Non-Functional Requirements

Peter Eeles
peter.eeles@uk.ibm.com
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]

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

*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

<table>
<thead>
<tr>
<th>Analysis Mechanism</th>
<th>Design Mechanism</th>
<th>Implementation Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>RDBMS</td>
<td>Oracle</td>
</tr>
<tr>
<td></td>
<td>OODBMS</td>
<td>Ingres</td>
</tr>
<tr>
<td>Communication</td>
<td>Object Request Broker</td>
<td>ObjectStore</td>
</tr>
<tr>
<td></td>
<td>Message Queue</td>
<td>Orbix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VisiBroker</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSMQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MQSeries</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td>Analysis mechanisms</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>requirements</td>
<td></td>
<td>Design mechanisms</td>
<td></td>
</tr>
<tr>
<td>Implementation requirements</td>
<td></td>
<td></td>
<td>Implementation mechanisms</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>NFR</th>
<th>Questions</th>
<th>Impact</th>
<th>Answers</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>Are there any requirements regarding system &quot;up time&quot;? This may be specified in terms of Mean Time Between Failures (MTBF).</td>
<td>The higher the availability, the longer the time to market.</td>
<td>Availability is a key product feature. The product must have a MTBF of 60 days.</td>
<td>High</td>
</tr>
</tbody>
</table>
The questionnaire and RUP

- Requirements workflow
Activity - Elicit stakeholder requests

- Elicit Stakeholder Requests
  - RequisitePro
  - Word

NFR questionnaire

Stakeholder requests

Includes completed questionnaire
**Activity - Elicit stakeholder requests**

- **NFR questionnaire in RequisitePro**

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Question</th>
<th>Impact</th>
<th>Answer</th>
<th>Priority</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTRQ1 Auditing</td>
<td>Functional</td>
<td>Is such capability required?</td>
<td>The need for this capability will increase the time to market, and increase the amount of maintenance.</td>
<td>Yes</td>
<td>High</td>
<td>Software Architect</td>
</tr>
<tr>
<td>ASTRQ1.1 Scope</td>
<td>Supportability</td>
<td>What level of auditing is needed?</td>
<td>The greater the sophistication of this capability, the longer the time to market, and the greater the long-term maintenance cost.</td>
<td>No</td>
<td>Medium</td>
<td>Software Architect</td>
</tr>
<tr>
<td>ASTRQ1.2 Constraints</td>
<td>Implementation Requirement</td>
<td>Are there any constraints on the auditing implementation?</td>
<td>Use of an existing and understood implementation can improve time to market.</td>
<td>No</td>
<td>N/A</td>
<td>Software Architect</td>
</tr>
<tr>
<td>ASTRQ2 Licensing</td>
<td>Functional</td>
<td>Will the system, or partly the system, be</td>
<td>The need for this capability will increase the time to market, and increase the amount of maintenance.</td>
<td>Yes</td>
<td>Medium</td>
<td>Marketer</td>
</tr>
<tr>
<td>ASTRQ2.1 Scope</td>
<td>Supportability</td>
<td>What level of licensing is needed?</td>
<td>The greater the sophistication of this capability, the longer the time to market, and the greater the long-term maintenance cost.</td>
<td>It must be possible to license &quot;value add&quot; capability (such as online order status reporting) separately.</td>
<td>Medium</td>
<td>Marketer</td>
</tr>
</tbody>
</table>

Ready 11 requirements
Activity - Find actors and use cases

Elicit Stakeholder Requests
- RequisitePro
- Word

Find Actors And Use Cases
- Rose
- Word

NFR questionnaire
Stakeholder requests
Includes completed questionnaire
Use case model
Supplementary specification
Activity - Find actors and use cases

Supplementary Specification

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 Model

3. Special Requirements
   [A special requirement is typically a nonfunctional requ-
   3.1 < First Special Requirement >
Activity - Manage dependencies

- Elicit Stakeholder Requests
  - RequisitePro
  - Word
  - Stakeholder requests
  - Includes completed questionnaire

- Find Actors And Use Cases
  - Rose
  - Word
  - Use case model
    - Supplementary specification

- Manage Dependencies
  - RequisitePro
  - Requirements attributes
Activity - Manage dependencies

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

<table>
<thead>
<tr>
<th>Scope</th>
<th>Example</th>
<th>Location in a RUP artifact</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system as a whole</td>
<td>Due to the nature of our target markets, the system must be deployed in English, French, Chinese and Arabic.</td>
<td>Supplementary specification.</td>
</tr>
<tr>
<td>A use case as a whole</td>
<td>Any order that is processed can contain up to 10,000 items.</td>
<td>“Special requirements” section in a use case specification.</td>
</tr>
<tr>
<td>A particular flow of events within a use case</td>
<td>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.</td>
<td>A “flow of events” section in a use case specification.</td>
</tr>
</tbody>
</table>
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!