



IBM Software Group

Non-Functional Requirements

Peter Eeles
peter.eeles@uk.ibm.com

Rational software



@business on demand software

Agenda

- Definitions
- Types of requirement
- Classifying requirements
- Capturing NFRs
- Summary



Definitions

- **Functional Requirement**
 - ▶ Functional requirements describe the behaviors (functions or services) of the system that support user goals, tasks or activities. [Malan]
- **Non-Functional Requirement**
 - ▶ Non-functional requirements include constraints and qualities. [Malan]
 - ▶ [System] qualities are properties or characteristics of the system that its stakeholders care about and hence will affect their degree of satisfaction with the system. [Malan]
 - ▶ A constraint is a restriction on the degree of freedom we have in providing a solution. [Leffingwell]

[Leffingwell] Managing Software Requirements – a Unified Approach, Dean Leffingwell and Don Widrig.

[Malan] Defining Non-Functional Requirements, Ruth Malan and Dana Bredemeyer.



Agenda

- Definitions
- Types of requirement
- Classifying requirements
- Capturing NFRs
- Summary



Types of Requirement

- Use Cases
 - ▶ Defines the behavior of the system from an external perspective
- System-Wide Requirements
 - ▶ Legal and regulatory requirements, application standards, qualities that the system exhibits (such as usability, reliability, scalability, performance), operating system and environment requirements, compatibility requirements, and other design and implementation constraints
- Change Cases
 - ▶ Likely future changes to either the system, in terms of its capabilities and properties, or its environment. Although such changes may not be accommodated in the initial release of the system, they may impact future releases, and therefore the architecture




Examples of Requirements

- The product will support multiple human languages
- The persistence will be handled by a relational database
- The database will be Oracle 8i
- The system will run 7 days a week, 24 hours a day
- An online help system is required
- All presentation logic will be written in Visual Basic



Agenda

- Definitions
- Types of requirement
-  Classifying requirements
- Capturing NFRs
- Summary



Classifying Requirements

- **FURPS**

- ▶ Functionality
- ▶ Usability
- ▶ Reliability
- ▶ Performance
- ▶ Supportability

- **+**

- ▶ Design requirements
- ▶ Implementation requirements
- ▶ Interface requirements
- ▶ Physical requirements



Functional requirements



Non-functional requirements

*The FURPS classification was devised by Robert Grady at Hewlett-Packard



“FURPS+” - Functionality

- All functional requirements
- Usually represent main product features
 - ▶ E.g. Order Processing requirements
- Can also be architecturally significant
 - ▶ Auditing, Licensing, Localization, Mail, Online help, Printing, Reporting, Security, System management, Workflow



“FURPS+”

- Usability
 - ▶ User interface issues such as accessibility, aesthetics and consistency
- Reliability
 - ▶ Availability, accuracy, recoverability
- Performance
 - ▶ Throughput, response time, recovery time, start-up time
- Supportability
 - ▶ Testability, adaptability, maintainability, compatibility, configurability, installability, scalability and localizability



“FURPS₊”

- Design requirement
 - ▶ Constrains the design
 - ▶ E.g. a relational database is required
- Implementation requirement
 - ▶ Constrains the coding or construction
 - ▶ E.g. required standards, platform or implementation language
- Interface requirement
 - ▶ A requirement to interact with an external item
- Physical requirement
 - ▶ A physical constraint imposed on the hardware used to house the system; for example, shape, size and weight



Classifying Requirements

- The product will support multiple human languages
 - ▶ is a supportability requirement
- The persistence will be handled by a relational database
 - ▶ is a design requirement
- The database will be Oracle 8i
 - ▶ is an implementation requirement
- The system will run 7 days a week, 24 hours a day
 - ▶ is a reliability requirement
- An online help system is required
 - ▶ is a functional requirement
- All presentation logic will be written in Visual Basic
 - ▶ is an implementation requirement



Agenda

- Definitions
- Types of requirement
- Classifying requirements
- Capturing NFRs
- Summary



Architectural mechanisms

Analysis Mechanism	Design Mechanism	Implementation Mechanism
Persistence	RDBMS	Oracle
		Ingres
	OODBMS	ObjectStore
Communication	Object Request Broker	Orbix
		VisiBroker
	Message Queue	MSMQ
		MQSeries

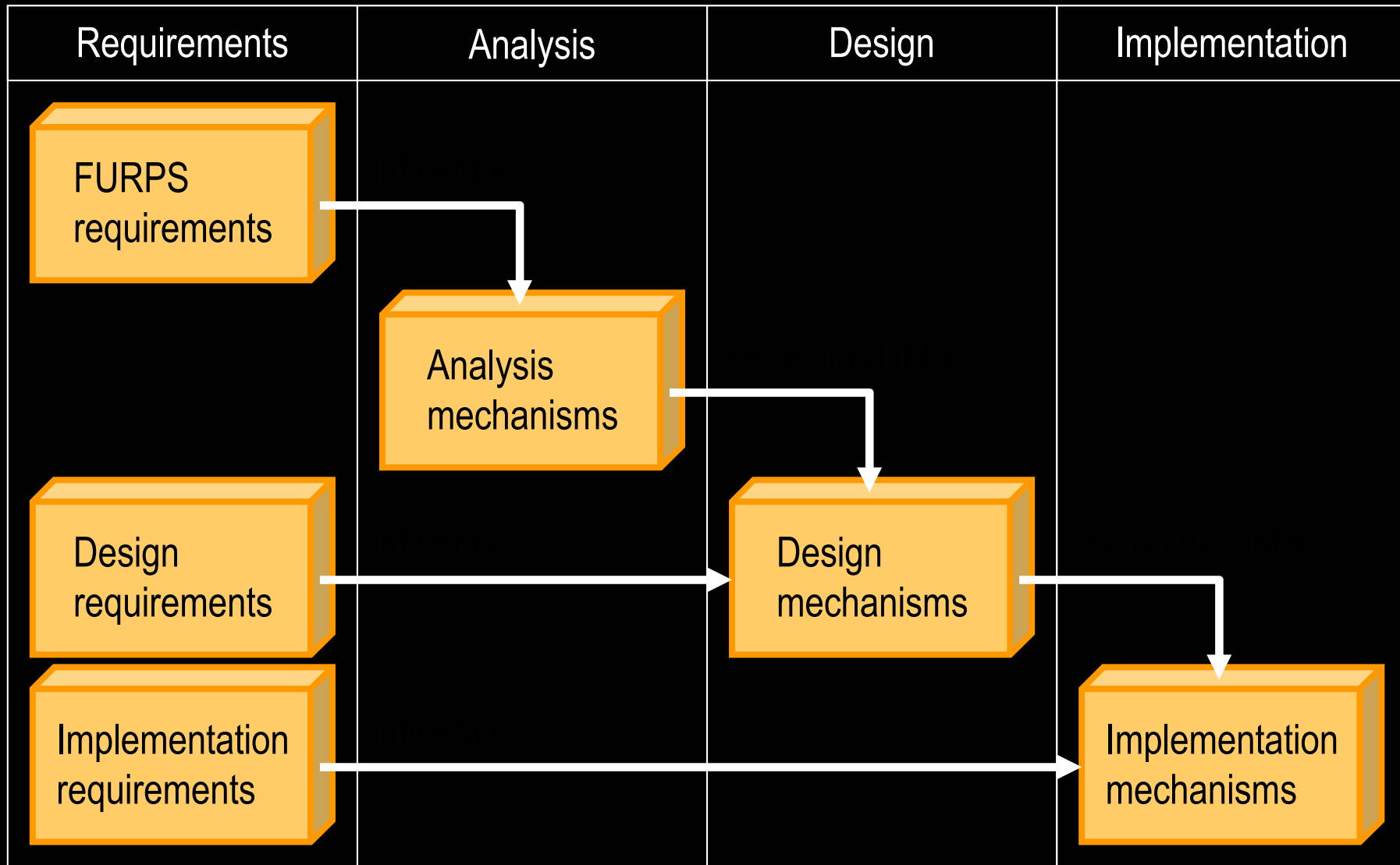


Analysis mechanisms

- Auditing
- Communication
- Debugging
- Error management
- Event management
- File management
- Graphics
- Information exchange
- Licensing
- Localization
- Mail
- Mega-data
- Memory management
- Meta-data
- Online help
- Persistence
- Printing
- Process management
- Reporting
- Resource management
- Scheduling
- Security
- System management
- Time
- Transaction management
- Workflow



Requirements and mechanisms



The NFR dichotomy

- NFRs are difficult to gather
 - ▶ Domain-specific requirements typically more visible
 - ▶ NFRs are unfamiliar to stakeholders
 - ▶ Few techniques for gathering NFRs
- Yet NFRs drive the foundations of our system (the architecture)
 - ▶ Often relevant in a system-wide context
 - ▶ Can be more significant than domain-specific requirements
 - ▶ Consider the availability (“up time”) of a life support machine



Eliciting NFRs

1. Maintain a complete list of NFRs
2. For each NFR, formulate one or more questions
3. Give visibility of the impact of answering a question one way or another
4. Capture the responses to each of the questions
5. Give each NFR a priority or weighting



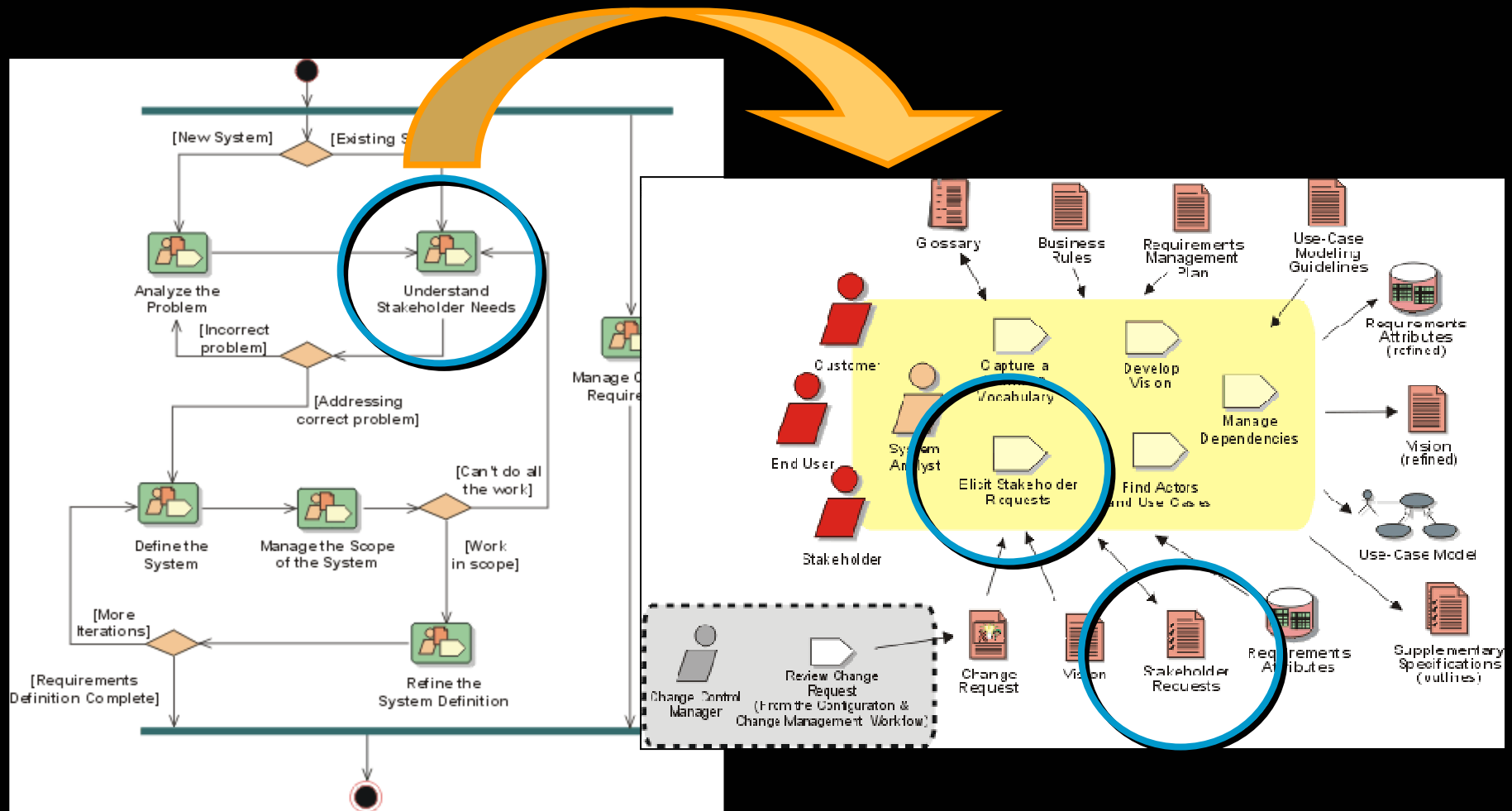
The NFR questionnaire

NFR	Questions	Impact	Answers	Priority
Availability	Are there any requirements regarding system "up time"? This may be specified in terms of Mean Time Between Failures (MTBF).	The higher the availability, the longer the time to market.	Availability is a key product feature. The product must have a MTBF of 60 days.	High

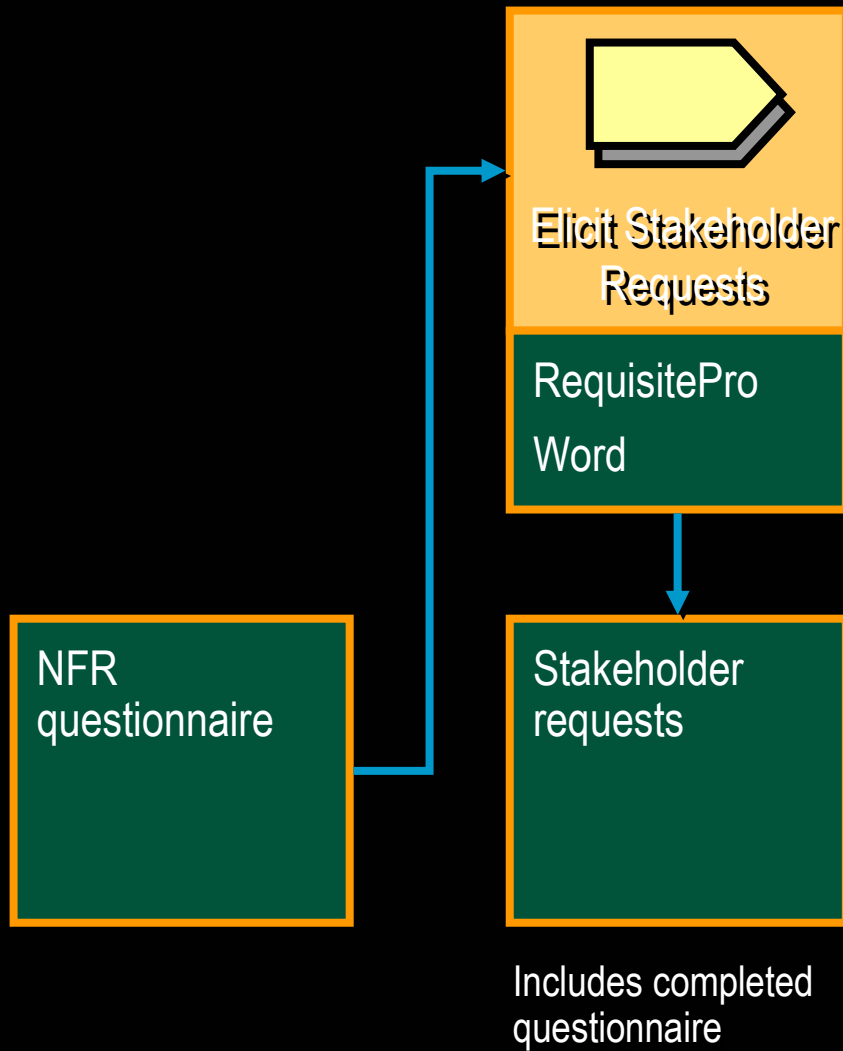


The questionnaire and RUP

- Requirements workflow



Activity - Elicit stakeholder requests



Activity - Elicit stakeholder requests

- NFR questionnaire in RequisitePro

Rational RequisitePro Views - [ASTRQ: Architectural Requirement By Role]

File View Requirement Window Help

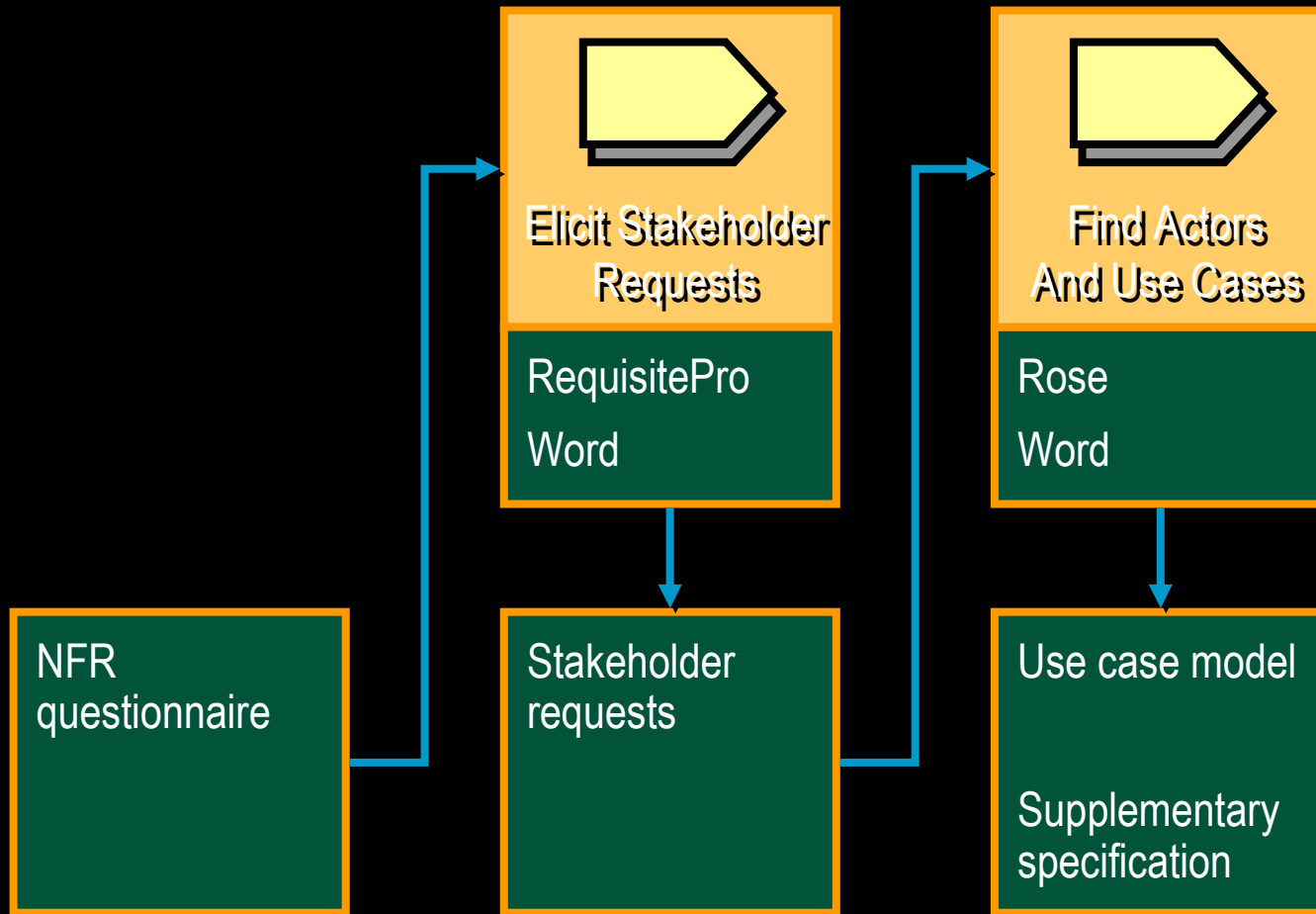
Requirements:	FURPS+ category	Question	Impact	Answer	Priority	Role
[-] ASTRQ1: Auditing	Functional	Is audit capability required?	The need for this capability will increase the time to market, and increase the amount of maintenance.	Yes.	High	Software Architect
ASTRQ1.1: Scope	Supportability	What level of auditing is needed?	The greater the sophistication of this capability, the longer the time to market, and the greater the long-term maintenance cost.	It must be possible to provide audit trails from the time of placing the order, to the shipping of goods from the warehouse.	Medium	Software Architect
ASTRQ1.2: Constraints	Implementation Requirement	Are there any constraints on the auditing implementation?	Use of an existing and understood implementation can improve time to market.	No.	N/A	Software Architect
[+] ASTRQ2: Licensing	Functional	Will the system, or parts of it, be licensed?	The need for this capability will increase the time to market, and increase the amount of maintenance.	Yes.	Medium	Marketeer
ASTRQ2.1: Scope	Supportability	What level of licensing is needed?	The greater the sophistication of this capability, the longer the time to market, and the greater the long-term maintenance cost.	It must be possible to license "value add" capability (such as online order status reporting) separately.	Medium	Marketeer
ASTRQ2.2: Constraints	Implementation Requirement	Are there any constraints on the licensing implementation?	Use of an existing and understood implementation can improve time to market.	Yes. The corporate policy is to license all software.	Low	Marketeer

ASTRQ2: Licensing

Ready | 11 requirements



Activity - Find actors and use cases



Includes completed questionnaire



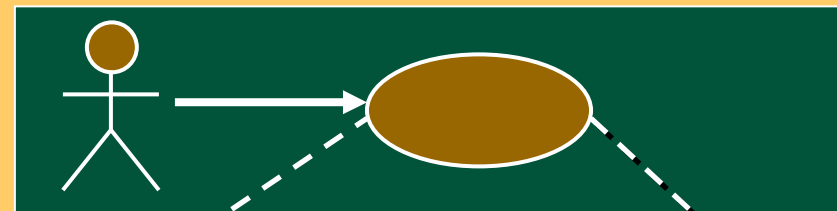
Activity - Find actors and use cases

Supplementary Specification

Use-Case Model

Table of Contents

- 1. Introduction
- 2. Functionality
 - 2.1 <Functional Requirement One>
- 3. Usability
 - 3.1 <Usability Requirement One>
- 4. Reliability
 - 4.1 <Reliability Requirement One>



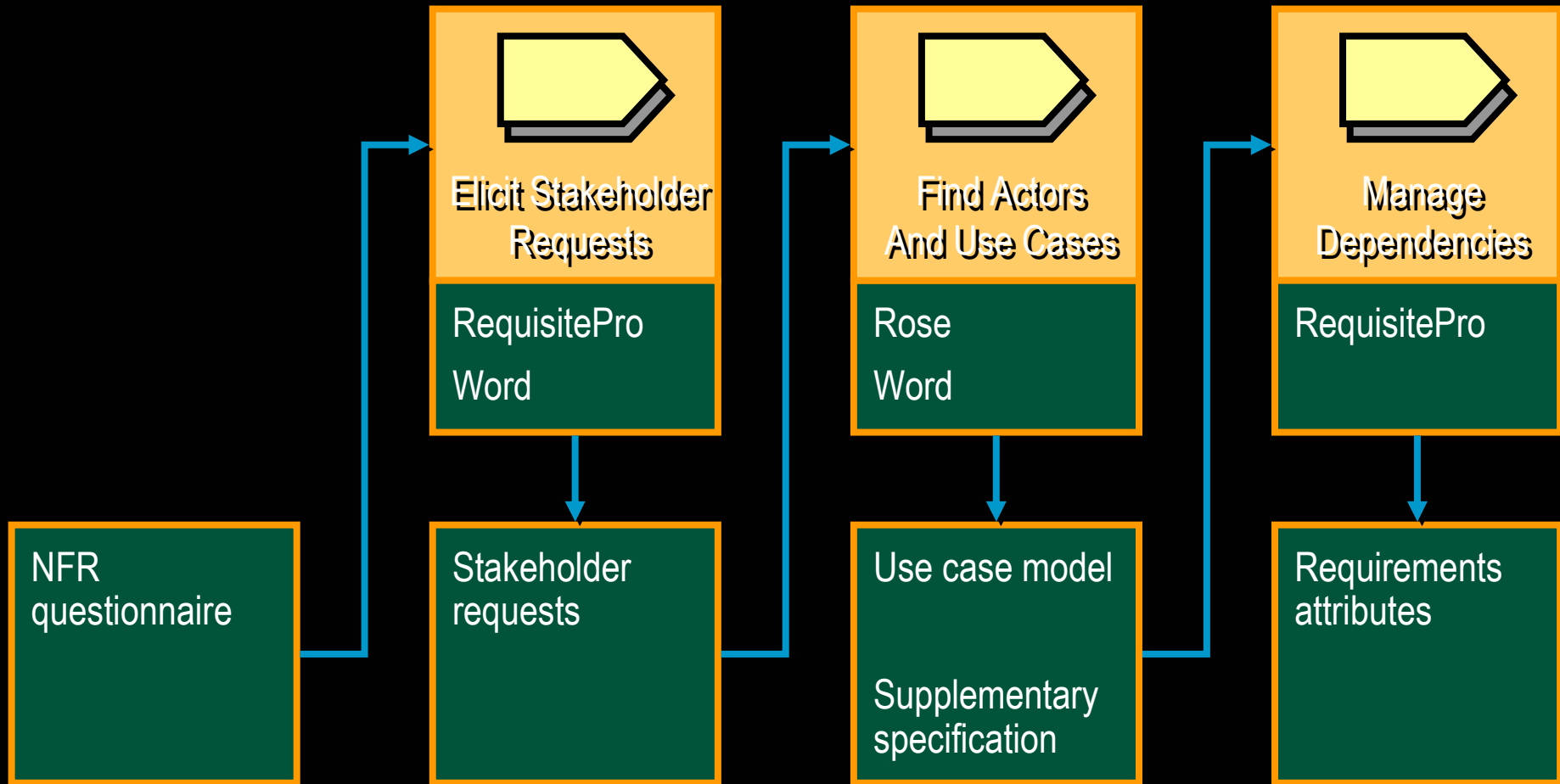
Use-Case Specification

3. Special Requirements

[A special requirement is typically a nonfunctional requirement easily or naturally specified in the text of the use case's include legal and regulatory requirements, application . built including usability, reliability, performance or sup requirements—such as operating systems and environm constraints—should be captured in this section.]

3.1 < First Special Requirement >

Activity - Manage dependencies

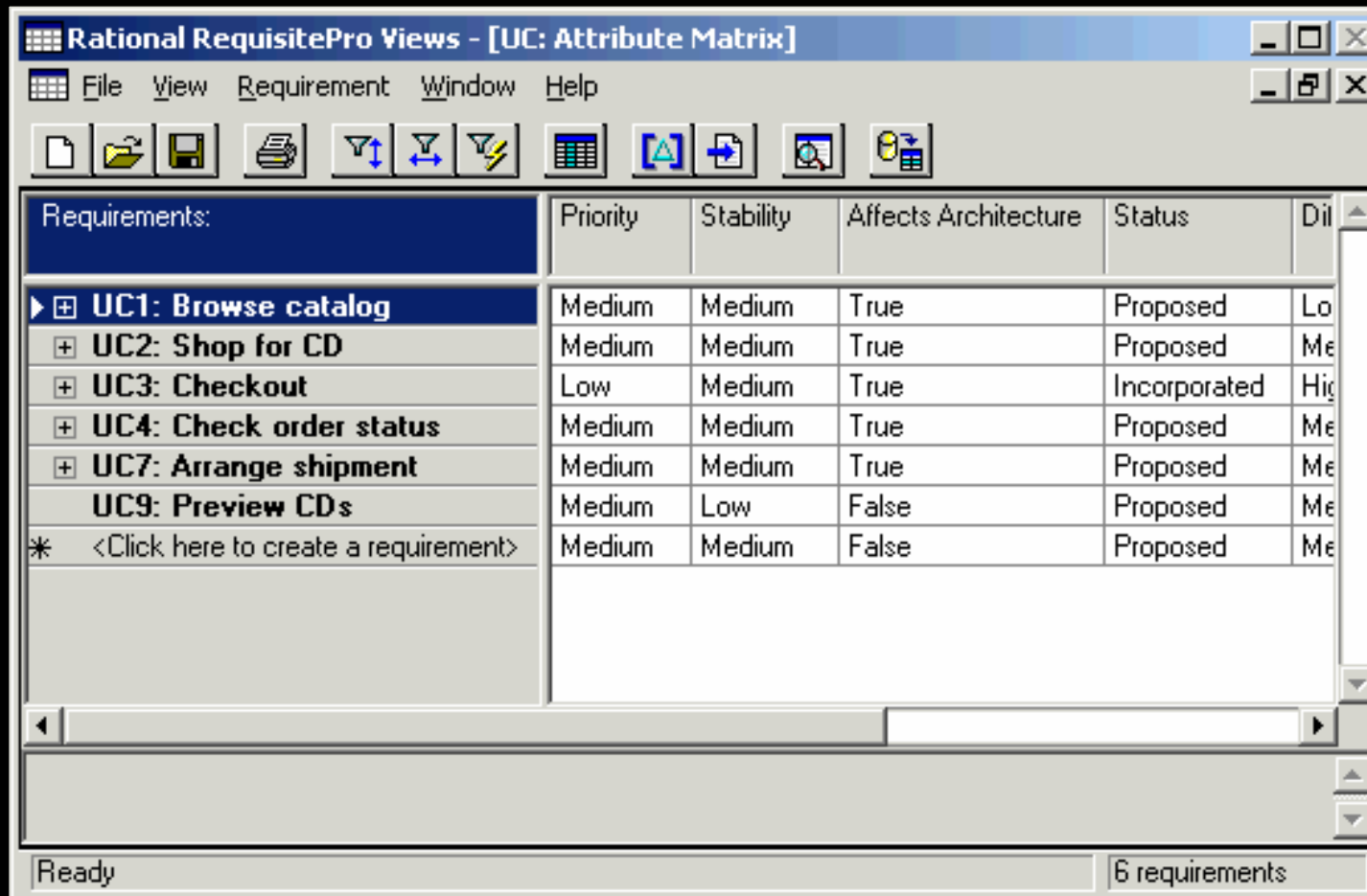


Includes completed questionnaire



Activity - Manage dependencies

- Specify requirements attributes in RequisitePro
 - ▶ E.g. risk, priority, stability



The screenshot shows the Rational RequisitePro Views - [UC: Attribute Matrix] window. The window has a menu bar (File, View, Requirement, Window, Help) and a toolbar with various icons. The main area displays a table of requirements with the following columns: Requirements, Priority, Stability, Affects Architecture, Status, and Dil. The table contains the following data:

Requirements:	Priority	Stability	Affects Architecture	Status	Dil
▶ ⊕ UC1: Browse catalog	Medium	Medium	True	Proposed	Lo
⊕ UC2: Shop for CD	Medium	Medium	True	Proposed	Me
⊕ UC3: Checkout	Low	Medium	True	Incorporated	Hig
⊕ UC4: Check order status	Medium	Medium	True	Proposed	Me
⊕ UC7: Arrange shipment	Medium	Medium	True	Proposed	Me
UC9: Preview CDs	Medium	Low	False	Proposed	Me
* <Click here to create a requirement>	Medium	Medium	False	Proposed	Me

The status bar at the bottom of the window shows "Ready" and "6 requirements".

Common pitfalls

- The “shopping cart” mentality

Analyst: "Does the product need support multiple human languages"?

Stakeholder: "That sounds good. We should plan to address foreign markets"

Analyst: "And what about security?"

Stakeholder: "Oh yes, the product should be secure"

Analyst: "Tell me about your reliability expectations"

Stakeholder: "Definitely 24 by 7 - no down time. That'll show our competitors"

- ▶ Ensure stakeholders understand the “cost” of their purchases



Common pitfalls (2)

- The NFR Questionnaire is technical
 - ▶ Ensure stakeholders understand the value of the questionnaire
- All requirements are equal
 - ▶ Prioritize requirements
- The requirement “parking lot”
 - ▶ Ensure that the requirements are used throughout development



Common pitfalls (3)

- Requirements are not measurable
 - ▶ Ensure that requirements are unambiguous and as measurable as possible
- Lack of time
 - ▶ Constantly remind stakeholders of the importance of these requirements
- Lack of ownership
 - ▶ Ensure that the System Analyst actively creates/tailors and understands the questionnaire
- Talking to the wrong people
 - ▶ Identify the type of stakeholder responsible for answering each question



Common pitfalls (4)

- Requirements are too general
 - ▶ Be as specific as possible

	Scope	Example	Location in a RUP artifact
Most general ↑ ↓ Most specific	The system as a whole	Due to the nature of our target markets, the system must be deployed in English, French, Chinese and Arabic.	Supplementary specification.
	A use case as a whole	Any order that is processed can contain up to 10,000 items.	“Special requirements” section in a use case specification.
	A particular flow of events within a use case	If in the basic flow, the plane undercarriage fails to engage, then an alarm will be sent to the central monitoring station in less than one second.	A “flow of events” section in a use case specification.

Realizing NFRs

- Each realization is very NFR-dependent
 - ▶ Realizing availability is very different to realizing usability
- Can represent an NFR as a UML class
 - ▶ Supported in RSA
- Can represent an “NFR Realization” as a UML collaboration
 - ▶ Supported in RSA



Summary

- Requirements can be classified using “FURPS+”
- Understanding the role of architectural mechanisms can help determine the questions to be asked
- An NFR Questionnaire can help ensure that requirement gathering is systematic
- Gathering stakeholder requests can be automated
- Avoid the common pitfalls!

