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.

Figure 4.1 User and system requirements

User requirement definition:
1. The HMC PBS staff generate monthly management reports showing the cost of drugs prescribed by each clinician during that month.

System requirements specification:
1.1 On the last working day of each month, a summary of the drugs prescribed, their cost, and the prescribing clinician shall be generated.
1.2 The system shall automatically generate the report for printing after 10:00 on the last working day of the month.
1.3 A report shall be created for each clinician and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed, and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g., 10mg, 20mg, etc.), separate reports shall be created for each dose unit.
1.5 Access to all user reports shall be limited to authorized users based on a management access control list.

A restaurant example

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

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

“contract”

Classification II

• Functional Requirements
  • what the system should do
  • Product

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

Figure 4.3 Types of nonfunctional requirement
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
• Unclear terminology
  • Numbered items
  • Standard formats 4.3.2
  • Traceability

• Vague, untestable requirements
  
• Amalgamation, lack of organisation

• A test scenario must be included

Problems and solutions
• Things that are not required occur
  • Rationale must be included
  
• Traceable to stakeholder
  • Mandatory/Desired

• Incomplete
  • 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

Requirements checklist

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