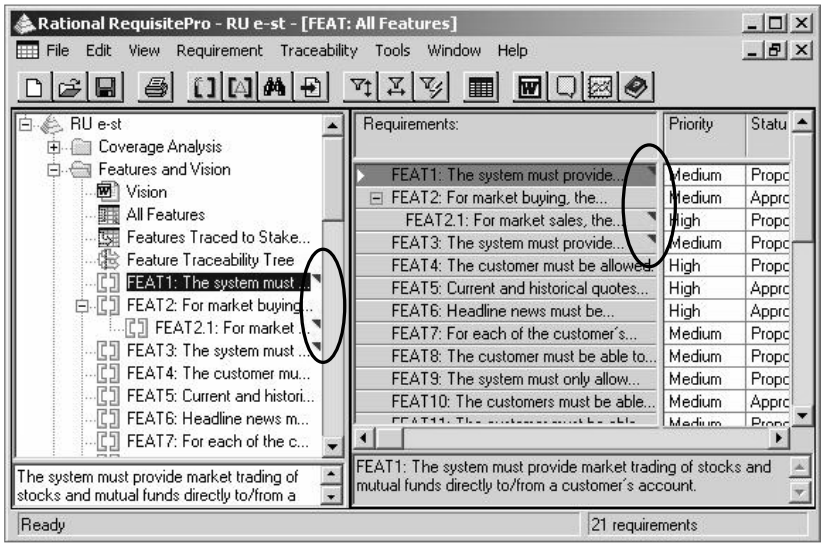
Participants may be assigned at a group level or to individual users.

A much faster way to create a discussion is to right-click a requirement in the Explorer, and then click **Discussions**.

This automatically associates the discussion with the requirement.

Identifying discussions in Rational RequisitePro

- Discussion indicators in views and Explorer.



12

IBM

Discussions record ideas about anything in the project. A discussion can be centered around multiple requirements or no requirements at all. In the latter case, the discussion is associated with the entire project.

When a requirement is associated with a discussion, a red triangle is displayed in the Explorer or in any view type to the right of the requirement text.

Even though a requirement is associated with a discussion, it may be manipulated by other members of the development team. Team members may update the requirement, set or remove traceability, or perform any other necessary editing. Attaching a discussion to a requirement does not affect or prevent changes to it.

All participants can view and read discussion items in a Rational RequisitePro project.

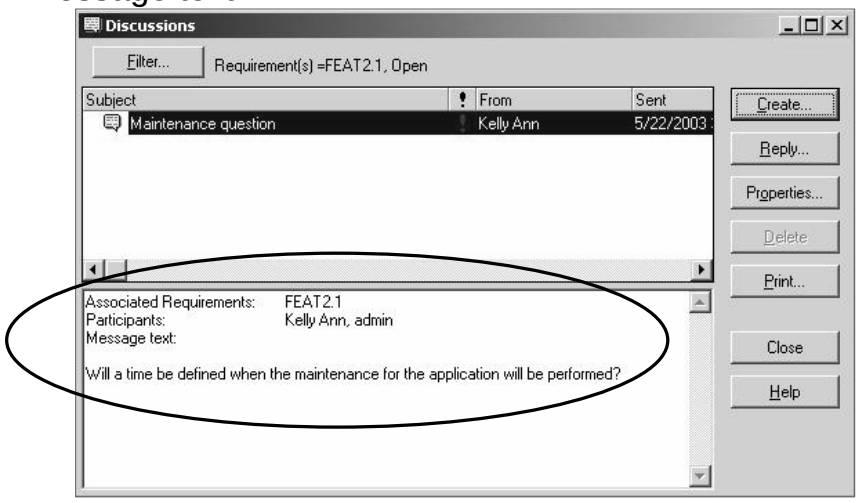
To access a discussion for a requirement, do one of the following:

- Click the small red triangle to open the **Discussions** dialog box.
- Right-click the requirement that has a discussion and select **Discussions**. You can do this from either the Explorer or the Views workplace.

This is a method of filtering the display of discussions, since you are presented with only those discussions that apply to that particular requirement.

Viewing discussions

- Associated requirement
- Defined participants
- Message text



13

Each discussion topic is listed in the **Discussions** dialog box, along with information regarding its origin. When a discussion topic is selected, the lower pane of the **Discussions** dialog box displays information regarding the topic.

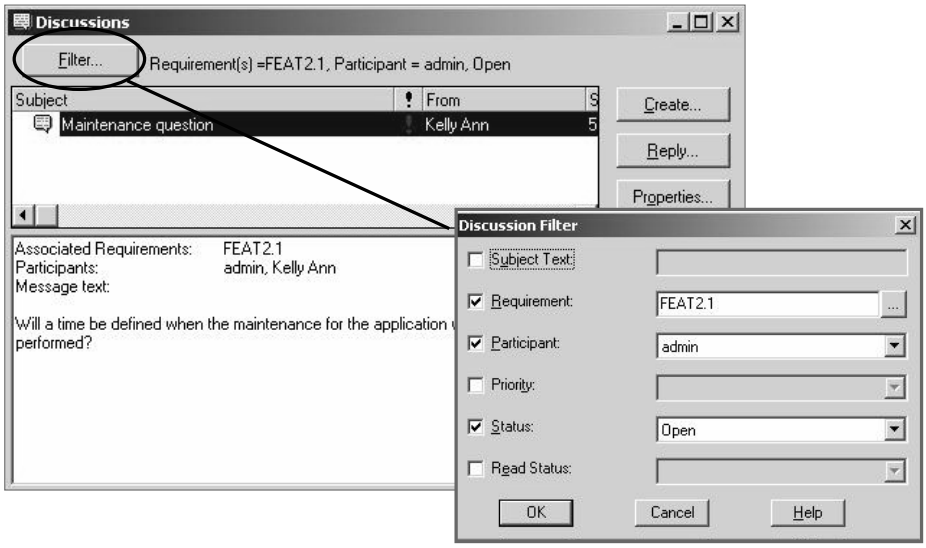
When a team member creates a new discussion, he or she has the opportunity to restrict the discussion to certain defined participants. If the discussion is restricted to participants, all team members may view the discussion item and responses, but only those who are participants can take an active part in the discussion.

Discussions can be e-mail-enabled. When e-mail is activated for a discussion, the participants must be defined in Rational RequisitePro before any e-mail can be sent.

All users can view and read discussion items in a project. Participants in discussion groups can create and reply to discussions in Rational RequisitePro.

Querying discussions

- Flexible discussion query mechanism.



14

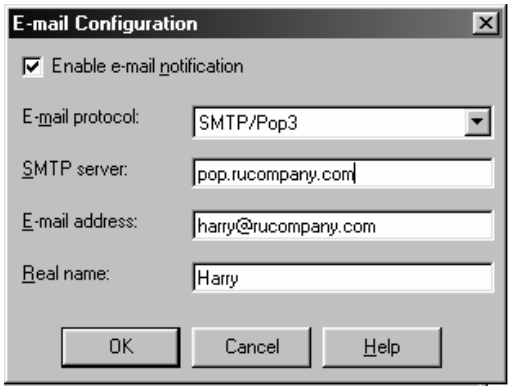
IBM

You can use filters to help you manage your discussions. Filters help you locate specific discussions by **subject**, **requirement**, **participants**, **priority**, **status** or **read status**.

By default a discussion that has been closed is not displayed when you open the **Discussions** window. To view closed discussions you must use a filter and select a status of **closed**.

Discussion through e-mail

- Promotes team collaboration.
- Automatic distribution of discussions.



15

Discussions can be set up so that participants are notified by e-mail when a discussion involves them. Participants are also able to reply to the e-mail and have their comments added to the discussion thread in Rational RequisitePro.

E-mail can be set up to use either SMTP/POP3 or MAPI mail servers. Consult your network administrator to find the name of your e-mail server.

To set up your e-mail properties, click **Tools** > **E–mail Setup**.

Communicate requirements across the organization

- Cross-Project Traceability.
  - Trace any requirement that bridges multiple projects:
    - Corporate business rules
    - Legislative requirements

16

Cross-project traceability allows you to establish traceability between requirements that reside in different projects. This is useful when you want to share requirements from one project to the next. Same examples of cross-project traceability are:

- Sharing business rules across multiple projects.
- Capturing regulatory requirements that impact all your projects.
- Sharing customer requirements across multiple projects. This usually is needed when you have a "system of systems."

To use cross-project traceability, you first open each external project and mark the requirement types in the projects for external traceability. Then you open the primary project and connect the external projects to it. Once you make the connection, you can add traceability relationships for all requirements of the marked requirement types.

External requirements are identified with a prefix that you set in the **General** tab of each project's **Project Properties** dialog box.

Lab 8: Group Discussions

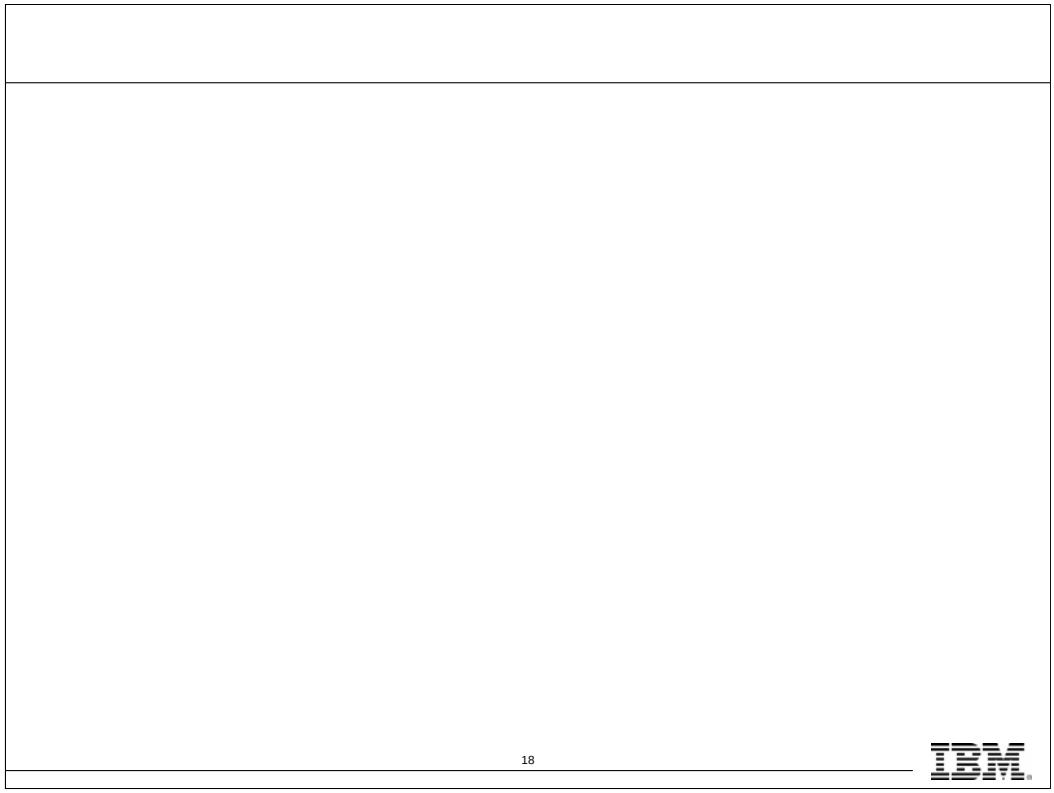
- Create and reply to a discussion group within Rational RequisitePro.

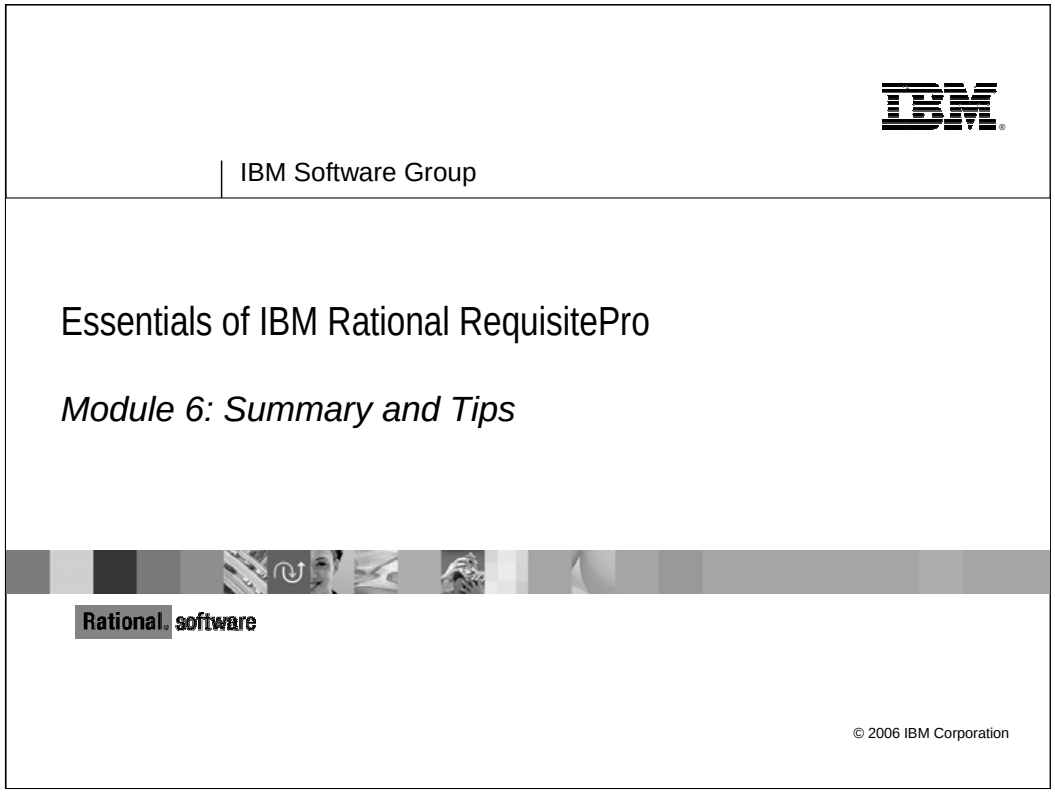
17

IBM

See Student Workbook Lab 8.

Goal: Create a discussion.





Contents

Summary	6-2
Qualities of software requirements sets	6-6
Tips for writing good requirements	6-8
RequisitePro tip: renumber requirements option	6-12
RequisitePro tip: capabilities for documents	6-13
Other sources of information	6-14

Summary

- Connected to an existing project.
- Customized the project structure.
- Created, imported, modified, and deleted requirements.
- Defined traceability links, assigned attribute values, and created views.
- Imported, modified, and created documents.
- Queried requirements and produced metric reports.
- Communicated requirement information via group discussions.

2

This slide presents a high-level summary of the activities that you performed today.

Summary: Plan your project

- Requirements Management Plan.
  - Document types
  - Requirement types
  - Attribute types and values
  - Traceability
- Project Infrastructure.
  - Security
    - Who needs write access to what?
    - What type of user authentication is required?
  - Database selection
    - Estimate Repository size

3

The Requirements Management plan is central to a coordinated approach to requirements management. In a nontrivial project, an unplanned RM process is a recipe for disaster.

The infrastructure that you need to support your project is also important to consider. If you have a large number of requirements, choose an enterprise database, such as IBM DB2. Microsoft Access will not scale to large projects. Refer to the Rational RequisitePro Help for further details.

Rational RequisitePro also allows you to establish security to ensure that access and changes to your requirements is controlled. You can use the user authentication provided by RequisitePro or you can use LDAP authentication.

Summary: Gather, organize, and document

- Where are your requirements now?
  - What can you import vs. create manually?
- Support the management process.
  - Attributes, metrics
- Organize your requirements.
  - Packages, hierarchy

4

Getting your requirements into Rational RequisitePro is one of the first activities you must perform. Are they stored electronically? If so, Rational RequisitePro provides many features to facilitate the import process.

Always strive to write quality requirements. Can you validate that a requirement has been met? Is there only one interpretation for each requirement? Do any requirements contradict each other?

When determining the attributes to be assigned to each requirement, remember that someone has to maintain them. A simple approach is always best. You must also consider the types of information that management is going to require from the repository.

Having all your requirements in one giant bucket can make them difficult to manage. Packages and hierarchy can help you organize your requirements so that they are easy to locate and manage.

Summary: Manage your requirements

- Traceability.
  - Analyze the impact of change.
  - Control feature creep.
- Metrics.
  - Prioritize requirements.
  - Assess project status.
    - Static analysis using filters
    - Trend analysis using time-sensitive filters

5

Managing the requirements that you have is important. Rational RequisitePro provides a number of different views that allow you to see your requirements from a different perspective.

A traceability matrix allows you to see a two-dimensional view of dependencies between requirements.

An attribute matrix allows you to view all of the requirements by the values that their attributes contain. Filtering this view allows you to see things like highest priority requirements, and so on.

A traceability tree shows you the dependencies between requirements at multiple levels.

Metrics can be obtained to show things such as the rate of changes to your requirements.

Qualities of software requirements sets

- Correct.
  - Is a true statement of something the system must do.
- Complete.
  - Describes all significant requirements of concern to the user.
- Consistent.
  - Does not conflict other requirements.
- Unambiguous.
  - Is subject to one and only one interpretation.

ref – Leffingwell & Widrig (1999). IEEE 830-1993, § 4.3.2, 1994

6

Correct

- Does every requirement state something required of the system?

  "A set of requirements is correct if and only if every requirement stated therein represents something required of the system to be built." *Davis (1993)*

  It is not possible to determine if a requirement is correct simply by reading the requirement. The correctness is verified by a subject matter expert during a review.

Complete

- Does the set of requirements include all significant requirements, whether related to functionality, performance design constraints, attributes, or external interfaces?
- Have the expected ranges of input values in all possible scenarios been identified and addressed?
- Have responses been included to both valid and invalid input values?
- Do all figures, tables, and diagrams include full labels and references and definitions of all terms and units of measure?

Consistent

- Is it internally consistent, with no subset of individual requirements described which are in conflict? (E.g. Vision document, the use-case model and the Supplementary Specifications)

Unambiguous

- Does each requirement have one, and only one, interpretation?

Qualities of software requirements sets (cont.)

- Verifiable.
  - Can be tested cost effectively.
- Ranked for importance and stability.
  - Can be sorted based on customer importance and stability of the requirement itself.
- Modifiable.
  - Changes do not affect the structure and style of the set.
- Traceable.
  - The origin of each requirement can be found.
- Understandable.
  - Comprehended by users and developers.

ref – Leffingwell & Widrig (1999). IEEE 830-1993, § 4.3.2, 1994

7

Verifiable

- Is every requirement stated verifiable?
- Is there some finite cost-effective process with which a person or machine can check that the software product meets the requirement?

Ability to Rank Requirements

- Has each requirement been tagged with an identifier to indicate either the importance and stability of that particular requirement?

Modifiable

- Are the structure and style of the requirements in the software requirements set (use cases, supplementary specification; or software requirements specification) such that any changes to the requirements can be made easily, completely, and consistently while retaining the structure and style?
- Has redundancy been identified, minimized, and cross-referenced?

Traceable

- Does each requirement have a clear identifier?
- Is it distinguishable from non-essential statements in the requirements set?
- Is the origin of each requirement clear?
- Is backward traceability maintained by explicitly referencing earlier artifacts?
- Is a reasonable amount of forward traceability maintained to artifacts spawned by the requirements set? For example, test cases.

Tips for writing good requirements

- Write sentences that are:
  - Complete.
  - Simple.
  - In the active voice.
- Ensure that the requirement is:
  - Non-conflicting.
  - Verifiable or testable.
- Write “what” shall be done and not “how” it will be done.

8

Non-conflicting: Focusing on each requirement as if it were an independent entity can easily lead to conflicting requirements for a given system. Organizing your requirements based on the requirement types discussed earlier helps determine assumptions and find possible conflicting requirements.

Complete sentences: Keep in mind that clearly communicating requirements to all stakeholders is our primary goal. If the requirements are difficult to read, the possibility of misinterpretation is increased.

Simple Sentences: Grammatically, a simple sentence contains a subject and predicate. When a simple sentence is used in a software requirement, the subject should identify the person or system that is being discussed.

Active voice: In the active voice, the subject is the agent of the action denoted by the verb. In the passive voice, the subject is the object of the action denoted by the verb.

Verifiable and testable: A key principle for writing a good requirement is to write it such that it can be verified or tested. This means that the requirement, if tested, could yield a “success” or “failed” result. Whenever possible, include metrics (something that is measurable or quantifiable) in your requirement so that what is expected from the system is absolutely clear.

“What,” not “How”: Another key principle to writing a good software requirement is to always write “what” the system will do and NOT “how” the system will do a particular task. By focusing on the “how,” the author falls into the trap of designing the system through requirements.

Signs of unclear or poorly written requirement

- Undefined jargon is used.
- The word “use” is used.
- Conjunctions can be found.
- Exception statements are used.
- Graphical depictions are used in lieu of a detailed textual description.
- Generalizations are used.
- Relative terms are used.
- Suggestive terms are used.
- Clarifications are used.
- The word “not” is used.

9

IBM

Undefined “jargon”: Plug-n-play, point-and-click, WYSIWYG.

The word “use” is used: Example (vague): The system uses the sensor to make forecasts.

Example (clear): The system reads the barometric pressure from the sensor and applies the value to the forecast formula.

Conjunctions can be found: Use of: and, or, also, with.

Exception statements are used: Use of: if, but, when, except, unless, although.

Graphical depictions are used in lieu of a detailed textual description: Graphical depictions should only be used to augment a detailed textual description.

Generalization terms are used: Examples: Generally, usually, often, normally, typically, “as possible”, approximately, paradigm.

Relative terms are used: Examples: User-friendly, fast, flexible, adaptive, intuitive.

Suggestive terms are used: Examples: Could, should, may, might, maybe, ought, perhaps, probably.

Clarifications are used: Examples: “That is”, “for example”, or “like”.

The word “not” is used: The problem with stating what is NOT allowed is that you haven’t said what IS allowed. If the list isn’t exhaustive you have created a hole in your requirements. It is usually much easier to state what is allowed rather than what is not allowed.

Group exercise: Identify “good” requirements

- Example 1:
  - The system shall have a user-friendly interface to facilitate fast data-entry.
- Example 2:
  - The system shall not allow numbers in a user’s password.
- Example 3:
  - The user shall select Login and the system will display a dialog box allowing the user to type in a User ID and Password.
- Example 4:
  - The system shall generally be accessible 24X7, except for when the database is being updated.

10

Read each requirement example and answer the following questions:

Is it a “good” or “bad” requirement example?

Why?

- Refer to previous slides for hints.

If it is a “poor” example, what could make it a “good” requirement?

If it is a “good” example, why?

Summary: Communicating requirements

- All requirements are maintained in a repository with centralized access.
- Familiar interface:
  - Microsoft Word
  - RequisitePro Explorer
- Revision notification
  - Informs team members and stakeholders of requirement changes that may affect their work
- Discussion groups:
  - Facilitate information transfer among team members.
  - Provide a simple facilitation mechanism.

11

IBM

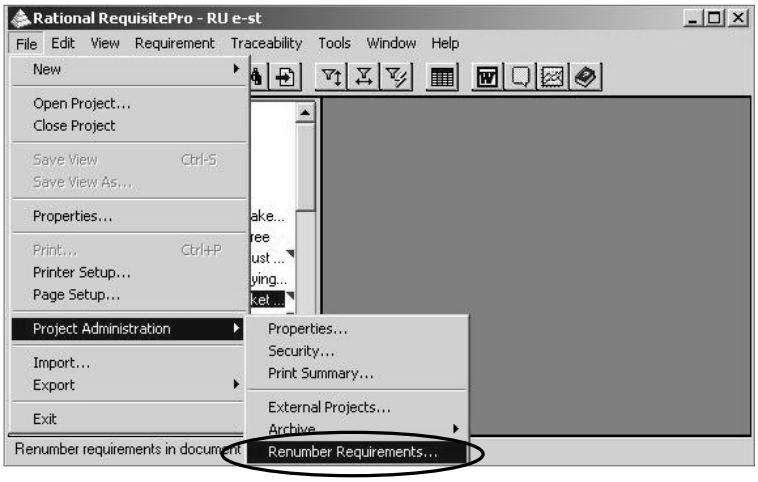
Knowing where your requirements are is at the heart of your ability to communicate them to stakeholders. Rational RequisitePro ensures that all your requirements are located in a central repository. Access to the repository is via a familiar browser and the industry standard word processor, Microsoft Word.

Revision notification allows users to subscribe requirements and receive e-mail notification when changes occur to them. RequisitePro monitors projects for revisions and sends an e-mail message containing the revision information to subscribed users of the requirements that have changed. Everyone with an interest in requirements can be informed of changes that may impact their work.

Discussion groups provide a simple, yet powerful mechanism to disseminate information related to one or more requirements. The information can be accessed through Rational RequisitePro or e-mail.

RequisitePro tip: Renumber requirements option

- Eliminates the "holes" in a numbering scheme.
- Requirements are renumbered in the order in which they appear in a document.



12

The renumbering feature is useful for eliminating "holes" in a numbering scheme, which can result when you delete requirements.

To renumber requirements, you must open the project in exclusive mode.

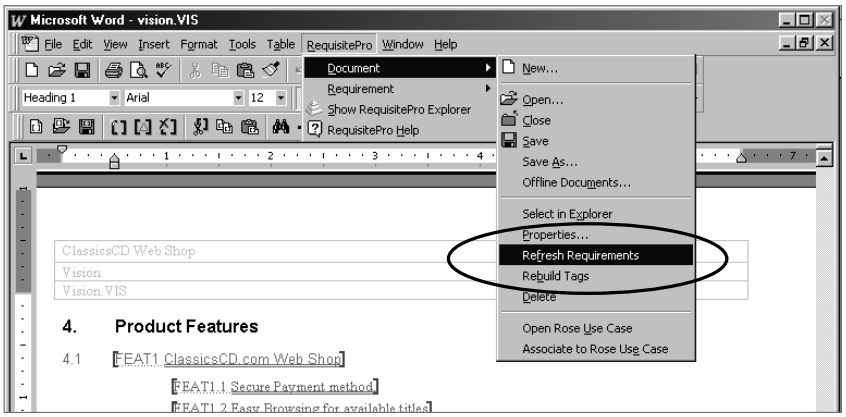
Renumbering requirements also renumbers all requirements of a given type in the entire project, not just requirements in a particular document.

To renumber requirements in documents, first open all documents that contain requirements of that type, and then click

File > **Project Administration** > **Renumber Requirements**.

RequisitePro tip: Capabilities for documents

- Refresh requirements.
- Rebuild tags.



13

Refresh Requirements

You might inadvertently overwrite the style of the requirements. A requirement's color and/or style may be overwritten with the settings attributed to another style in the document.

To restore the original requirement type style and color settings:

In the Word workplace, click **RequisitePro > Document > Refresh Requirements**.

Requirement tags can be accidentally corrupted or deleted while editing a document.

The **Rebuild Tags** command rebuilds a requirement tag that is partially or completely deleted or corrupted. Rational RequisitePro provides a list of rebuilt tags. To rebuild tags:

In the Word workplace, click **RequisitePro > Document > Rebuild Tags**.

To display/not display tags in a document, select or clear the **Show Tags** check box on the **Document General** tab in the **Properties** dialog box for the project.

Other sources of information

- Related courses
  - *REQ480 Requirements Management with Use Cases*
  - *PRJ270 Essentials of Rational Unified Process*
  - *REQ270 Essentials of IBM Rational RequisiteWeb* (classroom training) or *REQ210 Essentials of IBM Rational RequisiteWeb* Web-based training
- IBM® developerWorks® Rational zone
  - www.ibm.com/developerworks/rational/
  - Product documentation, technical articles, online discussions, *the Rational Edge* newsletter, user groups, and more
- Rational Web site
  - www.ibm.com/software/rational/

14

IBM

IBM® developerWorks® (Rational products zone) is an online publication and community that provides articles, tutorials, and a forum to exchange ideas and best practices. Content on the site has been edited for quality and relevance, and it is organized by technology and process. There are also links to Web-based training. You can use developerWorks to:

- Learn about new tools and methods
- Increase proficiency in existing solutions
- Find tips on handling changes in project definition or technology

Rational RequisitePro Glossary

The terms that follow are used throughout *Rational RequisitePro Users Guide*. Most are specific to RequisitePro.

active document:

The document that has processing focus. You can have more than one document open at a time; however, only one of the documents is active.

active view:

The view that has processing focus. You can have more than one view open at a time; however, only one of the views is active. See also *view*.

activity:

A unit of work an individual, or a set of individuals working together as a team, may be asked to perform.

actor:

Someone or something outside the system that interacts with the system.

Add-ins command:

A command on the **Tools** menu that allows you to add your own customized commands to any RequisitePro menu. These menu commands can be used to start external applications, such as e-mail or the Windows Notepad, directly from RequisitePro. In addition, you can configure a command to open an individual file with the application of your choice.

administrators group:

A group of users with full permissions to work in a project. These users can change a project structure, create and modify data, modify and delete project-wide views, and set and maintain security permissions. Users can be added and removed from the group, but the group cannot be deleted. Administrator group permissions cannot be modified.

analyst:

See *System analyst*.

anchor:

The first item you click when using multiple select actions to select a range of items. In RequisitePro, the term *anchor* is generally used to mean the first cell you click in a view. You use the anchor to select a range of cells. For example, if you click the cell in row 1, column 1 of a Traceability Matrix, that cell becomes the anchor. If you then Shift-click the cell in row 10, column 3 of the matrix, all of the cells in rows 1 through 10 and columns 1 through 3 are selected. To reset the anchor, click anywhere outside the range. The cell you click is the new anchor. You can also use an anchor to select a range of items in documents and in certain dialog boxes that support multiple select actions. See *multiple select actions*.

archiving:

The process of duplicating the database, documents, and all related files in a project for the purpose of restoring them at a later time.

attribute:

Descriptive information that provides important details about a requirement or discussion. See also *requirement attribute.*

attribute label:

The name of the requirement attribute, such as risk, priority, or author. Also referred to as *requirement attribute label*.

Attribute Matrix:

A spreadsheet-like table view that displays requirements in rows and the attributes that describe them in columns. You can add or change values in the attribute fields. Also referred to as *Requirement Attribute Matrix*.

attribute type:

A set of descriptive and operational information associated with a requirement attribute when the attribute is created. Attribute values can be list-type or entry-type. For list-type attributes, you

select a textual attribute value from a drop-down list box. For entry-type attributes, you type in a value, such as a number, text string, date, or time. Also referred to as *requirement attribute type*.

attribute value:

Information assigned to a requirement attribute. Attribute values can be text or numbers. For example, the attribute *priority* can be assigned the values Low, Medium, and High. Also referred to as r*equirement attribute value*.

author:

The user responsible for creating or modifying a document or requirement.

author (version information):

The user responsible for the changes associated with a given revision.

.bak file:

A backup of a document file. A new .bak file is created in the same directory each time a document is saved and contains a version of the document before it was saved. You can delete .bak files at any time; they are created whenever the original documents are saved.

baseline:

A Rational ClearCase Unified Change Management object that typically represents a stable configuration of one or more components. A baseline identifies activities and one version of every element visible in one or more components. You can create a development stream or rebase an existing development stream from a baseline. See also *Unified Change Management.*

bookmark:

A marker inserted at a specific point in a document to which the user may return for later reference. In Microsoft Word, a bookmark is a location or selection of text named for reference purposes. Bookmarks are implemented in RequisitePro documents to designate requirement text. Bookmarks are designated by square brackets, as in the following: [example].

change description (version information):

Textual information that specifies the justification or reasoning behind the changes associated with a given revision of a project, document, or requirement. See also *version information*.

change-managed relationship:

A connection between two requirements that implies dependency or another type of relationship. RequisitePro tracks two types of change-managed relationships: *hierarchical relationships* and *traceability relationships*. Changes to either of the requirements can cause a ìsuspectî condition in the relationship. See also *suspect relationship state*.

child requirement:

A requirement that participates in a hierarchical relationship with a parent requirement. A child can have only one parent requirement. See also *parent requirement, hierarchical requirement*.

circular traceability relationship:

A relationship between a requirement and itself, or an indirect relationship that leads back to a previously traced-from node. Traceability relationships cannot have circular references. RequisitePro checks for circular references each time you establish a traceability relationship. RequisitePro also checks for circularity when you create an external relationship. The circular check references only internal requirements and first-level external requirements.

ClearCase:

See *Rational ClearCase*.

ClearQuest:

See *Rational ClearQuest*.

collapse indicator:

See *expand/collapse indicator*.

cross-project traceability:

A RequisitePro feature that allows you to establish traceability relationships between requirements that reside in different projects. See also *external traceability*.

CSV file:

A comma-separated value text file, commonly used to exchange files between database systems that use different formats. Sometimes called *comma-delimited files*, CSV files can be imported into RequisitePro.

.def file:

A file, located in the RequisitePro outlines directory, that specifies information about a particular document outline used by RequisitePro. This type of file has the following format: outline_logical_name is the name of the outline (up to 64 bytes); outline_description is a description of the outline (up to 256 bytes); and outline_filename.dot is the name of the corresponding Microsoft Word template file. See *document template*.

deleted users group:

A security group of users who have been removed from other groups, have no permissions, and cannot log on to RequisitePro. This group is maintained by the project administrator for project history.

dialog box:

An interactive window displayed on the screen that solicits a response from the user.

dialog box shortcut:

See *keyboard shortcuts*.

directory: A catalog for file names and other directories. A directory is an organization of files (or folders) grouped for the userís convenience. The topmost directory is called the *root directory*. The directories within a directory are called *subdirectories*.

discussions: A RequisitePro feature that lets you address comments, issues, and questions to a group of discussion participants. Discussions can be associated with one or more specific requirements or refer to the project in general.

discussion item:

A discussion item is either the initial discussion topic or a response. A response can be either to the initial discussion text or to another response.

.doc file:

A DOS file name extension for Microsoft Word documents. Do not use the .doc file name extension as a file name extension for document types.

document:

Any Microsoft Word document. It may be part of the RequisitePro project or outside the project, and it may or may not contain requirements. See also *requirements document*.

document outline: A reference document or template used to create new documents in RequisitePro. An outline can include formats, page layout information, fonts, and Word styles. It is useful for maintaining consistency across documents of the same type. Document outline information is stored in a .def file. It includes the full name of the template, a description of the outline, and the associated Word template's DOS filename (.dot). See also *document type.*

document properties:

Specific items of information about a document. Documents have attributes associated with them such as author, date and time of last revision, and file name.

document type:

Defines descriptive and operational information associated with a requirements document. The document type serves as a template for each requirements document you create of that type. The document type defines the documentís file extension, such as .prd; all documents of the same document type share the same file extension. The document type identifies a default requirement type; each new requirement you create in the document will be based on the default requirement type unless you indicate otherwise. The document type also identifies a document template which controls the document page layout, default text, paragraph styles, and other structure.

document version:

A revision of a document, identified in the document properties revision history. A new version is created when a document is changed and saved. See *revision.*

.dot file (Word template):

A DOS file name extension for Microsoft Word template files. A Word template determines the basic structure for a document, including fonts, menus, special formatting, and styles. A Word template is associated with a document template, which can be used to create Word documents in RequisitePro. Document outline information is stored in .def files. You can edit the .dot files associated with your document types to include your company name and other information specific to your organization. You can enter the company name and other information in the Microsoft Word **File > Properties** dialog box. Then in the document itself, right-click the Company-Name field, and select **Update Field** (Word template).

enhancement request:

A type of stakeholder request that specifies a new feature or functionality of the system. Certain Rational RequisitePro requirement types have attributes that associate the requirement with specific records in Rational ClearQuest. By default, if you are using a project based on the Use-Case project template, the feature (FEAT) requirement type includes EnhancementRequest and Defect attributes that can associate the feature requirement with enhancement request and defect records in ClearQuest.

enterprise database:

A database that is designed to support a large, networked organization. An enterprise database offers scalability, security, and administrative features for large, complex projects. RequisitePro supports the use of the enterprise databases Oracle and SQL Server for RequisitePro projects

entry-type attributes:

Entry-type attributes are numeric, text, time, or date values that are typed in by project users. An entry-type attribute can be configured with a default value, which users can accept or replace as they create or modify requirements. For example, if you want to set a default cost of $20, format the entry-type attribute ìCostî to accept integer values and have a default value of 20. You define attributes in the **Attributes** tab of the Project Properties dialog box. Requirement attributes can be entry type or list type. If you have configured an integration with Rational ClearQuest, you have two additional default requirement types, EnhancementRequest and Defect. See also *attribute types, attribute values.* Compare *list-type attributes.*

exclusive access:

A condition by which access to a RequisitePro project and its documents is only available to the user who opens the project. Exclusive access is required to change a variety of project characteristics. You set access when you open a project. See also *concurrency.*

expand/collapse indicator:

A small box displayed in one of the following locations: next to a requirement in a view, indicating that the requirement has one or more child requirements; or next to a discussion item in the Discussions dialog box, indicating that the discussion item has responses. The box contains a + or a -, indicating whether or not the requirement or discussion is displayed in an expanded or collapsed state. An item that can be expanded (that is, preceded by a +) contains hidden subitems that can be displayed. An item that can be collapsed (that is, preceded by a -) does not have hidden subitems.

Explorer:

RequisiteProís primary navigation window. In this window, project artifacts (documents, requirements, and views) are displayed in packages in a tree browser.

export:

To move information from one system or program to another. You can export requirements from a RequisitePro view to Microsoft Word documents and to CSV files.

extension:

See *file name extension*.

external project:

A project containing requirements used to establish traceability relationships with requirements in a currently open project. See *cross-project traceability*.

external requirement:

A requirement residing in an external project. You can create traceability relationships between requirements in different projects.

external traceability:

Traceability relationships between requirements in two RequisitePro projects. See also *traceability*.

file name extension:

A DOS file name extension (for example, .prd). File name extensions are used in RequisitePro to indicate document type and can contain a maximum of 20 characters.

filtering:

A process by which you can change the amount of information displayed in a view. You specify certain criteria on which to filter information. You can filter requirements by specifying certain distinct criteria for any or all of their attributes. For example, instead of displaying all requirements, you might apply a filter to display only requirements having a high priority. See also *sorting* and *query*.

hidden text:

A Microsoft Word feature used to hide text within a document. Hidden text is delineated with a dotted underline. A requirement within a Word document begins with a requirement tag formatted in hidden text style. You can hide the tags for improved readability. Click Tools, then Options, and then click the **View** tab. Clear the **Hidden text** check box and click **OK**.

hierarchical requirement:

A requirement that is included within an ordered relationship with other requirements of the same type. Use hierarchical relationships to subdivide a general requirement into more explicit requirements. Child requirements provide additional detail for their parent requirement. For example:
Parent requirement: The system shall display customer information.
Child requirements provided for detail: Name, Address, Date of Birth.
Each child requirement can only have one parent, but a requirement can be both a parent and a child. If a parent requirement is changed, the relationships with its children become suspect.

import:

To bring information from one system or program into another. In RequisitePro, you can import documents and requirements from Microsoft Word files and from CSV files.

Integrated Use Case Management:

An integration between Rational RequisitePro and Rational Rose that allows you to associate use cases in Rose with RequisitePro documents and requirements. You can easily navigate between Rose use-case models and RequisitePro use-case documents and requirements.

internationalization:

The process of developing a program core whose feature and code designs don't make assumptions based on a single language or locale and whose source code base simplifies the creation of different language editions of a program.

keyboard shortcuts:

A *dialog box shortcut* is a keyboard combination you can use to move to tabs, fields, and buttons in a RequisitePro dialog box. To use a dialog box shortcut, press the Alt key while simultaneously pressing the underlined letter in a dialog tab, field, or button. For example, in the Project Properties, Document Properties, or Requirement Properties dialog boxes, press Alt-G to open the **General** tab.
A *menu shortcut* is a keyboard combination you can use to open a menu. To use a menu shortcut, press the Alt key while simultaneously pressing the underlined letter in the menu. For example, to open the Requirement menu, press Alt-R. When a menu is displayed, you can select a command by pressing the key for the underlined letter in the command.

keyword:

Words that are used to help specify and define requirements. Many organizations use words such as ìshall,î ìmust,î or ìwillî in their requirements, making it easier to separate a requirement from surrounding text in the document. Keywords are a part of the definition of a requirement type. You enter keywords in the **Requirement Must Contain** field in the Requirement Type dialog box.

label (version information):

A string (no more than 20 characters long) that contains user-defined revision information. You can use a revision label for noting requirement, document, or project/archive version information.

licensed user:

A RequisitePro user. Each user who is associated with a license of RequisitePro is a licensed RequisitePro user. Also referred to as *user*.

list-type attributes:

Requirement attributes are either list-type or entry-type. List-type attributes are sets of descriptive values (for example, Status, which may contain the list values Proposed, Approved, In progress, and Complete). You can assign single values or multiple values for list-type attributes. You define attribute values in the Attributes tab in the Project Properties dialog box. See also *attribute types*, *attribute values.* Compare *entry-type attributes.*

matrix:

An arrangement of rows and columns used for organizing related items in a view. See *Attribute Matrix, Traceability Matrix.*

menu file:

A text file with specific syntax that is used to add menu commands to any RequisitePro menu. Menu files typically have an MNU or TXT file extension. Menu files are referenced using the Add-ins command on the Tools menu. The commands appear on the RequisitePro menus in the order in which they are specified in the menu file.

menu shortcuts:

See *keyboard shortcuts*.

Metrics:

See *Requirement Metrics*.

multiple select actions:

Mouse actions and combination keyboard/mouse actions that you use to select multiple items in RequisitePro. Use *Ctrl-Click* to select multiple items that are not adjacent to each other. Hold down the Ctrl key and click the mouse. (You can also use Ctrl-click to clear a selected item.) Use *Shift-click* to select a range of items. Hold down the Shift key and click the mouse. Use *Drag* to select a range of items. With one item selected, drag the mouse. Release the mouse when the range is selected.

multiple value list:

An attribute type that supports one-to-many associations. You can use a multiple value list when you want to assign a requirement more than one value for a specific attribute.
For example, you might create an attribute type called Owner, with multiple values Bob, John, and Sue, representing different users working on the same project. For this attribute, you might then assign only the values Bob and Sue to a particular requirement, indicating that Bob and Sue are responsible for this particular requirement, but John is not.

name:

See *requirement name*.

offline author:

A user who takes a RequisitePro document offline.

offline authoring:

A RequisitePro feature that lets users read and modify a document in Microsoft Word outside of RequisitePro (i.e., offline). When you take a document offline, RequisitePro creates a copy of the document in the directory you specify. The original is still stored in RequisitePro, but it is changed to a read-only document. Other users can view the document in RequisitePro, but they cannot edit it until you bring it back online.

OLE:

Object linking and embedding. You can create an OLE object in one application (original application) and insert it into files in other (source) applications. If you modify the OLE object in its original application, the changes are automatically reflected in any files containing the OLE object in a source application. You can use OLE objects in RequisitePro requirements documents.

outline:

See *document outline*.

package:

A container, represented in the Explorer as a folder, that can contain requirements, documents, views, and other packages. You can place related artifacts in a single package for organizing your project. All project packages are shared by all project users.

parent requirement:

A requirement that participates in a hierarchical relationship with one or more child requirements. A parent can reside at the root level, or it can be nested in the hierarchy with a parent of its own. In either case, parents reside above their children in the hierarchy. See also *child requirement, hierarchical requirement*.

parse:

To break information into component parts so that a program can act upon it. Used by the Import Wizard to separate requirements in a document.

peer requirement:

A requirement that is at the same hierarchical level as another requirement. Two requirements are peer requirements when they are children of the same parent. All requirements at the root level are peer requirements of one another. See also *parent requirement, child requirement, hierarchical requirement*.

pending requirement:

A newly created requirement in a document or a view that has not yet been saved in the database.

pending tag number:

A tag number for a newly created requirement in a RequisitePro document. The requirement maintains the pending tag number until the document is saved. See also *Requirement tag.*

permissions:

The RequisitePro privileges granted to a group of RequisitePro users. RequisitePro administrators or members of a group with project security permissions can assign permissions to groups. Also referred to as *privileges* or *rights*. See also *security.*

project:

A RequisitePro project. A project includes a database, documents, document types, requirements and their attributes, requirement types, requirement traceability, discussions, and user and group security. Within RequisitePro, you create a project first. You then create requirement documents and requirements in each document.

project database:

The requirements database managed by RequisitePro. The project database stores all requirements in the project, including those created in requirement documents. Changes you make to requirements in the documents are reflected in the project database. You can create requirements directly in the project database rather than in a requirement document using an Attribute Matrix view. These requirements can be modified or deleted in a view. Microsoft Access, Oracle, and Microsoft SQL Server databases are available for use in RequisitePro projects. See also *database.*

project directory:

A directory that contains information for a single RequisitePro project. You define the project directory when you create a new project. (By default, RequisitePro creates a directory name that matches what is typed in the Name text box in the RequisitePro Project Properties dialog box.) For projects that use the Microsoft Access database, the database is stored in the project directory. The project directory does not store the database for projects that use an enterprise database (i.e., Oracle and SQL Server databases). Requirements documents for the project may be stored in another directory. The project directory path can contain up to 256 characters. See also *directory.*

project prefix:

An identifier used for requirements in cross-project traceability. In an external project, the prefix is displayed before the tag requirement, separated from the tag by a period. For example, if a requirementís tag is PR234 and the projectís prefix is GLOBAL, then the external requirement is displayed as GLOBAL.PR234. The prefix can be up to 8 characters and must be uppercase letters. You can change the prefix at any time in the Project Properties dialog box, **General** tab.

project summary report:

A report that displays the projectís entire structure in report form. The requirement types (and their attributes, including multiple value lists), document types, and document details are included in the printout.

project template:

A RequisitePro template that contains project structure including document types, requirement types, requirement attributes, users, groups, and security.

query:

A method for filtering and sorting requirements in views by limiting the values of one or more attributes or by limiting traceability, and specifying the order in which the filtered requirements are displayed. For example, you might create a query such that only requirements having a high or medium priority are displayed. Within that filter, you might sort requirements first by priority, and then by cost.

Rational Administrator

A Rational tool that enables integrations between Rational products. Use Rational Administrator to configure Rational projects, which store software testing and development information.

Rational ClearCase:

Rational's configuration management tool, which can be used to store copies of RequisitePro projects. Implementing a Unified Change Management model, ClearCase can also be used to create baselines of RequisitePro projects.

Rational ClearQuest:

Rationalís change-request management software. ClearQuest offers a flexible approach to track and manage all the change activities that are associated with an organization, such as defects and enhancement requests. In RequisitePro, the terms *ClearQuest* and *Rational ClearQuest* are used interchangeably.

Rational Rose:

Rationalís visual modeling tool for designing and visualizing object-oriented and component-based applications. RequisitePro is integrated with Rose through Integrated Use Case Management.

Rational TestManager:

A Rational tool designed to help software development and testing professionals track software testing information through all phases of the software development, test, and revision cycles. RequisitePro requirements can be referenced from TestManager to ensure complete test coverage for all requirements. If a requirement changes, TestManager marks the associated test cases and test input as suspect. Using TestManager, you can create reports that show all suspect test cases and test input development coverage reports that show the percentage of requirements that have implemented test cases. If you have installed TestManager, we recommend that you develop all test artifacts in that tool.

Rational Unified Process (RUP):

Rational's online knowledge base of proven software-development principles. RUP captures many of the best practices in modern software development in a form that can be tailored to a wide range of projects and organizations. RUP includes Tool Mentors, which provide descriptions on how Rational Software tools can be used to support particular steps and activities.

read-only access:

An option by which a RequisitePro project or document is restricted to viewing only.

renumbering:

The process of assigning new numbers to requirements. The renumber feature is useful for eliminating ìholesî in a numbering scheme, which can result when you delete requirements. To renumber requirements, you must open the project in exclusive mode. In addition, if security is has been set, you must have project structure permission to renumber requirements. To renumber requirements in documents, all documents that contain requirements of that type must be open.

requirement:

Describes a condition or capability that a system must provide; it is either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document.

requirement attribute:

A descriptive field of information associated with a requirement. RequisitePro requirement attributes are either system attributes (defined by RequisitePro) or user-defined (defined by the project owner). See *attribute*.

requirement attribute label:

The name of the requirement attribute, such as risk, priority, or author. See *attribute label*.

requirement Attribute Matrix:

See *Attribute Matrix*.

requirement attribute type:

See *attribute type*.

requirement attribute value:

See *attribute value*.

requirement location:

The location where a requirement was created or the location to which it was last moved. The location can be either a specific requirements document or the project database.

Requirement Metrics:

A Rational RequisitePro feature for compiling statistics on requirement name, text, attributes, relationships, and revisions. These report results are displayed in Microsoft Excel and can be manipulated using Excel's charting capabilities. Two types of reports are available: static reports (which show results about the project at the present time) and trend analysis reports (which use time-sensitive filters that analyze changes in requirement text, attributes, traceability, and hierarchical relationships).

requirement name:

A user-defined title for a requirement. Like tag and text attributes, the name can be used to reference requirements. The name is displayed by default in all Rational RequisitePro views. All requirements must have either a name, text, or both, but requirements located in documents must have text. A name does not need to be unique and can be changed at any time if you created the requirement or belong to a group that has update permissions.

requirement tag:

The unique identifier of each requirement in a project. A requirement tag is composed of a tag prefix and a unique numerical value, such as PR100.1.2. The tag prefix is always the requirement type, as defined in the Project Properties dialog box. The numerical value is generated by RequisitePro. The tag prefix can contain a maximum of 20 characters. Each requirement type in your project has a different tag prefix (for example, FEAT, SR, UCS) that you determine when you create a new project or add a new requirement type. When you add a requirement to a requirements document or database, RequisitePro assigns the requirement the appropriate tag prefix, dependent on its requirement type, and an incremental numerical value. See also *leaf tag, pending tag number*.

requirement text:

The full textual content of a requirement. In a document, it may include embedded and linked objects, such as graphics, tables, and Microsoft Word files.

requirement type:

A set of descriptive and operational information associated with a requirement when the requirement is created. A requirement type serves as an outline for all requirements of the same type and is useful for classifying and grouping similar requirements in a project. Each requirement type has its own set of user-defined attributes. See also *requirement attribute*.

requirement version:

A requirement attribute that shows the version of the requirement. As a requirement evolves, it is helpful to identify the version numbers (and history) of requirements changes.

requirements document:

A document created in Microsoft Word or RequisitePro that captures requirements and is used to communicate product development efforts. Each requirements document addresses a particular requirement type, such as product features, use cases, and supplementary specifications. A RequisitePro requirements document differs from a Word document in that you can access requirement attributes and other information directly from within the requirements document. Also referred to as *document* or *RequisitePro document*.

requirements management:

A systematic approach to eliciting, organizing, documenting, and managing the changing requirements of a software application.

requirements text:

The full textutal content of a requirement. In a document, it may include embedded and linked objects, such as graphics, tables, and Microsoft Word files.

RequisitePro document:

A Microsoft Word document that is part of a RequisitePro project. Most Word functionality is available within RequisitePro documents.

RequisiteWeb:

Rational's Web interface that allows clients to access RequisitePro requirements information across the Internet or an intranet. By using browsers -- Netscape Navigator or Microsoft

Internet Explorer -- RequisiteWeb provides a thin client solution to access project documents and data. No Requisite application-specific files need to be installed on individual machines.

revision:

A distinct version of a project, document, or requirement. A revision is identified by a unique internal revision number, generated by RequisitePro.

revision number:

The number identifying a revision. Revision numbers are automatically incremented each time you or another user changes and saves a project, document, or requirement.

root requirement:

A requirement at the uppermost level of the requirements hierarchy. A root requirement does not have a parent requirement. See also *parent requirement, child requirement, hierarchical requirement*.

Rose:

See *Rational Rose*.

.rqs file:

A text file, created and updated automatically by RequisitePro, that stores project-specific information. Every RequisitePro project has an .rqs file, which resides in the project directory. Do not modify any section of this file, except the [MSProjectMappings] section, as described in RequisitePro Online Help.

RUP:

See *Rational Unified Process*.

security:

Prevention of system use, potential harm, or data loss by unauthorized individuals. In RequisitePro, users are given specific permissions that determine the level of access they have to projects. If you are a RequisitePro administrator or a member of a group with project security permissions, you can assign permissions. See also *permissions.*

Select:

See *multiple select actions*.

SHIFT-click:

See *multiple select actions.*

SoDA:

Rational's Software Documentation Automation tool.

sorting:

The process of establishing a specific order for the information in a RequisitePro view. You can sort requirements using one or many requirement attributes. For example, you might sort requirements first by priority (high, medium, or low). Then, within priority, you might sub-sort by unique ID number. See also *filtering, query.*

.~sr file:

A text file created for each user who accesses a project. When a project is opened, a file will be written to the project directory with a unique system-generated file name and an extension of *.~sr. This file simply records the user ID, the type of access the user has to the project, and a session ID. The exclusive lock file name will always be exclusive.~sr. All other file names will begin with a prefix that indicates the type of project access reflected by the user in the project: RO indicates read only access, and RW indicates read/write access.

suspect relationship state:

A state applied to a traceability or hierarchical relationship when a change occurs to one or both of the requirements in the relationship. A suspect relationship state indicates that, because of the modification to one or both requirements, the relationship may require modification as well. When suspect relationships are displayed in an Attribute Matrix, an **(s)** appears after the requirement tag in the Traced-to or Traced-from column. See also *change-managed relationship, Traceability Matrix*.

system analyst:

An individual who leads and coordinates requirements elicitation and use-case modeling by outlining the system's functionality and delimiting the system.

tag:

See requirement tag.

tag prefix:

The prefix for a requirement type. A tag is a unique identifier assigned to each requirement you create. The tag prefix can be up to 20 characters long, and it is defined as a part of the requirement type.

template:

See *.dot file (Word template)* or *project template*.

TestManager:

See Rational TestManager.

text:

See *requirement text*.

trace to/trace from relationship:

A relationship between two requirements that implies the source, derivation, or dependencies between the requirements. The trace to/trace from state appears in a Traceability Matrix or Traceability Tree when you create a relationship between two requirements.
For example, you might create a feature requirement (Requirement A) that calls for a new command to be added to a particular menu in your application. Creating this requirement might cause you to create a software requirement (Requirement B) associated with that menu item. You would thus create a traceability relationship in which the software requirement (B) is traced from the feature requirement (A). There can be only one traceability relationship between any two requirements. The difference between calling it a ìtrace toî or ìtrace fromî relationship is one of perspective. For the example above, both of the following statements are true: A is traced to B, and B is traced from A. In RequisitePro, the terms *trace to/trace from relationship* and *traceability relationship* are used interchangeably.

traceability:

A relationship between two requirements that implies the source, derivation, or dependencies between the requirements using the trace to and ìtrace from features.

Traceability Matrix:

A view that illustrates the relationships between requirements of the same or different types. You use this matrix to create, modify, and delete traceability relationships and view indirect relationships and traceability relationships with a suspect state. You can also use the Traceability Matrix to filter and sort the row requirements and column requirements separately. See also *view*.

traceability relationship:

See *trace to/trace from relationship*.

Traceability Tree:

A view that displays all internal and external requirements traced to or from a requirement (depending on the direction of the tree). The Traceability Tree displays only the first level cross-project traceability. For example, if an internal requirement is traced to an external requirement, the external requirement is displayed in the Traceability Tree, but other requirements that the external requirement is traced to are not displayed in the tree. Also referred to as a *tree view*. See also *view*.

tree view:

See *Traceability Tree.*

type:

See *requirement type.*

Unified Change Management (UCM):

Rational's out-of-the-box usage model for managing change in software system development, from requirements to release. UCM is a unified approach to configuration management and change request management, automated in Rational ClearCase and Rational ClearQuest. ClearCase can be set up to implement the UCM model for basic activity-based configuration management or in conjunction with ClearQuest to provide full change-request management. To implement the UCM model when creating a baseline of a Rational RequisitePro project, the project must be associated with a UCM-enabled Rational project in the Rational Administrator.

Universal Naming Convention (UNC):

The system of naming files among computers on a network so that a file on a given computer will have the same path when accessed from any of the other computers on the network. For example, if the directory c:\path1\path2\...pathn on computer servern is shared under the name pathdirs, a user on another computer would open \\servern\pathdirs\filename.ext to access the file c:\path1\path2\...pathn\filename.ext on servern. Also called *Uniform Naming Convention*.

use case:

A sequence of actions that a system performs, yielding an observable result of value (a work output) to a particular actor (someone or something outside the system that interacts with the system).

user information:

A RequisitePro users user name, password, and e-mail address. If you are a member of a group with project security permissions, you can edit another userís information. In addition, a user can edit his or her own user information.

users group:

A security group that can, by default, read documents and requirements, create views, and participate in discussions. Administrators can modify Users group permissions.

version information:

Unique, identifying information associated with a project, document, or requirement each time it is modified. The internally generated revision number, combined with information about the author, date, time, and reason for modification, constitute version information. See also *revision.*

view:

A window that displays requirements, the attributes assigned to requirements, and the relationships between requirements. A view displays information in spreadsheet-like tables or in outline trees. In RequisitePro, the terms *view* and *view window* are used interchangeably. See also *Attribute Matrix, Traceability Matrix, Traceability Tree.*

wildcard character:

A keyboard character that can be used to represent one or many characters. Typically the asterisk (*) represents one or more characters and the underscore (_) represents a single character. For example, the file specification *.* (pronounced ìstar-dot-starî) uses the asterisk wildcard to mean ìany combination of file name and extension.î When filtering the subject of discussions, the percent symbol (%) is the wildcard character used to represent one or more characters.

Word document:

A document created in Microsoft Word or RequisitePro that is saved in Word format. See also *RequisitePro document.*

Word Template:

See .dot file.

RU Financial Services

Requirements Management Plan
RU e-st

Version <0.6>

Revision History

Date	Version	Description	Author
1st September 2001	<0.1>	Initial version	B Baker
5th September 2001	<0.2>	Review changes	B Baker
17th September 2001	<0.3>	Added missing attributes to UC type	B Baker
2nd October	<0.4>	Changed extension for SUP document type	B Baker
12th October	<0.5>	Post Beta changes	B Baker
8th November 2002	<0.6>	Removed Defect and Enhancement Request attributes for SUPL requirement type	B. Baker

Table of Contents

1. Introduction 5
   - 1.1 Purpose 5
   - 1.2 Scope 5
   - 1.3 Definitions, Acronyms, and Abbreviations 5
     - 1.3.1 Baseline 5
     - 1.3.2 Business Rule 5
     - 1.3.3 Business Unit Manager 5
     - 1.3.4 Customer 5
     - 1.3.5 Engineering Time 5
     - 1.3.6 NCSS 5
     - 1.3.7 Pareto Chart 5
     - 1.3.8 Product Feature 6
     - 1.3.9 Rational RequisitePro 6
     - 1.3.10 Rational Rose® 6
     - 1.3.11 Rational SoDA® 6
     - 1.3.12 Rational TestManager™ 6
     - 1.3.13 Stakeholder 6
     - 1.3.14 Stakeholder Request 6
     - 1.3.15 Vision Document 6
   - 1.4 References 6
   - 1.5 Overview 7

2. Requirements Management 8
   - 2.1 Organization, Responsibilities, and Interfaces 8
     - **2.1.1 customer** 8
     - **2.1.2 user** 8
     - **2.1.3 stakeholder** 8
     - **2.1.4 project manager** 8
     - **2.1.5 quality assurance (QA)** 8
     - **2.1.6 developer** 8
     - **2.1.7 team leader** 8
     - **2.1.8 configuration manager** 8
     - **2.1.9 requirements specifier** 8
   - 2.2 Contact Table 9
   - 2.3 Tools, Environment, and Infrastructure 9

3. Requirements Artifacts 10
   - 3.1 Artifact Description 10
     - 3.1.1 Document Types 10
     - 3.1.2 Requirement Types 11
     - 3.1.3 Attributes 12
   - 3.2 Traceability 20
     - 3.2.1 Traceability Criteria for Requirement Types 21
   - 3.3 Reports and Measures 21

4. Training and Resources 24

Requirements Management Plan

1. Introduction

1.1 Purpose

The purpose of this plan is to establish and document a systematic approach to eliciting, organizing, and documenting the requirements of the system. This plan also establishes and maintains agreement between the customer and the project team on the changing requirements of the system.

1.2 Scope

This plan provides guidelines for the management of the RU e-st project.

1.3 Definitions, Acronyms, and Abbreviations

The following definitions relate specifically to the Requirements Management plan. For definitions relating to your project refer to your project Glossary.

1.3.1 Baseline

A reviewed and approved release of artifacts that constitutes an agreed basis for further evolution or development and that can be changed only through a formal procedure, such as change management and configuration control.

1.3.2 Business Rule

A formal regulation or bylaw imposed by an organization or simply the standard practices of users governing the way the organization conducts its business. Business rules may be classified as definitions, facts (relationships, connections), constraints ('must have' versus 'must not have') and derivation rules (inferring new facts from existing ones).

1.3.3 Business Unit Manager

A member of an RU e-st business unit. Responsible and accountable for communicating requirements for systems to IT and for accepting delivery of systems.

1.3.4 Customer

The economic buyer of a project developed by IT.

1.3.5 Engineering Time

A measurement unit describing engineering effort. Usually expressed in units of weeks or months. The move away from terms like man-months, or person-months is deliberate. Men and months are interchangeable commodities only when a task can be partitioned among many workers *with no communication among them*. In most uses, engineering time is used to understand the relative size of something, not as an advertised elapsed time to complete a task.

1.3.6 NCSS

Non-Commented Source Statements. A metric used to estimate project risk, estimate schedules, and most importantly, a component of software release decision when used in defect density calculation.

1.3.7 Pareto Chart

A useful tool for graphically depicting where allocating time, human, and financial resources will yield the best results. Dr. Joseph Juran (of total quality management fame) formulated the Pareto Principle after expanding on the work of Wilfredo Pareto, a nineteenth century economist and sociologist. The Pareto Principle states that a small number of causes is responsible for a large percentage of the effect–usually a 20 percent to 80 percent ratio.

1.3.8 Product Feature

A capability or characteristic of a system that directly fulfills a Stakeholder Need. Often thought of as the "advertised benefits" of the system.

1.3.9 Rational RequisitePro

Rational RequisitePro helps teams organize, prioritize, track and control changing requirements of a system or application.

Rational Requisite®Web helps teams organize, prioritize, track and control changing requirements of a system or application via a Web browser interface.

1.3.10 Rational Rose®

Rational Rose® is a graphical component modeling and development tool, using the industry-standard Unified Modeling Language (UML).

1.3.11 Rational SoDA®

Rational SoDA® provides automatic generation of software documentation.

1.3.12 Rational TestManager™

Designed to help you track software testing information through all phases of the software development, test, and revision cycles. You can use TestManager™ to plan testing strategies, and to track information related to test execution.

1.3.13 Stakeholder

A stakeholder is defined as anyone who is materially affected by the outcome of the project. Effectively solving any complex problem involves satisfying the needs of a diverse group of stakeholders. Stakeholders will typically have different perspectives on the problem, and different needs that must be addressed by the solution.

1.3.14 Stakeholder Request

Requests a stakeholder makes for the system to be developed. It may also contain references to external sources to which the system must comply.

1.3.15 Vision Document

A general vision of the core project's requirements. It provides the contractual basis for the more detailed technical requirements. This is a project management document owned by the IT project manager. A system analyst authors it with primary input educed from the Stakeholders.

1.4 References

Kruchten, Philippe. 1999. *The Rational Unified Process*. Menlo Park, CA: Addison Wesley

Leffingwell, D. and Don Widrig. 2000. *Managing Software Requirements*. Menlo Park, CA: Addison Wesley.

Spence, I. and L. Probasco. 1998. *Traceability Strategies for Managing Requirements with Use Cases*. Cupertino, CA: Rational Software Corporation.

Rational Unified Process®, Version 2002.05.00. Copyright © 1987 – 2001. Rational Software Corporation

1.5 Overview

This Requirements Management Plan is being created to address identified problems in the requirements management process experienced previously in software projects delivered by RU Financial Services. Problems included:

- Poor communication of changes to requirements, including the use of e-mail to communicate changes;
- Data changes out of sync with code changes;
- Lack of formal handoffs between team members for software artifacts;
- Minimal contact with stakeholders;
- Users not knowing what they want until they see it;
- Fast pace of requirements change;
- Geographically disbursed team, a problem that has been remedied;
- Lack of clear understanding of roles within the requirements process;
- Separation of the subject matter experts and developers may result in decreased customer satisfaction;
- Inconsistent documentation;
- Inability to easily find requirement documents.

What follows is in response to these problems. A standard set of documents used to express requirements of all levels will be defined. An established set of requirement types to capture stakeholder problems, needed features of the system, software requirements, standard terms, and business rules will be described. For each requirement type a collection of requirements attributes, used to manage delivery of the needed system and manage changes to the requirements over the lifecycle of the project, will be identified along with values and ranges appropriate for each. A model of traces between requirements will be established to help communicate requirements change to all members of the project team. An initial list of predefined views of requirements information will be defined. This list must evolve with use. A set of tool extensions to provide needed functionality specifically for RU e-st will be created. Finally, a list of roles within the requirements management process will be identified.

The premise of this endeavor from the start was that an initial process and tool configuration for requirements management be defined, and that these would evolve through use. This being the case, this document should be considered a “living” document and changes to it over time and experience are expected. Changes to this document should be made in a controlled manner and only after review by stakeholders.

2. Requirements Management

2.1 Organization, Responsibilities, and Interfaces

*2.1.1 **customer***

A person or organization, internal or external to the producing organization, who defines the business needs for the project. In a large system this may not be the end user. The customer is the ultimate recipient of the business benefit obtained by deploying the system.

*2.1.2 **user***

A person who will use the system that is developed.

*2.1.3 **stakeholder***

An individual or organization who is materially affected by the outcome of the system.

*2.1.4 **project manager***

The role with overall responsibility for the project. The Project Manager needs to ensure tasks are scheduled, allocated and completed in accordance with project schedules, budgets, and quality requirements.

*2.1.5 **quality assurance (QA)***

The function of Quality Assurance is the responsibility of (reports to) the Project Manager and is responsible for ensuring that project standards are correctly and verifiably followed by all project staff.

*2.1.6 **developer***

A person responsible for developing the required functionality in accordance with project-adopted standards and procedures. This can include performing activities in any of the *requirements, analysis & design, implementation,* and *test* disciplines.

*2.1.7 **team leader***

The team leader is the interface between project management and developers. The team leader is responsible for ensuring that a task is allocated and monitored to completion. The team leader is responsible for ensuring that development staff follow project standards, and adhere to project schedules.

*2.1.8 **configuration manager***

The configuration manager is responsible for setting up the product structure in the Change Management system, for defining and allocating workspaces for developers, and for integration. The configuration manager also extracts the appropriate status and metrics reports for the project manager.

*2.1.9 **requirements specifier***

The requirements specifier details the specification of a part of the system's functionality by describing the requirements aspect of one or several use cases and other supporting software requirements. The requirements specifier may also be responsible for a use-case package, and maintains the integrity of that package. It is recommended that the requirements specifier responsible for a use-case package is also responsible for its contained use cases and actors.

2.2 Contact Table

The following table identifies contact information for the various roles within the project. Some of these roles, such as user, represent a large number of people. In this case the contact represents all parties taking on that role.

Role	Name	Title	Organization	Contact
Customer	J.R. Slingsby	CFO	RU Financial Services	jslingsby@rucompany.com
User	J. Doe	Customer Service	RU Financial Services	jdoe@rucompany.com
Project manager	P. Murphy	Software Project Manager	RU Financial Services	pmurphy@rucomany.com
Quality assurance	B.V. Tam	Senior Testing Manager	RU Financial Services	btam@rucompany.com
Team leader	H. Moriyuke	Senior Developer	RU Financial Services	hmoriyuke@rucompany.com
Requirements specifier	P. Murphy	Software Project Manager	RU Financial Services	pmurphy@rucompany.com
Administrator	M. Mutevelic	IT director	RU Financial Services	mmutevelic@rucompany.com
Configuration manager	K. Zahar	SeniorSoftware Engineer	RU Financial Services	kzahar@rucompany.com

2.3 Tools, Environment, and Infrastructure

Tool	Description	Technical Support	Website
Rational RequisitePro	For managing requirements.	support@rational.com	www.rational.com

3. Requirements Artifacts

3.1 Artifact Description

3.1.1 Document Types

Document Type	Description	Default Requirement Type
Stakeholder Requests	Key requests from stakeholders. The file name extension will be **STR**. Outline: RUP Stakeholder Requests	Stakeholder Request (STRQ)
Vision	Conditions or capabilities of this release of the system. The file name extension will be **VIS**. Outline: RUP Vision Document	Feature (FEAT)
Use-Case Specification	Use-case description and elaboration. The file name extension will be **UCS**. Outline: None	Use Case (UC)
Supplementary Specification	The Supplementary Specifications capture system requirements not readily captured in use cases. Such requirements include: legal and regulatory requirements; quality requirements such as usability, reliability, performance and supportability; and design constraints. The file name extension shall be: **SUP**. Outline: RUP: Supplementary Specification	Supplementary Requirement (SUPL)
Glossary	Used to capture common vocabulary. The file name extension will be **GLS**. Outline: RUP Glossary	Glossary Item (TERM)
Requirements Management Plan	This document type describes requirements and strategies specific to the management and development of the Requirements Management Plan. The file name extension will be **RMP**. Outline: RUP Requirements Management Plan	Requirements Management Plan (RMP)

3.1.2 Requirement Types

Requirement Type	Description	Attributes
Stakeholder Request (STRQ)	A stakeholder request defines a business need of one of the stakeholders.	Priority, Status, Cost, Difficulty, Stability, Assigned to
Feature (FEAT)	A capability provided by the system that directly fulfills a stakeholder need.	Priority, Status, Planned Iteration, Actual Iteration, Difficulty, Stability, Assigned to, Origin, Rationale, Cost, EnhancementRequest, Defect
Use Case (UC)	A use case requirement is a functional requirement that defines what the system must do at a specific step in the use of the system.	Property, Affects Architecture, Planned Iteration, Actual Iteration, Assigned to, Rank, Test, Priority, Status, Difficulty, Stability, Cost, EnhancementRequest, Defect
Glossary Item (TERM)	Glossary terms are words that have meanings specific to the project that cannot be found in the dictionary.	None
Supplementary Requirement (SUPL)	Defines non-functional and functional requirements that are not appropriate to locate in a Use Case.	Priority, Status, Cost, Difficulty, Stability, Assigned To, Affects Architecture, Risk, EnhancementRequest, Defect
RM Plan (RMP)	Requirements Management Plan requirements are used to track anything significant in the requirements management plan.	None

3.1.3 Attributes

3.1.3.1 Requirement Attributes for Stakeholder Request (STRQ)

Requirement text consists of one or more phrases that describe the request a stakeholder makes for the system to be developed. It may also contain references to external sources to which the system must comply.

Priority

Set by the Business System Manager. Ranking requirements by their relative benefit to the business opens a dialogue with customers, analysts, and members of the development team. Used in managing scope and determining development priority.

High	Essential. Failure to implement means the system will not meet customer needs. Must be implemented in the release.
Medium (default)	Important to the effectiveness and efficiency of the system for most applications. Lack of inclusion may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of this item.
Low	Useful in less typical applications, will be used less frequently, or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.

Table 3-1 Priority attribute values for STRQ requirement type.

Status

Set after negotiation and review by the project management team and Business Unit Managers. Tracks progress during definition of the project baseline. Used to control scope.

Proposed	Under discussion, but has not yet been reviewed and accepted by a working group consisting minimally of Business Unit Managers and IT Project Management.
Approved (default)	Deemed useful and feasible and has been approved for implementation by the working group.
Incorporated	Incorporated into the product baseline at a specific point in time.
Validated	Implemented and validated.

Table 3-2 Status attribute values for STRQ requirement type.

Difficulty

Set by the Development Team, it indicates the estimated effort required for implementing and validating the requirement. Used in managing scope and determining development priority.

High	Above average effort to implement the request.
Medium (default)	Average effort to implement the request.
Low	Below average effort to implement the request.

Table 3-3 Difficulty attribute values for STRQ requirement type.

Stability

Set by Business Unit Manager and the System Analyst based on the probability that the behavior expressed by the requirement will change or the team's understanding of that behavior will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.

High	It is very unlikely that this behavior will change, or that the development team's understanding of the behavior will change.
Medium (default)	No indicator exists to predict the likelihood of change for this behavior. Default value.
Low	It is very likely that this behavior will change, or that the development team's understanding of the behavior will change.

Table 3-4 Stability attribute values for STRQ requirement type.

Assigned to

Text giving the analyst's name who has been assigned to fully describe this item. Set by the System Analyst. The name is given in the format first-name last-name – example: Sue Collins.

Cost

Numeric value set by Development Team based on the estimated cost to fully develop this item. Used in managing scope and determining development priority.

3.1.3.2 Requirement Attributes for Feature (FEAT)

Requirement Text is the feature description.

Refer to section 3.1.3.1 – Requirement Attributes for Stakeholder Request (STRQ) for a description of the attribute values for Priority, Status, Difficulty and Stability.

Priority

Set by the Business System Manager. Ranking requirements by their relative benefit to the business opens a dialogue with customers, analysts and members of the development team. Used in managing scope and determining development priority.

Status

Set after negotiation and review by the project management team and Business Unit Managers. Tracks progress during definition of the project baseline. Used to control scope.

Planned Iteration

Text. The iteration when this requirement is planned to begin to be implemented. Integer value.

Actual Iteration

Text. The iteration when this requirement actually began to be implemented. Integer value.

Difficulty

Set by the Development Team, it indicates the estimated effort required for implementing and validating the requirement. Because some requirements require more time and resources than others, estimating the engineering time is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.

Difficulty estimate is based on estimates of many factors, including effort, size, and coordination complexity.

When estimating effort keep in mind all of the activities associated with the production of a software product. Rule of thumb: Projects created primarily from reused software take about one fourth the time and resources of those that are new. Measurement will be integer value, unit of engineering weeks.

Size refers to the estimated number of non-commented source statements needed to implement the requirement. The greater the number of lines of code the greater the complexity of the project. Reused software lines of code should be counted at a quarter of their number. Measurement will be an integer value; number of non-commented source statements/1000(KNCSS).

Coordination complexity is estimated by analyst and development team based on the reliance on organizations outside their control needed to implement the requirement. Measurement will be qualitative: high, medium or low.

Stability

Set by the Business System Manager and development. Used to help establish development priorities and determine the items for which additional exploration and discovery is the appropriate next action.

Risk

Set by the development team based on technology risk, development risk, and other risk factors. Normally performed at the project level, the Technology Risk Assessment (TRA), Appendix 1, for the requirement allows the development team to understand the slope they must climb to deliver the goods. Used to help establish development risk. Measurement will be an integer value between zero and one hundred-fifty. A value of zero indicates no assessment has been performed.

Risk can be measured and calculated in a variety of ways. For example, each risk component requirement attribute (i.e. effort, size, coordination complexity, technology risk, architectural impact) is evaluated for its value in the matrix, and the resulting value is multiplied by the weight and totaled. The result is an integer valued between 0-100. This measures the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation. The result is then recorded as a qualitative category: high, medium, or low.

High	It is very likely that implementing this requirement will result in undesirable events, such as cost overruns, schedule delays or cancellation. The development team is very concerned about completing this requirement, with the given constraints. Reasons for the concern are documented in a separate document.
Medium (default)	No indicator exists to predict the likelihood of undesirable events. The development team feels comfortable implementing this requirement under the constraints. But, there are some unknown elements that need to be overcome. Reasons for these concerns are documented in a separate document. Default value.
Low	It is very unlikely that implementing this requirement will result in undesirable events. The development team is fully confident of being able to implement the requirement within the project's constraints. The team has done similar work before with success.

Table 3-5 Stability attribute values for FEAT requirement type.

Assigned to

Text giving the analyst's name who has been assigned to fully describe this item. Set by the System Analyst. The name is given in the format: first-name last-name – example: Sue Collins.

Origin

List value giving the source of the requirement: hotline, partners, competitors, or large customers. Set by the System Analyst.

Rationale

Text giving the rationale for the requirement. Set by the System Analyst.

Cost

Set by Development Team based on the estimated cost to fully develop this requirement. Used in managing scope and determining development priority.

Enhancement Request

Text. Contains the unique identifier of the Enhancement.

Defect

Text. Contains the unique identifier of the Defect.

3.1.3.3 Requirement Attributes for Use Case (UC)

Requirement text describes what the system should do.

Refer to section 3.1.3.1 – Requirement Attributes for Stakeholder Request (STRQ) for a description of the attribute values for Priority, Status, Difficulty and Stability.

Priority

Set by the Business System Manager. Ranking requirements by their relative benefit to the business opens a dialogue with customers, analysts and members of the development team. Used in managing scope and determining development priority. Values: High, Medium (default), and Low.

Status

Set after negotiation and review by the project management team and Business Unit Managers. Tracks progress during definition of the project baseline. Used to control scope. Values: Proposed (default), Approved, Incorporated, Validated.

Difficulty

Set by the Development Team, it indicates the estimated effort required for implementing and validating the requirement. Because some requirements require more time and resources than others, estimating the engineering time is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.

Difficulty estimate is based on estimates of many factors, including effort, size, and coordination complexity.

When estimating effort keep in mind all of the activities associated with the production of a software product. Rule of thumb: Projects created primarily from reused software take about one fourth the time and resources of those that are new. Measurement will be an integer value; unit of engineering weeks.

Size refers to the estimated number of non-commented source statements needed to implement the requirement. The greater the number of lines of code the greater the complexity of the project. Reused software lines of code, should be counted at a quarter of their number. Measurement will be an integer value; number of non-commented source statements/1000(KNCSS).

Coordination complexity is estimated by analyst and development team based on the reliance on organizations outside their control needed to implement the requirement. Measurement will be qualitative: High, Medium, or Low.

Stability

Set by the Business System Manager and development. Used to help establish development priorities and determine the items for which additional exploration and discovery is the appropriate next action. Values: High, Medium (default), and Low.

Planned Iteration

The iteration when this use case or use-case section is planned to begin to be implemented. Integer value.

Actual Iteration

The iteration when this use case or use-case section actually began to be implemented. Integer value.

Assigned to

Text giving the analyst's name who has been assigned to fully describe this item. Set by the System Analyst. The name is given in the format: first-name last-name – example: Sue Collins.

Property

Set by the System Analyst. Indicates the location within a Use-Case Report where the requirement lives.

Name	The requirement is found in the name of the use-case.
Brief Description	The requirement is found in the brief description of the use case.
Basic Flow	The requirement is found in the basic flow of the use case.
Alternate Flow	The requirement is found in an alternate flow of the use case.
Special Requirements	The requirement is found in the special requirements section of the use case.
Precondition	The requirement is found in the pre-condition section of the use case.
Postcondition	The requirement is found in the post-condition section of the use case.
Relationships	The requirement is found in the post-condition section of the use case.
Extension Points	The requirement is found in the post-condition section of the Use-Case.

Table 3-6 Stability attribute values for UC requirement type.

Affects Architecture

Set by the Software Architect, this indicates the requirement has an influence on the software architecture. Values: True or False (default).

Rank

Set by the development team based on impact on the architecture or its importance for a release. Ranking requirements opens a dialogue with customers, analysts, and members of the development team. Used in determining development priority. Integer.

Test

Set by the Test Designer, it indicates if the use case has been tested. Values: True, False.

Cost

Numeric value set by Development Team based on the estimated cost to fully develop this Requirement. Used in managing scope and determining development priority. Real value.

Enhancement request

Text. Contains the unique identifier of the Enhancement.

Defect

Text. Contains the unique identifier of the Defect

3.1.3.4 Requirement Attributes for Supplementary Requirements (SUPL)

Requirement text describes what the system should do.

Refer to section 3.1.3.1 – Requirement Attributes for Stakeholder Request (STRQ) for a description of the attribute values for Priority, Status, Difficulty, and Stability.

Priority

Set by the Business System Manager. Ranking requirements by their relative benefit to the business opens a dialogue with customers, analysts and members of the development team. Used in managing scope and determining development priority. Values: High, Medium (default), and Low.

Status

Set after negotiation and review by the project management team and Business Unit Managers. Tracks progress during definition of the project baseline. Used to control scope. Values: Proposed (default), Approved, Incorporated, Validated.

Cost

Numeric value set by Development Team based on the estimated cost to fully develop this Requirement. Used in managing scope and determining development priority. Real value.

Difficulty

Set by the development team, it indicates the estimated effort required for implementing and validating the requirement. Used in determining development priority. Values: High, Medium (default), and Low.

Stability

Set by the Business System Manager and development. Used to help establish development priorities and determine the items for which additional exploration and discovery is the appropriate next action. Values: High, Medium (default), and Low.

Assigned to

Text giving the analyst's name that has been assigned to fully describe this item. Set by the System Analyst. The name is given in the format first-name last-name – example: Sue Collins.

Affects Architecture

Set by the Software Architect, this indicates the requirement has an influence on the software architecture. Values: True or False (default).

Risk

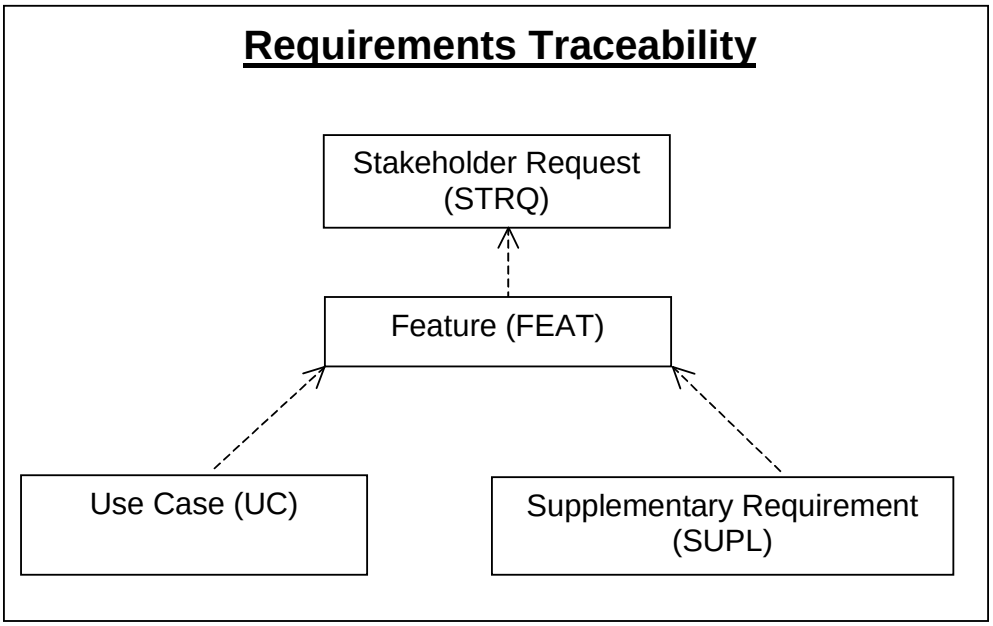
Set by the development team based on technology risk, development risk, and other risk factors. Normally performed at the project level, the Technology Risk Assessment (TRA), Appendix 1, for the requirement allows the development team to understand the slope they must climb to deliver the goods. Used to help establish development risk. Measurement will be an integer value between 0-150. A value of zero indicates no assessment has been performed.

Risk can be measured and calculated in a variety of ways. For example, each risk component requirement attribute (i.e. effort, size, coordination complexity, technology risk, architectural impact) is evaluated for its value in the matrix, and the resulting value is multiplied by the weight and totaled. The result is an integer valued between 0-100. This measures the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation. The result is then recorded as a qualitative category: High, Medium, or Low.

High	It is very likely that implementing this requirement will result in undesirable events, such as cost overruns, schedule delays, or cancellation. The development team is very concerned about completing this requirement, with the given constraints. Reasons for the concern are documented in a separate document.
Medium	No indicator exists to predict the likelihood of undesirable events. The development team feels comfortable implementing this requirement under the constraints. But, there are some unknown elements that need to be overcome. Reasons for these concerns are documented in a separate document. Default value.
Low	It is very unlikely that implementing this requirement will result in undesirable events. The development team is fully confident of being able to implement the requirement within the project's constraints. The team has done similar work before with success.

Table 3-7 Stability attribute values for SUPL requirement type.

3.2 Traceability



3.2.1 Traceability Criteria for Requirement Types

Requirement Type	Guidelines	Notes
Stakeholder Request (STRQ)	Every stakeholder request with "Approved" status must trace from one or more Features.	
Feature (FEAT)	Every feature with "Approved" status or greater must trace to one or more Stakeholder Requests.	
Use Case (UC)	Use Cases trace to one or more Features. Use cases do not trace from any other requirement type.	
Glossary Item (TERM)	Glossary terms do not trace to any other requirement type.	
Supplementary Requirement (SUPL)	Supplementary requirements trace to Feature requirements.	

3.3 Reports and Measures

For reports and measures on this project, use the Requirement Metrics tool, which is available from the Tool menu. In Requirement Metrics, create reports based on requirement types or saved views and query on the following filters:

Attribute Value:

An Attribute Value filter returns the requirements whose attributes have values that match your criteria. The choices you make depend on the data type of the attribute you select in the Attributes drop-down list.

Attribute Value Change:

An Attribute Value Change filter returns the requirements with a changed attribute value that matches your BEFORE and AFTER selections. The choices you make depend on the data type of the attribute you select in the Attributes drop-down list. If several changes have been made to the attribute value, your BEFORE selection must specify the value which immediately preceded the current (AFTER) value. To report any change in an attribute value of the selected requirement type, use the Select All buttons for the BEFORE and AFTER selections.

Base Filter:

The Base filter defines the requirement type for a query. Every query is specific to a single requirement type. When you use a saved RequisitePro view defined in the Views Workplace, the Base filter serves as the first filter for requirements. The Base filter cannot be deleted, and is only changed by selecting a new view from the Choose a Requirement View drop-down list.

Children:

A Children filter returns the requirements that have the number of direct children that matches your selection criteria. You must choose which operator to use and enter at least one numeric value. If you choose the "Between" or "Not Between" operator, you must also enter a second numeric value. The default setting (> 0) reports all requirements of the selected type that have any children.

Parent Change

A Parent Change filter returns the requirements whose parent relationship has changed from your BEFORE selection to your AFTER selection. The selections allow you to report requirements that were changed from or to any parent, no parent, or one or more selected parents which you choose. When reporting changes to selected requirements, if a requirement had several changes of parent assignments, your BEFORE selection must specify the value which immediately preceded the current (AFTER) value.

Requirement Creation:

The Requirement Creation filter returns all requirements of the specified requirement type. Typically, this filter is used with the Time Period option to determine which requirements have been created in a specified time period.

Requirement Text Change:

A Requirement Text Change filter returns the requirements whose text has changed the number of times you specify. The filter allows you to choose comparison operators, such as "equal to" (=), "greater than" (>), etc., when specifying the number of times that the text has changed.

Traceability Change:

A Traceability Change filter returns the requirements that had a trace relationship either removed or added, depending on your selection.

View Descriptions:

Query Name	Description	Requirement Type	Attributes	Attribute Value Range
Features	displays all requirements of the Feature Type	FEAT	all	all
Glossary Terms	displays all requirements of the Glossary Term Type	TERM	all	all
Supplementary Requirements	displays all supplementary requirements	SUPL	all	all

Stakeholder Request	displays all requirements of the Stakeholder Request Type	STRQ	all	all
Use Case Survey	displays all requirements of the Use Case Type	UC	all	all

4. Training and Resources

TBD

RU Financial Services

Glossary

Version 1.4

Revision History

Date	Version	Description	Author
Jan 8, 2002	1.0	First draft – will continually update	J. Carlos Goti
Jan 13, 2002	1.1	Additional entries	J. Carlos Goti
Mar 28, 2002	1.2	Relocated Business Rules and Transactions into the Supplementary Specifications doc	J. Carlos Goti
November 15, 2002	1.3	Removed categories from terms and added new terms	J. Bell
November 30, 2002	1.4	Editorial changes	B. Baker

Table of Contents

1. Introduction 5

Purpose 5
Scope 5
References 5
Overview 5

2. Definitions 6

Account 6
Account alias 6
Account information 6
Account funding type 6
Account id 6
Account last transaction date 6
Account operation type 6
Account status 7
Account total 7
Account type 7
Asset 7
Asset name 7
Asset total 7
Asset type 7
Bid/ask price 7
Cash asset 7
Customer id 7
Customer password 8
Current price 8
Daily change price 8
Daily high price 8
Daily low price 8
Dividend 8
Earnings per share 8
Executed per unit amount 8
Executed total amount 8
Fifty-two week range 8
IRA 8
IRS 8
Limit trade order 8
Marital status 8
Market trade order 8
Minimum cash asset 8
News list 9
Number of dependents 9
Number of shares owned 9
Number of shares per transaction 9
Opening price 9
Percentage price change 9
Previous close 9
Price/earnings ratio 9
Quote 9

RU e-st 9
RU e-st terms and conditions 9
Security 9
Securities list 9
Securities trading 10
Social security number 10
Stock symbol 10
Time stamp executed transaction 10
Total current shares traded 10
Trade order type 10
Trade transaction type 10
Trading customer 10
Trading symbol 10
Transaction fees 10
Transaction id 10
Transfer amount 10
Transfer trade order 10
Volume 10

Glossary

1. Introduction

This glossary is part of the set of artifacts for the **RU e-st** project. This document is used to define terminology specific to the securities trading problem domain, explaining terms that may be unfamiliar to the reader of the use-case descriptions or other project documents. This document can also be used as an informal *data dictionary*; it captures data definitions so that use-case descriptions and other project documents can focus on what the system must do with the information.

Purpose

Record the definitions of the principal terms used in the stock/bond trading domain as they relate to the project.

Analysts and engineers involved in the project are expected to make frequent references to this document as they complete their activities.

This document is updated frequently during the progress of the project, when it is necessary to define new terms.

Scope

Typical terms that should be included in this glossary are:

- Domain specific
- System's components
- Systems interacting with RU e-st
- Other

References

Standard stock trading terminology is used whenever possible.

The RU e-st Supplementary Specification document contains Business Rules referenced in the terms of this Glossary.

Overview

Definitions are shown next to the corresponding terms.

Terms are kept in alphabetical sequence.

Since the project does not involve multiple domains, all terms are organized in one single list and "groups of terms" are not used.

2. Definitions

Account

A particular account that a trading customer has established. A minimum of $2500 is required to open an account. A customer may have one or more accounts. Accounts can be non-retirement (Individual or Joint) or retirement (Individual). Accounts have a number of assets contained in them depending on their type. All accounts have a cash asset.

Account alias

Alias that the customer can use to identify his/her account.

•

Account funding type

Accounts can have the following funding options:

- Wire
- Check
- Transfer
- Stock
- Transfer from an existing RU e-st account

Account id

Unique identifier of a customer account.

Account information

An account description includes:

- Account id
- Primary trading customer name, address, phone number, e-mail address, social security number, date of birth, marital status, number of dependents, citizenship, employment status
- (If a joint account) Second trading customer name, address, phone number, e-mail address, social security number, date of birth, marital status, number of dependents, citizenship and employment status
- Account alias
- Account type
- Account operation type
- Account funding type

Account status

Account last transaction date

Date of the last transaction executed in a given account.

Account operation type

Accounts can operate with the following characteristics:

- Cash
- Cash and margin
- Cash and margin and option

Account status

The customer account must be in one of the following states:

- **Probationary**: the customer is not allowed to perform buy or sell transactions until the method of payment has been received and credited to the customer's account, or the value of the cash account complies with the account cash minimum business rule.
- **Operational**: the customer is allowed to perform any transaction
- **Not_in_licensed_US state**: the account is in a U.S. state in which RU e-st has no license to operate. Transactions are not allowed.
- **Other?**

Account total

Total current market value of all assets in an account. It is the sum of the asset totals for all assets included in the account.

Account type

RU e-st supports the following types of accounts:

- **Non-retirement (stocks, mutual funds, and cash assets):**
  - **Individual**: a non-tax-deferred brokerage account has only one owner. Upon the death of the owner, all property goes to his/her estate.
  - **Joint**: a form of joint ownership exclusively for married couples. Joint accounts can be either Joint Tenants with Rights of Survivorship, Tenants in Common, Tenants by the Entirety, or Community Property.
- **Retirement (mutual funds and cash assets)**: an IRA account, always individual. These can be a single contributory IRA, Roth IRA, Rollover IRA, or a SIMPLE plan.

Asset

The value of a mutual fund, a stock, or cash. Assets are contained in accounts.

Asset name

Full name corresponding to an asset – corresponds uniquely to a stock symbol.

Asset total

Total value of an asset in an account. It is computed as the number of shares owned multiplied by the current market value of the corresponding security.

Asset type

Assets contained in an account can be:

- Mutual fund
- Stock
- Cash

Bid/ask price

Per unit value of the last bid and ask prices of a security.

Cash asset

The cash value, in dollars, in an account.

Customer id

Customer id for logging on to RU e-st. A customer has only one customer id, no matter how many accounts he or she has.

Customer password

Customer password. A customer has only one password, no matter how many accounts he or she has.

Current price

Per unit price paid for a given security in the most recent trade. The current price changes whenever a new trade of the given security occurs.

Daily change price

Per unit difference between the current price and the last closing price of an asset.

Daily high price

Highest per unit value reached by an asset during the progress of a trading day.

Daily low price

Lowest per unit value reached by an asset during the progress of a trading day.

Dividend

Latest per share dividend distributed by the company.

Earnings per share

Latest yearly earnings per share reported by the company.

Executed per unit amount

The dollar amount per unit for a security that is bought or sold in a trade transaction. This amount is final (not an estimate).

Executed total amount

Total dollar amount of a transaction. It is computed as the number of shares multiplied by the executed per unit amount.

Fifty-two week range

A pair of values showing the highest and the lowest closing prices per unit that occurred in the last 52 weeks.

IRA

Individual Retirement Account. A special type of account which must conform with IRS rules in order to obtain tax reductions.

IRS

Internal Revenue Service. The tax collection agency for the US Government.

Limit trade order

A trade order placed in the securities market to buy or sell stock at a specific price. The order can be executed only at that price or a better one. It sets the maximum price the customer is willing to pay as a buyer, and the minimum price the customer is willing to accept as a seller.

Marital status

Customer marital status: single, married, divorced.

Market trade order

A trade order placed in the securities market to buy or sell stock immediately at the best current price.

Minimum cash asset

The minimum amount of cash that an account is required to have at all times. See Business Rule Account Cash Minimum for the calculation formula of the minimum cash asset.

News list

List of the sixty latest news headlines that relate to a particular stock symbol-obtained from the News System.

Number of dependents

Customer's number of dependents: spouse, children or other legal dependents.

Number of shares owned

Total number of shares of a given security that are owned in an account.

Number of shares per transaction

Number of shares of a given security bought or sold in a transaction.

Opening price

Per unit price of the first trade of an asset during a trading day.

Percentage price change

Percentage of change between the current price and the previous close price, using the previous close price as the base.

Previous close

Per unit price of the last trade of an asset on the previous trading day.

Price/earnings ratio

Ratio of the current price to the earnings per share.

Quote

The unit price and other information about a given security, as obtained from the Quote System.
Components are:

- Current price
- Daily high price
- Daily low price
- Opening price
- Previous close
- Bid – ask price
- Fifty-two week range
- Percentage price change
- Volume
- Price/earnings ratio
- Earnings per share
- Dividend
- Yield
- Chart displaying the stock history for the last twelve months

RU e-st

Abbreviation for RationalU e-st, the system under development for online securities trading.

RU e-st terms and conditions

"Fine print" that describes how RU e-st operates from the business point of view.

Security

An asset that is a stock, bond, or mutual fund. Securities are assets that can be traded.

Securities list

A list of all securities that can be traded through RU e-st.

Securities trading

A transaction in which the ownership of a security changes from one person to another.

Social security number

Customer's social security number

Stock symbol

Trading symbol used to identify a particular stock.

Time stamp executed transaction

Date and time of the market execution of a transaction.

Total current shares traded

Total number of shares of a given security traded as the trading day progresses.

Trade order type

Trade orders can be of the following types:

- **Market**: the order is placed in the market to buy or sell stock immediately at the best current price.
- **Limit**: allows the customer to buy or sell stock at a specific price. The order can be executed only at that price or a better one. It sets the maximum price the customer is willing to pay as a buyer, and the minimum price the customer is willing to accept as a seller
- **Transfer**: allows the customer to transfer dollars from mutual fund asset in a source account to mutual fund assets in a target (could be the same) account. The customer can not transfer funds to an asset which they do not already own shares. The customer can only transfer funds between mutual funds.

Trade transaction type

Trade transactions can be of type:

- Buy
- Sell

Trading customer

A person who has a customer id and password in the RU e-st system.

Trading symbol

A unique identifier for a given stock or other security. The trading symbol identifies the security that is being bought or sold.

Transaction fees

Total cost charged to the customer for a given transaction. See corresponding Business Rule.

Transaction id

Unique identifier of a customer transaction.

Transfer amount

Dollar amount involved in a transfer between accounts.

Transfer trade order

A trade order which allows the trading customer to transfer dollars from one mutual fund asset in a source account to mutual fund assets in a target (could be the same) account. The customer cannot transfer funds to an asset in which they do not already own shares. The customer can only transfer funds between mutual funds.

Volume

The total number of shares of a given security that have been traded on the current day.

RU Financial Services

RU e-st Product Vision

Version 1.3

Revision History

Date	Version	Description	Author
19 January, 2002	1.0	First draft	J. Carlos Goti
27 January, 2002	1.0	First complete draft	J. Carlos Goti
17 November, 2002	1.2	Revision for clarification	J. Bell
November 25, 2002	1.3	Editorial changes	B. Baker

Table of Contents

1. Introduction 5
   - 1.1 Purpose 5
   - 1.2 Scope 5
   - 1.3 Definitions, Acronyms, and Abbreviations 5
   - 1.4 References 5
   - 1.5 Overview 5

2. Positioning 5
   - 2.1 Business Opportunity 5
   - 2.2 Problem Statement 5
   - 2.3 Product Position Statement 6

3. Stakeholder and User Descriptions 6
   - 3.1 Market Demographics 6
   - 3.2 Stakeholder Summary 6
   - 3.3 User Summary 7
   - 3.4 User environment 7
   - 3.5 Stakeholder Profiles 8
     - 3.5.1 Financial Company Business Manager 8
     - **3.5.2 RU e-st** Product Manager 8
     - **3.5.3 RU e-st** Business Manager 8
     - **3.5.4 RU e-st** Development Manager 9
     - **3.5.5 RU e-st** User 9
     - **3.5.6 RU e-st** Partner Executive 10
   - 3.6 User Profiles 10
     - 3.6.1 User 10
     - **3.6.2 RU e-st** System Administrator 10
   - 3.7 Key Stakeholder / User Needs 11
   - 3.8 Alternatives and Competition 11
     - 3.8.1 Stay with the broker-phone method 12
     - 3.8.2 Buy an existing online securities trading service company 12
     - 3.8.3 Internally develop a securities trading system 12
     - 3.8.4 Buy an off-the-shelf securities trading system and install it. 12

4. Product Overview 12
   - 4.1 Product Perspective 13
   - 4.2 Summary of Capabilities 13
   - 4.3 Assumptions and Dependencies 13
   - 4.4 Cost and Pricing 14
   - 4.5 Licensing and Installation 14

5. Product Features 14
   - 5.1 Online market trading 14
   - 5.2 Online limit trading 14
   - 5.3 Online account setup 14
   - 5.4 Online quotes 14
   - 5.5 Online news 14
   - 5.6 Online account reports 14

5.7 Online transfers between accounts 14
5.8 Online transfers with banks 14
5.9 Online transfers with credit card account 15
5.10 IRA rollover 15
5.11 Historical account reports 15
5.12 Income tax account reports 15
5.13 IRS reporting forms 15
5.14 Netscape® and Microsoft® Internet Explorer® browser support 15
5.15 Other browser support 15

6. Constraints 15

7. Quality Ranges 15

8. Precedence and Priority 15

9. Other Product Requirements 16
9.1 Applicable Standards 16
9.2 System Requirements 16
9.3 Performance Requirements 16
9.4 Environmental Requirements 16

10. Documentation Requirements 16
10.1 User Manual 16
10.2 Online Help 16
10.3 Installation Guides, Configuration, Read Me File 16
10.4 Labeling and Packaging 16

11. Appendix 1 - Feature Attributes 17
11.1 Status 17
11.2 Priority 17
11.3 Effort 17
11.4 Risk 18
11.5 Stability 18
11.6 Target Release 18
11.7 Assigned To 18
11.8 Rationale 18

RU e-st Project

1. Introduction

The purpose of this document is to collect, analyze and define high-level needs and features of the **RationalU e-st** (**RU e-st**) system. It focuses on the capabilities needed by the stakeholders, the target users, and why these needs exist. The details of how RU e-st fulfills these needs are shown in the use case and supplementary specifications.

This document describes the vision for the proposed RU e-st system. The RU e-st system lets its users trade securities online, using a Web interface (an existing browser). It enables its users to perform the basic operations: open accounts, trade, and obtain information about their accounts and about what is happening in the securities markets.

1.1 Purpose

The objective of writing a Vision document for the RU e-st system is to enable agreement among stakeholders, developers and business representatives. Another purpose of writing a Vision document is to provide a common platform for agreement between the developers themselves.

1.2 Scope

This document records the vision for the RU e-st system. It is associated with a project to provide Web-based trading facilities through financial institutions.

1.3 Definitions, Acronyms, and Abbreviations

Definitions of terms and allowed operations match the standard ones used in the field of securities trading. See the Glossary for details.

1.4 References

N/A

1.5 Overview

This document contains product positioning statements, an analysis of the system's stakeholders, an analysis of the expected system users, and a list of features that the system should deliver. These features are derived from the input provided by the stakeholders.

2. Positioning

2.1 Business Opportunity

This product will be sold to financial institutions that want to provide online trading to their customers. Many of these institutions are expected to join this market soon and will need a system to support their customers on short notice (that is, will not have the time to develop their own systems).

2.2 Problem Statement

The problem of	unmet customer demand for trading securities online.
Affects	financial institutions that find themselves in a poor competitive position.
The impact of which is	loss of customers and commissions.
A successful solution would	enable a quick installation and delivery of the online trading facilities.

2.3 Product Position Statement

For	Financial institutions
Who	see their customers moving to the competition.
The RationalU e-st	is a Web-based securities trading system
That	has full function and can be quickly installed and put in production.
Unlike	in-house or contract development of such a system
Our product	is packaged and load-tested to assure immediate success.

3. Stakeholder and User Descriptions

3.1 Market Demographics

Online securities trading companies are taking away customers from traditional trading companies. These traditional trading companies and other financial institutions that envision entering the securities trading business are the main set of prospects to buy the RationalU e-st system.

Globalization, Internet popularity trends, and financial institution deregulation trends indicate that the set of potential customers is very large and will increase over time.

RU Financial Services has hired a consultant to assess the size of this market and expect to have the final results during 2Q01.

3.2 Stakeholder Summary

There are typically a large number of stakeholders for a project. The following table lists the ones who have primary responsibilities and have the largest influence on the project's success.

Name	Represents	Role
Financial Company Business Manager	The customer who is going to buy the system.	Concerned with ROI for his or her company.
RU e-st Product Manager	The set of customers who will buy the system.	Defines a product that sells according to plan. Evaluates options and scopes the different versions of the system.
RU e-st Business Manager	The business entity that makes the funding decisions for development.	Ensures that the company has the plans and resources to deliver the system and that it is a good business decision for the company to spend the funds.

RU e-st Development Manager	The team that develops and releases the system.	Ensures that the development team delivers a system that complies with the specifications supported by the RU e-st Product Manager.
RU e-st End User	A person who trades securities using the system.	Validate that the features offered by the system are adequate and responsive to the needs of traders.
RU e-st Partner Executive	Companies whose systems support the operation of the system. They are: Market Trading, News System, and Quote System.	Approve business agreements for specialized functions needed to support online securities trading.

3.3 User Summary

Name	Description	Stakeholder
End User - Day Trader	A knowledgeable user who is on the system frequently and uses advanced features.	RU e-st End User
End User - Naïve	First-time or infrequent end user of the system. This person needs guidance and may need to refer to online help for definitions and business consequences of trading actions.	RU e-st End User
RU e-st Administrator	Does customization and setup activities on the system. Maintains up-to-date information and produces and analyzes operational reports of system performance.	RU e-st Development Manager
RU e-st Support	Helps calling customers use the system correctly. Reports errors and develops operational fixes.	RU e-st Development Manager
RU e-st Developer	Part of the team that develops the system (software developer, tester, and so on.)	RU e-st Development Manager

3.4 User Environment

Users typically work alone. They have logged on to the Internet (hotel room, airport, etc.) and want to check the status of their investments or perform one or more trades.

Except for the people who share an account, there is no interplay between the users. They are independent of one another.

It is expected that many thousands of users will be trading at the same time.

Users will access the system through a standard Internet browser; they do not need to learn new user interface (UI) techniques. They will access the system from many different client platforms such as Windows, Macintosh, Linux, OS/2, and so on.

3.5 Stakeholder Profiles

3.5.1 Financial Company Business Manager

Representative	The customer who is going to buy the system.
Description	Business manager concerned with return on investment (ROI) for his or her company.
Type	This stakeholder may not be an user of the system.
Responsibilities	Ensures a good return for his or her company when using RU e-st.
Success Criteria	The business success of the financial institution's use of the system is measured by the number of customers using it and the commissions obtained through their trading. Also, there can be "customer" drag from the securities trading activities into other services of the institution. In some ways, the financial institution shall have the online trading service to stay in business, so even if it is a financial loss, the institution may need to keep it going.
Involvement	Mostly from the business side. Wants a system that is delivered within cost and function promises.
Deliverables	Considers the system an off-the-shelf purchase. Will need demos and assessment from staff.
Comments/Issues	Need to ensure that the system competes well with the competition.

*3.5.2 **RU e-st** Product Manager*

Representative Of	The set of customers who will buy the system.
Description	Defines a product that sells according to plan. Evaluate options and scope the different versions of the system.
Type	Is an expert user of the system.
Responsibilities	Does the business analysis of the system for justification to the RU e-st Business Manager. Achieves a definition of the system that can be built and delivered on time and within cost. Commits the marketing and sales resources to make money with the product.
Success Criteria	The product (system) is accepted and makes money.
Involvement	Reviews plans and development progress. Assesses how features are being supported. Validates of expenses by system feature.
Deliverables	Participates in iteration assessment sessions and is present when project decisions are made.
Comments / Issues	Needs to keep an eye on the competition to ensure success in the marketplace.

*3.5.3 **RU e-st** Business Manager*

Representative	The business entity that makes the funding decisions for development.
Description	Ensures that the company has the plans and resources to deliver the system and that it is a good business decision for the company to spend the funds.
Type	Probably will not be a user of the system.

Responsibilities	Ensures that the funds are committed to the project and adjustments are made with a sound business sense. Influences marketing and sales to prioritize the system according to the company's vision. Promotes the system to potential customers.
Success Criteria	The system makes money for the company and is aligned with the overall strategy. The company gets good press from the system, and the stock price rises.
Involvement	Participates in the system's phase assessments.
Deliverables	Provides the project team with resources and promote the system vision with good press inside and outside the company.
Comments/Issues	The RU e-st Business Manager should be supported with solid marketing and positioning information about the system.

*3.5.4 **RU e-st** Development Manager*

Representative Of	The team that develops and releases the system.
Description	Assures that the development team delivers a system that complies with the specifications supported by the RU e-st Product Manager.
Type	Very proficient in the use of the system.
Responsibilities	Allocates the resources and provide the technical and business controls to assure the success of the project.
Success Criteria	The project completes a sound delivery that can be sold, as specified by the RU e-st Product Manager.
Involvement	The Development Manager works closely with the Project Manager, the Architect, and the Analyst to ensure that all is on the right track.
Deliverables	Support the project with controls, imaginative resolution of problems, allocation of resources, and team building activities.
Comments / Issues	The Development Manager can act as a technical leader in some areas of expertise.

*3.5.5 **RU e-st** User*

Representative Of	People who trade securities using the system.
Description	Validates that the features offered by the system are adequate and responsive to the needs of traders.
Type	End users may have different expertise levels - some are naïve, others are experts.
Responsibilities	User contacts are responsible for feedback about the system's function and capabilities.
Success Criteria	95% of the targeted user population is satisfied with the features offered by RU e-st.
Involvement	Validates use cases and participates in iteration reviews.
Deliverables	Comments and sometimes signs off on different artifacts (use cases, UI designs, and so on).
Comments / Issues	Users may give information that is subjective. Their comments may not apply to most of or all the user population. Input from one user can be validated using input from other users. The development team expects users to have been validated as sound examples of the using population so that their feedback is meaningful.

*3.5.6 **RU e-st** Partner Executive*

Representative	Companies whose systems support the operation of RU e-st. They are: Market Trading, News System, Quotes System, and the Financial Network System.
Description	Approve business agreements for specialized functions needed to support online securities trading.
Type	They may not use the system at all.
Responsibilities	Achieve business agreements for cooperation between systems. Agree on business synergism and mutual charging schemes.
Success Criteria	Smooth business operation (low bureaucracy) during the system's joint operations.
Involvement	Early agreements during the inception phase of the system. The business arrangements should be completed early in the development process.
Deliverables	Mutual service support during the operation of the systems. Clear arrangements for production problem resolution.
Comments / Issues	Business arrangements are limited in time. Backup arrangements are always good to have. In some cases we may want to have two or more partners providing the same services, to minimize full dependencies.

3.6 User Profiles

Users are listed in two categories: end users and RU e-st system administrators. The breakdown shown above is too detailed for this section.

3.6.1 User

Representative	The RU e-st Product Manager is the representative of End Users.
Description	Use the RU e-st system to perform securities trades.
Type	The Day Trader is very knowledgeable of the system's operations and features because of constant utilization. The naïve user is either a first-time user or an infrequent user.
Responsibilities	End users give the development team feedback about how the system supports their trading needs. They should point out lacking features and awkward operational characteristics of the system.
Success Criteria	Users benefit from an improved system access to their needs.
Involvement	Users should participate in the description of use cases, the evaluation of partial implementations, and assessment of the help features of the system.
Deliverables	Users deliver feedback to the team's analysts and architect either verbally or in writing.
Comments / Issues	It is very important to the project to have good user representatives. They can improve the system by sharing their domain expertise and operational needs.

*3.6.2 **RU e-st** System Administrator*

Representative	System administrators are represented by the RU e-st Development Manager.
Description	System administrators do set up, customization, support and development of the system.
Type	All system administrators are users at the expert level.

Responsibilities	Prepare and execute installation and customization operations. Tend to production problems (errors, performance, work-around solutions, etc.) Support end users that call for help or to report errors. Develop and test parts of the system.
Success Criteria	Customer satisfaction measures.
Involvement	Should participate during development in the same way that end users do. They should concentrate on the support and maintenance use cases.
Deliverables	Comments to improve the support and maintenance aspects of the system.
Comments / Issues	Often, the system administrator users are not brought on-board in time to improve the system. A special effort should be made to get their feedback as early as possible.

3.7 Key Stakeholder / User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Compete with other companies that offer online securities trading	High	Our financial institution does not offer online trading and customers are moving to other companies.	Trading involves a broker and is done by phone.	Install online trading.
Ease of trading securities	High	Speed and ease of securities trading.	Trade by phoning a broker.	Trade online, at the convenience of the trader.
Cost of trading securities	High	It costs too much to trade securities.	Go through a broker.	Trade online, at reduced rates.
Access to securities information	Medium	Users can get recent information about securities from different sources. A source from where trading is done is valuable.	Access securities information and then call the broker.	Access securities information online and perform the operation from the same site.
Fast trade confirmation	High	Trades are ordered but the final results need to be confirmed.	Broker calls back on the phone (or mailer arrives).	By waiting a few seconds, the order confirmation is delivered online.

3.8 Alternatives and Competition

A financial institution that wants to enter the securities trading business has the following alternatives:

3.8.1 Stay with the broker-phone method

This is not a real alternative. Soon the broker-phone method will be a business of the past. Traders are demanding instant response, which can only be achieved through online trading. The competition of online trading systems will cause the "traditionalists" to disappear.

3.8.2 Buy an existing online securities trading service company

The financial institution could buy an existing online trading company and merge with it. This is surely a possibility, but there are more institutions wanting to buy than the number of companies in a position to be bought. Most companies trying to make this move will need an alternative.

3.8.3 Internally develop a securities trading system

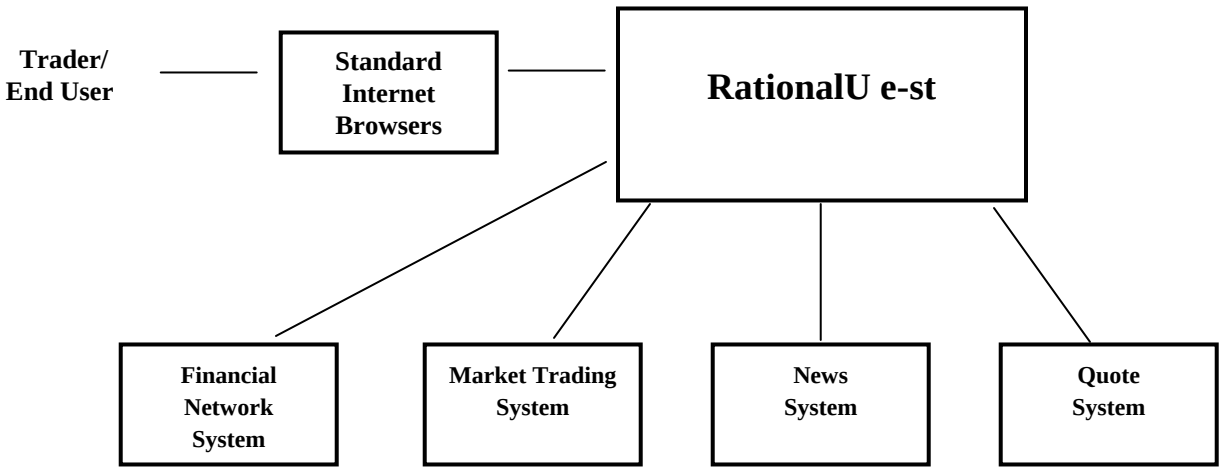
This could be an attractive proposition, especially for large companies with large IT organizations. The question is how long it would take to do the development, testing, validation, and installation. More and more companies are opting to buy a non-strategic system and spend their resources on truly strategic projects. They would rather buy and install a system quickly and get the benefits of it immediately.

3.8.4 Buy an off-the-shelf securities trading system and install it.

This option is attractive because it eliminates development risks and produces results very quickly. We think many companies will opt for this solution, especially after an installation that shows the RU e-st as a reliable system and provides a quick, successful entry into the securities trading market.

4. Product Overview

RU e-st is planned to interface with other systems in the following fashion:



All end users will access the system's features through standard Internet browsers. The system provides facilities for traders to set up accounts, get securities quotes, obtain news headlines about specific securities, and perform trades. Information about user's accounts and the assets in them, are kept by RU e-st.

4.1 Product Perspective

RU e-st interacts with other systems for some specific tasks. They are:

- **Market-Trading System**: to complete trade orders as specified by the trader.
- **News System**: to retrieve headline news and the corresponding links to full articles for a given security.
- **Quote System**: to retrieve current and historical quote information for a given security.
- **Financial Network System**: to perform money transfers to and from the customers' financial institution accounts.

One of the features of the system is to easily adapt to different instances of the systems listed above because different customer financial institutions may want to connect to different servicing systems providing equivalent functionality.

Another high priority feature is to interact with all Internet browsers that are commonly used by trading end users.

4.2 Summary of Capabilities

RationalU e-st System	
Customer Benefit	**Supporting Features**
Trading online	• Online trading • Online account setup • Online quotes • Online news
Securities research online	• Online quotes • Online news • Online account reports
Portfolio management online	• Online account setup • Online accounts reports • Online transfers between accounts • Online transfers with banks
Portfolio record keeping	• Historical account reports • Income tax account reports • IRS reporting forms

4.3 Assumptions and Dependencies

The basic features of the system are attainable with today's technologies. No high-risk dependencies are envisioned.

4.4 Cost and Pricing

Marketing is working on the pricing points.
Development is working on the cost estimates.

4.5 Licensing and Installation

To be determined.

5. Product Features

5.1 Online market trading

The system must provide market trading of stocks and mutual funds directly to/from a customer's account.
For market sales, the securities are in the customer's account.
For market buying, the required cash must be in the customer's account.

5.2 Online limit trading

The system must provide online limit trading similar to market trading, except that customers place an upper limit for a buy and a lower limit for a sale. The system is expected to comply with these limits.

5.3 Online account setup

The customer must be allowed to create an account online. Since no operations are allowed until there is a cash asset in the account, the customer needs to transfer the money into a new account to activate it.

5.4 Online quotes

Current and historical quotes for a particular trading symbol must be presented for the customer's information.

5.5 Online news

Headline news must be presented together with links to complete Internet documents for a particular trading symbol.

5.6 Online account reports

For each of the customer's accounts, and for each of the assets in the account, the system must present the current unit price, the number of shares, and the current total value.

5.7 Online transfers between accounts

The customer must be able to transfer assets from one of his/her accounts to another.
The system must only allow transfers that leave the account in a valid state (regarding the minimum cash asset business rule).

5.8 Online transfers with banks

The customers must be able to transfer out or into the cash asset of an account to or from a bank account.

5.9 Online transfers with credit card account

The customer must be able to charge to a credit card and place the funds into the cash asset of an account. The customer can transfer funds from a cash asset in an account as payment to a credit card.

5.10 IRA rollover

Customers must be able to rollover from an IRA in another institution into a retirement account managed by the system.

5.11 Historical account reports

The system must present the customer for each account and asset within it the number of shares, the current unit price, the total price and all of the buys and sells within the last year with the corresponding number of shares, unit price, total price, commission costs, and the date and time of the transaction.

5.12 Income tax account reports

For each account, the system must present to the customer the IRS and State relevant year-to-date information and the previous year totals. The information is: short and long term capital gains, distributions, dividends and interests.

5.13 IRS reporting forms

The system must present a summary of yearly totals as needed to report to the IRS and state tax boards. These forms must be communicated to the IRS and copies mailed to the customer. The information is also available online for customer viewing.

5.14 Netscape Navigator and Microsoft Internet Explorer browsers support

The user interface must be compatible with the last three levels of Netscape Navigator and Microsoft Internet Browser.

5.15 Other browsers support

The system must support other browsers, to be determined as required by customers purchasing the system.

6. Constraints

All of the online features must run on the selected Internet browsers.

Need to perform an additional assessment by the end of the elaboration phase.

7. Quality Ranges

Refer to expected industry standards in this area. Further research is needed.

8. Precedence and Priority

This information is included in the RequisitePro database. Appropriate reports are designed to convey this information.

9. Other Product Requirements

9.1 Applicable Standards

Refer to standards related to the versions of HTML and Java by the supported Internet browser.

Refer to standards for financial transactions between institutions in the Financial Network .

The application must conform to all the rules of the NASD, SIPC and other trading institutions involved in transactions.

9.2 System Requirements

The customer needs a computer (client) with capabilities to access the Internet. Access speed and local hard drive usage depends on the customer's configuration and practices of use.

9.3 Performance Requirements

Transactions (except for trades, which require a sale or buy operation) must execute in an Internet comfortable speed (less than 3 seconds, not counting transmission times).

9.4 Environmental Requirements

No additional requirements other than the ones in common practice for a computer with Internet connection capability.

10. Documentation Requirements

All documentation for the RU e-st system is available online and not in print form.

We may need some printed information at the brochure level for mailings and "getting started" help. This information should describe trading terms and conditions and a basic description of capabilities.

10.1 User Manual

No user manual will be offered.

10.2 Online Help

The system offers extensive online help.

The help has two basic objectives:
Assistance in the operation of the system
Assistance with the domain (definitions, descriptions of accounts, transactions, legal issues, ad so on.)

10.3 Installation Guides, Configuration, Read Me File

No installations are required. No need for this documentation.

10.4 Labeling and Packaging

There is no applicable labeling and packaging activity.

11. Appendix 1 - Feature Attributes

These attributes are assigned to each feature and listed in RequisitePro.

11.1 Status

Proposed	Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management, and user or customer community.
Approved	Capabilities that are deemed useful and feasible and have been approved for implementation by the official channel.
Incorporated	Features included into the product baseline at a specific point in time.
Validated	The quality organization has approved the implementation of this feature.

11.2 Priority

High	Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature.
Medium	Features that are useful in less typical applications will be used less frequently, or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.
Low	Not significant to the success of the product in the marketplace.

11.3 Effort

Effort is recorded as a numerical value in person-month units. For example, 4PM.

11.4 Risk

High	The development team is very concerned about completing this feature with the given constraints. Reasons for the concern are documented in a separate document.
Medium	The development team feels comfortable implementing this feature under the constraints. But, there are some unknown elements that need to be overcome. Reasons for these concerns are documented in a separate document.
Low	The development team is fully confident on being able to implement the feature within the project's constraints. The team has done similar work before with success.

11.5 Stability

High	The analyst and the development team are very confident that the contents of the feature and their understanding of it is complete and has little chance of changing.
Medium	The analyst and the development team feel that even though they understand the contents of the feature, there is some work left to make it very precise. Some additional elicitation is needed.
Low	The analyst and the development team feel that their understanding of the contents of the feature is incomplete or could easily change. Additional analysis work is really needed.

11.6 Target Release

A code that denotes the release in which the feature will be delivered. It is recorded as a decimal number proceeded by a "V". For example: V3.2.

11.7 Assigned To

The name of the analyst responsible for the feature in the format of first and last name. For example: Sue Collins.

11.8 Rationale

A reason for having this feature in the system. It can be a short explanation or a pointer to where the reason can be found. For example: "the main prospect wants this" and "see standard XYZ."

RU Financial Services

Use Case Report: Execute Trade

Version 1.3

Revision History

Date	Version	Description	Author
Jan 13, 2002	1.0	First draft	J. Carlos Goti
Jan 15, 2002	1.0	First complete draft	J. Carlos Goti
Nov 15, 2002	1.2	Revised for clarification	J. Bell
Dec 1, 2002	1.3	Editorial changes.	B. Baker

Table of Contents

1. Use Case Name: Execute Trade 4
   - 1.1 Brief Description 4

2. Flow of Events 4
   - 2.1 Basic Flow 4
   - 2.2 Alternative Flows 5
     - 2.2.1 Limit Sell Order (Non-Retirement Accounts only) 5
     - 2.2.2 Limit Buy Order (Non-Retirement Accounts only) 5
     - 2.2.3 Market Sell Order (Non-Retirement or Retirement Accounts) 5
     - 2.2.4 Market Buy Order (Non-Retirement or Retirement Accounts) 5
     - 2.2.5 Transfer Order (Retirement Accounts Only) 5
     - 2.2.6 Verify That Cash Is Available 5
     - 2.2.7 No Accounts 6
     - 2.2.8 Account Not Operational 6
     - 2.2.9 Unidentified Trading Customer 6
     - 2.2.10 Unauthorized Trader 6
     - 2.2.11 Quit 6
     - 2.2.12 Trade Is Rejected 6
     - 2.2.13 Market Trading System Unavailable 6

3. Special Requirements 6
   - 3.1 Reliable Cash Accounting 6

4. Pre-Conditions 6
   - 4.1 RU e-st Has Connection to Market Trading System 6

5. Post-Conditions 7
   - 5.1 Accounts Are Adjusted 7

6. Extension Points 7

7. Relationships 7

8. Use-Case Diagrams 7

9. Other Diagrams 7

Use Case Report: Use Case Report: Execute Trade

1. Use Case Name: Execute Trade

1.1 Brief Description

Trading Customers buy, sell, or transfer securities in their accounts. Immediately after a trade is completed, Trading Customers receive a market trading confirmation containing a transaction id and the trade results.

2. Flow of Events

2.1 Basic Flow

1. Customer Logs On

The use case starts when the Trading Customer logs on. The system validates the customer ID and password. The system presents a list of available functions.

2. Customer Selects "Trade" Function

The Trading Customer chooses to "Trade." The system displays the customer's Account List.

3. Customer Selects Account

The Trading Customer selects an account. The system presents the Account Display for the selected account. The system verifies that the account status is operational. The system displays available types of trade orders, based on the account type.

4. Customer Performs Trade

4.1 Prepare To Trade

The Trading Customer selects a trade order type. Depending on the trade order type selected, one of the following Alternative Flows is performed.

2.2.1 Limit Sell Order (Non-Retirement Accounts only)
2.2.2 Limit Buy Order (Non-Retirement Accounts only)
2.2.3 Market Sell Order (Non-Retirement or Retirement Accounts)
2.2.4 Market Buy Order (Non-Retirement or Retirement Accounts)
2.2.5 Transfer Order (Retirement Accounts only)

4.2 Initiate Trade

The system sends the Market Trading Request to the Market Trading System.

4.3 Make Trade

The system presents the transaction reference identifier to the Trading Customer. The system debits or credits the account based on the assumed sale price. The system receives the Market Trading Confirmation from the Market Trading System. The system debits or credits the account based on the actual trade price. The system calculates fees and debits the account by subtracting the fees from the cash in the account. The system displays the Market Trading Confirmation to the Trading Customer.

5. Customer Begins New Transaction

If the Trading Customer wishes to begin a new transaction, repeat from step 3 until the Trading Customer indicates that they do not wish to make any additional trades.

6. Customer Logs Off

The Trading Customer logs off the system. The use case ends.

2.2 Alternative Flows

2.2.1 Limit Sell Order (Non-Retirement Accounts only)

In Step 4.1 of the Basic Flow, if the Trading Customer selects **Limit Sell Order – Non-Retirement Account**, then the Trading Customer enters Asset Limit Sale information. The use case resumes at Step 4.2 of the Basic Flow. *Notice that the Trading System may be able to partially sell the order because there may be no takers for all the shares at the limit price. Also, the Trading System may be able to sell some or all the shares at a price equal to or higher than the limit.*

2.2.2 Limit Buy Order (Non-Retirement Accounts only)

In Step 4.1 of the Basic Flow, if the Trading Customer selects **Limit Buy Order – Non-Retirement Account**, then the Trading Customer enters Asset Limit Purchase information. The use case performs Alternative Flow 2.2.6, Verify That Cash Is Available. The use case resumes at Step 4.2 of the Basic Flow. *Note that the Market Trading System must comply with the specified limit – this may result in a partial purchase because there were no takers for the total order in the market. Also the Trading System may be able to purchase some or all the requested shares at a price lower than or equal to the limit.*

2.2.3 Market Sell Order (Non-Retirement or Retirement Accounts)

In Step 4.1 of the Basic Flow, if the Trading Customer selects **Market Sell Order**, the Trading Customer enters Asset Sale information. The use case performs Alternative Flow 2.2.6, Verify That Cash Is Available. The use case resumes at Step 4.2 of the Basic Flow.

2.2.4 Market Buy Order (Non-Retirement or Retirement Accounts)

In Step 4.1 of the Basic Flow, if the Trading Customer selects **Market Buy Order**, the Trading Customer enters Asset Purchase information. The use case performs Alternative Flow 2.2.6, Verify That Cash Is Available. The use case resumes at Step 4.2 of the Basic Flow.

2.2.5 Transfer Order (Retirement Accounts Only)

In Step 4.1 of the Basic Flow, if the Trading Customer selects **Transfer Order – Retirement Account**, then the Trading Customer enters Dollar Transfer information. The use case performs Alternative Flow 2.2.6, Verify That Cash Is Available to assess compliance with the minimum cash rule, because load and fee costs will be deducted from the cash asset.
The system sends the buy and sell Output Market-Trading Requests to the Market Trading System. The use case resumes at Step 4.3 of the Basic Flow. *Note that the dollar transfer is done by selling a dollar amount of one mutual fund and buying the same dollar amount of another mutual fund, both funds already present in the account.*

2.2.6 Verify That Cash Is Available

If the type of trade order is **Market Buy**, **Limit Buy,** or **Transfer,** then the system calculates the total cost of the transaction (for Buy: number of shares*price + load + commission; for Transfer: load + commission) for compliance with the Account Cash Minimum Business Rule.

If there is sufficient cash in the account to comply with the rule, the use case resumes at the point in the alternative flow from which it came.

If the system determines that there is not enough cash in the account, the system notifies the Trading Customer that the transaction is not possible. The use case resumes at Step 4.1 of the Basic Flow.

2.2.7 No Accounts

In Step 2, Customer Selects "Trade" Function, in the Basic Flow, if there are no accounts for this Trading Customer, the system notifies the Trading Customer. The use case ends.

2.2.8 Account Not Operational

In Step 3, Customer Selects Account, in the Basic Flow, if the account status is not operational, the system notifies the Trading Customer that no trades are permitted on this account. The use case resumes at Step 2, Customer Selects "Trade" Function, in the Basic Flow.

How often should we check to see if the status of the account has changed from operational to non-operational? The status could change without any trade happening, because it depends on the value of stock and other assets versus the amount of cash in the account. Right now, we've assumed that we check only when the Trading Customer begins a transaction. Maybe we should check at other times too?

[This is a typical way to employ use cases. You may write your questions right down in the text and when you get your answers you have to correct the text accordingly. Or you may assume one way and let the reviewers decide; either the reviewers like it or they tell you how it should be.]

2.2.9 Unidentified Trading Customer

In Step 1, Customer Logs On, in the Basic Flow, if the system determines that the customer ID and/or password are not valid, an error message is displayed. The use case ends.

2.2.10 Unauthorized Trader

In Step 2, Customer Selects "Trade" Function, in the Basic Flow, if the system determines that the Trading Customer is not authorized to execute trades, the system notifies the Trading Customer. The use case ends.

2.2.11 Quit

The RU e-st System allows the Trading Customer to quit at any time during the use case. All trades that are in progress are cancelled. The use case ends.

2.2.12 Trade Is Rejected

In Step 4.3, Make Trade, in the Basic Flow, if the Market Trading System rejects the Market Trading Request, the system notifies the Trading Customer that the trade is rejected. The use case continues at Step 5, Customer Begins New Transaction, in the Basic Flow.

2.2.13 Market Trading System Unavailable

In Step 4.3, Make Trade, in the Basic Flow, if the system is unable to communicate with the Market Trading System, the system informs the Trading Customer. The use case ends.

3. Special Requirements

3.1 Reliable Cash Accounting

The RU e-st system shall round any changes to a trading account to the nearest penny.

4. Pre-Conditions

4.1 RU e-st Has Connection to Market Trading System

The RU e-st system must have a connection to the Market Trading System in order to begin this use case.

5. Post-Conditions

5.1 Accounts Are Adjusted

At the end of this use case, the system leaves all accounts in a sound accounting state (there is no extra money or lost money to the penny, and all changes can be explained).

6. Extension Points

None specified for this use case.

7. Relationships

The Actor starting this Use Case is:

Trading Customer

Actor(s) also involved in this Use Case:

Market Trading System

8. Use-Case Diagrams

None specified for this use case.

9. Other Diagrams

None specified for this use case.

RU Financial Services

Supplementary Specification

Version 1.3

Revision History

Date	Version	Description	Author
Mar 28, 2002	1.0	Added Business Rules and Inputs/Outputs	J. Carlos Goti
November 17, 2002	1.2	Additions and reorganization	J. Bell
November 22, 2002	1.3	Editorial changes.	B. Baker

Table of Contents

1. Introduction 5
   - 1.1 Purpose 5
   - 1.2 Scope 5
   - 1.3 Definitions, Acronyms and Abbreviations 5
   - 1.4 References 5
   - 1.5 Overview 5

2. Functionality 5
   - 2.1 Localization 5
   - 2.2 Business Rules 6
     - 2.2.1 Account Cash Minimum 6
     - 2.2.2 Limit Order Allowable Accounts 6
     - 2.2.3 Market Order Allowable Accounts 6
     - 2.2.4 RU e-st terms and conditions 6
     - 2.2.5 Transaction Fees 6

3. Usability 6
   - 3.1 Browser compatibility 6
   - 3.2 Access to information 6
   - 3.3 Online tutorial 6

4. Reliability 6
   - 4.1 Availability 6
   - 4.2 Timeliness and Accuracy 6

5. Performance 7
   - 5.1 Response Time 7

6. Supportability 7
   - 6.1 Installability 7

7. Design Constraints 7

8. Online User Documentation and Help System Requirements 7

9. Purchased Components 7

10. Interfaces 7
    - 10.1 Inputs 7
      - 10.1.1 Asset Limit Purchase Information 7
      - 10.1.2 Asset Limit Sale Information 7
      - 10.1.3 Asset Purchase Information 8
      - 10.1.4 Asset Sale Information 8
      - 10.1.5 Dollar Transfer Information 8
      - 10.1.6 New Account Information 8
      - 10.1.7 News System Response 8
      - 10.1.8 Quote System Response 9

10.2 Outputs 9
10.2.1 Account List 9
10.2.2 Headline News Display 9
10.2.3 Market Trading Confirmation 9
10.2.4 Market Trading Request 10
10.2.5 Account Display 10
10.2.6 Quote Display 10
10.2.7 Stock Yearly History Chart 10
10.3 Systems with Which to Interface 10
10.3.1 Quote System 10
10.3.2 Market Trading System 10
10.3.3 News System 11
10.3.4 Financial Network System 11
10.3.5 IRS Reporting System 11

11. Licensing Requirements 11

12. Legal, Copyright and Other Notices 11

13. Applicable Standards 11
13.1 NASD, SIPC rules 11

Supplementary Specification

1. Introduction

The purpose of this document is to collect, analyze and define the supplementary specifications for the RU e-st system.

1.1 Purpose

This document records specifications that have general applicability to the RU e-st product, with the objective of achieving agreement of what the system does between stakeholders and developers.

1.2 Scope

This document is a record of specifications that affect the RU e-st product as a whole, or large parts of it.

1.3 Definitions, Acronyms and Abbreviations

See Glossary.

1.4 References

Vision document
Glossary
Use cases

1.5 Overview

This document contains the supplementary specifications, organized by type of requirement: Functionality, Usability, Reliability, Performance, Supportability, Design Constraints, Documentation Requirements, Purchased Components, Interfaces, Licensing Requirements, Legal Requirements, and Applicable Standards.

2. Functionality

Functionality at a high level is documented in the Vision document. Detailed functional requirements are primarily documented in the Use Case Reports for individual uses cases of the RU e-st system. Here we record a localization requirement that pertains to all use cases, and business rules for the RU e-st system.

2.1 Localization

The application will allow the user to select a language preference of English, Spanish, French, German or Swedish. The application will display all information to the customer in their selected language.

2.2 Business Rules

2.2.1 Account Cash Minimum

For each account that a customer has and independently of other accounts (with different asset values and a cash asset), the cash asset in the account must be maintained at 10% of the total account value for the status of the account to be "operational" – meaning that trade transactions are allowed.

RU e-st forecasts the cash value percentage of the account after a transaction proposed by the customer and prevents the transaction from being executed if it would break this rule.

An account may break this rule without a customer transaction because of market value fluctuations. RU e-st performs periodic (how often?) checks and notifies customers of their accounts switched to probationary status because of this reason. But in no case will a transaction be allowed if the account is in probationary state or if the transaction is estimated to place the account in probationary state.

If it is considered that a transaction will not cause the account to violate this rule, the transaction is allowed. Once the Trading System returns the final results of the transaction, RU e-st recalculates the cash value percent and places the account in probationary status if the rule is broken.

2.2.2 Limit Order Allowable Accounts

Limit orders for stock are only allowed on non-retirement accounts.

2.2.3 Market Order Allowable Accounts

Market orders for stock are only allowed on non-retirement accounts.

2.2.4 RU e-st terms and conditions

An account cannot be in operational state if the customer has not indicated that he-she agrees with the terms and conditions.

2.2.5 Transaction Fees

When a transaction is confirmed by the Trading System, the fees are calculated and deducted from the cash in the account. These monies are transferred to the RU e-st holdings.

3. Usability

3.1 Browser compatibility

The application will be able to run under either MS Internet Explorer or Netscape Navigator.

3.2 Access to information

The user will be able to gather all information related to their accounts on one screen.

3.3 Online tutorial

The application will provide an online tutorial for the customer.

4. Reliability

4.1 Availability

The application will be up and available 100% of the time during NYSE market hours. The application should not be down for more than 15 minutes at any other time.

4.2 Timeliness and Accuracy

The application must provide accurate market data that is no more than fifteen minutes old.

5. Performance

5.1 Response Time

A customer must receive status on their transaction within five seconds of placing the order. Even though the transaction may not execute immediately, the customer should receive notification that the order has been successfully accepted and is ready to be executed.

6. Supportability

6.1 Installability

The user will not have to install any additional software in order to use the application.

7. Design Constraints

TBD

8. Online User Documentation and Help System Requirements

TBD

9. Purchased Components

N/A

10. Interfaces

In this section we specify the interfaces that must be supported by the application. The first two subsections record the contents of all inputs and outputs that RU e-st has to support/provide. Also, we show the list of systems that RU e-st interfaces with.

10.1 Inputs

Each of the following sections specifies the content for a particular kind of input to the RU e-st system.

10.1.1 Asset Limit Purchase Information

Components are (for the asset the customer wishes to acquire):

- Trading symbol
- Number of shares
- Limit price

10.1.2 Asset Limit Sale Information

Components are (for the asset the customer wishes to sell):

- Trading symbol
- Number of shares
- Limit price

10.1.3 Asset Purchase Information

Components are (for the asset the customer wishes to acquire):

- Trading symbol
- Number of shares

10.1.4 Asset Sale Information

Components are (for the asset the customer wishes to sell):

- Trading symbol
- Number of shares

10.1.5 Dollar Transfer Information

Components are:

- For the source account:
  - Account id
  - Trading symbol
  - Dollar amount that they would like to transfer
- For the target account:
  - Account id
  - Trading symbol of the asset that they are going to transfer the funds to

10.1.6 New Account Information

Components are:

- Name
- Address
- Phone number
- E-mail address
- Social security number
- Date of birth
- Marital status
- Number of dependents
- Citizenship status
- Employment status
- Account operation type
- Account funding type
- The customer must also indicate that they agree with RU e-st terms and conditions

10.1.7 News System Response

Components are:

- List of headlines for the news stories for the last six months that relate to a given security

10.1.8 Quote System Response

Components are:

- Current price
- Daily high price
- Daily low price
- Opening price
- Previous close
- Bid – ask price
- Fifty-two week range
- Percentage price change
- Volume
- Price/earnings ratio
- Earnings per share
- Dividend
- Yield
- Chart displaying the stock history for the last twelve months

10.2 Outputs

Each of the following sections specifies the content for a particular kind of output from the RU e-st system. The name of each output gives the type of content of the output and (if not a display to the end-user) the system to which the output will be sent.

10.2.1 Account List

Components are:

- All accounts the customer has opened with RU e-st and for each:
  - Account id
  - Account type
  - Account total
  - Account alias

10.2.2 Headline News Display

- List of headlines for the news stories for the last six months that relate to a given security.

10.2.3 Market Trading Confirmation

Components are:

- Trading symbol
- Trade transaction type (buy or sell)
- Number of shares traded
- Executed price
- Date and time of the transaction
- Transaction id

10.2.4 Market Trading Request

Components are:

- Trading symbol
- Trade transaction type (buy or sell)
- Number of shares for requested trade
- Limit trade order price (a special code indicates “market” trade)
- Date and time of the transaction
- Transaction id

10.2.5 Account Display

Components are:

- All securities the customer has in the non-retirement account showing for each one:
  - Trading symbol
  - Asset name
  - Number of shares owned
  - Asset total value
  - Asset last transaction date
- Cash asset value
- Account total value
- Account status (operational or not)

10.2.6 Quote Display

Components are:

- Current price
- Daily high price
- Daily low price
- Opening price
- Previous close
- Bid – ask price
- Fifty-two week range
- Percentage price change
- Volume
- Price/earnings ratio
- Earnings per share
- Dividend
- Yield
- Chart displaying the stock history for the last twelve months

10.2.7 Stock Yearly History Chart

Chart showing the stock’s closing prices for the last fifty-two weeks.

10.3 Systems with Which to Interface

10.3.1 Quote System

System that provides market quotes based on trading symbols.

10.3.2 Market Trading System

System that performs the trade orders requested by RU e-st and communicates trade transaction results back to RU e-st.

10.3.3 News System

System that provides news headlines linked to a particular trading symbol.

10.3.4 Financial Network System

System that provides the ability to transfer funds between RU e-st accounts and accounts external to RU e-st such as bank accounts or credit card accounts.

10.3.5 IRS Reporting System

System that accepts electronic reports of tax information.

11. Licensing Requirements

12. Legal, Copyright and Other Notices

TBD – Contracts and Legal Department

13. Applicable Standards

13.1 NASD, SIPC rules

The application must conform to all the rules of the NASD, SIPC and any other securities trading institutions that interface with the RU e-st system.

The application must conform to the rules of the Federal Trade Commission (FTC) and the Securities and Exchange Commission (SEC).