Requirements Engineering [4]

• What are requirements?
• How to produce a requirements specification.

Classification I

• User Requirements
  • abstract
  • natural language + simple diagrams
  • what users want

• System Requirements
  • more concrete, detailed
  • natural + formal language + diagrams
  • what system provides
  • used as contract or product description

Requirements are about WHAT, not HOW.

Classification II

• Functional Requirements
  what the system should do
  • Product

• Non-functional Requirements
  constraints on the system
  • Process
  constraints on the development process

A restaurant example

• User Requirements
  • food should look good
  • taste good
  • enough
  • fish

• System Requirements
  • chicken soup
  • baked salmon with …
  • chocolate mousse

"contract"
Problems and solutions

• Unclear terminology
  • Example: a life insurance policy is closed

• Glossary dictionary of all "technical" terms

Sales: If it’s sold and paid for
Accounting: When it has been paid back

Problems and solutions

• Vague, untestable requirements

• Amalgamation, lack of organisation

• A test scenario must be included
  • Numbered items
  • Standard formats 4.3.2
  • Traceability

Problems and solutions

• Things that are not required occur

• Rationale must be included

• Traceable to stakeholder

• Mandatory/Desired

• Formal model

• User manual

• Use cases

Problems and solutions

• Incomplete

• Rationale must be included

• Traceable to stakeholder

• Mandatory/Desired

• Formal model

• User manual

• Use cases

Traceability

Stakeholder → URD → SRS → Test → Design → User manual
CASE tool (database + ...) for requirements.
CASE = Computer Aided Software Engineering

What’s in the requirements document and who should read it?
How do we get the requirements right?

- Interview with end-user
- use cases
- paper prototype
- prototype
- comparison with similar systems
- user manual

Domain understanding

Requirements collection

Classification

Conflict detection

Validation

Requirements checklist

- Understandable (properly explained)
- Validity (is this required, rationale)
- Verifiable (test cases included)
- Realism (feasibility, prototype)
- Complete (all cases covered, exceptions)
- Consistent
- Traceability