

REQUIREMENTS

- A *requirement* is a property of the system being constructed
- It is generated by the customer in conjunction with the system engineer or analyst
- Requirements should focus on *what* the proposed system is intended to do, not *how* it will do it

SCOPE

- Requirements should deal with both functional and non-functional concerns
- Requirements should specify what is required and what is or isn't allowed
- Requirements should state what to expect for valid and invalid input

PROPERTIES OF REQUIREMENTS

- Requirements should be simple not compound
- Requirements should be testable
 - Else they are *objectives*
- Requirements should be organized
 - Related requirements grouped
 - Abstract requirements containing detailed requirements
 - Priorities indicated
- Requirements should be numbered
 - So that they can be *traced*
- Requirements should be descriptive (*what*) not prescriptive (*how*)

REQUIREMENTS PROCESS

- Elicitation / gathering
- Refinement
- Analysis and modeling
- Verification
 - Consistency, completeness, etc.
- Validation
 - Rapid prototyping, feedback from customers
- Specification
- Management
 - Traceability, change management

SOURCES OF REQUIREMENTS

- *Stakeholders*
- Customers
- Potential users
- Consultants
- Lawyers
- Developers
- Regulators
- Automation of human tasks
- Analysis and modeling
- Standards
- Product development constraints
- Request for proposals (RFPs)
- Earlier versions
- Competitor products
- Ethnography / temporary assignment
- Business plans
- Market analysis

Requirements Elicitation Techniques

- Interview / meeting
- Survey / questionnaire
- Observation
- Domain analysis
- Prototypes
- Scenarios / user stories
- Embarrassing questions
- Brainstorming

CAVEATS

- Customers do not know what they want. It is the analyst's job to *elicit* requirements and validate them with the customer
- There is a natural tendency for requirements to change over time
- The process of designing and building a system inherently raises new requirements

REQUIREMENTS DOCUMENT (SRS)

- Introduction - purpose, context, objectives
- Analyzed requirements
- System model
- Files (BNF)
- User interface - presentation, dialog
- Errors - situation, message, remedy
- Glossary - definitions, acronyms
- Unprocessed user needs (appendix)

USES OF A SOFTWARE REQUIREMENTS DOCUMENT

- Design and development of software
- User's manual
- Acceptance test plan
- Marketing
- Management

REQUIREMENTS VERIFICATION

- Correctness
 - Accurately reflects needs
- Completeness
 - No missing pieces
- Consistency
 - Absence of conflicts
- Clarity
 - Absence of ambiguity
- Coherence
 - Singleness of purpose
- Feasibility
 - Capable of being accomplished
- Testability
- Traceability
 - Throughout the life cycle
- *What not how*
- Modularity / organized
- Needed by the customer

OTHER PROBLEMS WITH REQUIREMENTS

- Forward reference
 - Mention of a feature before it is defined
- Noise
 - Text that contains no relevant information; redundancy, remorse
- Wishful thinking
 - A requirement that cannot be validated

NON-FUNCTIONAL REQUIREMENTS

- Performance - efficiency, response time, load
- Resource utilization - memory, disk, ...
- Accuracy - for numerical calculations
- Development approach - methodology, language
- Environment - hardware, operating system
- Documentation
- Configurations - options, subsets, binary / source
- Installation
- Cost and schedule
- Acceptance criteria

ILITIES

- Integrity - loss of information
- Security - controlled access to information
- Reliability / availability - mean time to failure; mean time to repair
- Portability - to other operating systems, programming languages, libraries, hardware
- Maintainability
- Reusability

ANALYSIS TECHNIQUES

- Goal hierarchies
- Use cases/scenarios
- Context diagrams
- Formal modeling
- Simulation
- Object-oriented analysis

GOAL HIERARCHY

- Nodes denote customer goals
(achievement, maintenance, defensive)
- Levels indicate the goal/subgoal hierarchy
- Arc types indicate ordering constraints
 - Sequential (this goal must be accomplished before this goal)
 - Interleaved/parallel - may be achieved concurrently
 - Alternative (successful completion of any of the subgoals satisfies the parent goal)

USE CASE / SCENARIO

- Narrative description of intended system behavior from the user's point of view
- Illustrating the accomplishment of a specific goal
- Format: actor/action/subject
 - actor: provides stimulus
 - action: system response
 - subject: item acted upon by action
- Goal accomplishment may be blocked by obstacles

TYPES OF MODELS

- Data models - ER, object-oriented
- State machines / State charts
- Data flow diagrams
- Formal models
- ...

Some Samples:

- As Written

Software will not be loaded from unknown sources onto the system without first having the software tested and approved

- Better

3.2.5.2 Software shall be loaded onto the operational system only after it has been tested and approved

- Better

3.2.5.2 Software shall be loaded onto the operational system only after it has been tested to be in accordance with MIL-SPEC 3425 and approved by the Change Control Board (CCB)

Another Sample

- *3.4.6.3 The system shall*
 - *Prevent the processing of duplicate electronic files by checking a new SDATE record. An email message shall be sent upon occurrence*
- *3.4.6.3 The system shall*
 - a. *Prevent processing of duplicate electronic files by checking the date and time of submission*
 - b. *Send the following email message:*
 1. *Request updated submission of date and time, if necessary, or*
 2. *That the processing was successful when successful*