Requirements Validation & Verification

Requirements Engineering
Objectives

In this chapter, you will learn about:

- Criteria for validating requirements document
- Validation Techniques
- Verification Techniques
Recall the Requirements document consists of two key sections:

1. **Stakeholders Requirements Definitions** ("Business Requirements")
   - Describes what we are to deliver from the customer’s perspective

2. **Requirements Specification**
   - Provides details to the software designers/developers
   - Specifies what needs to be built
Requirements Definitions & Requirements Specifications

- Environment
- Common Interface
- System
- Requirements Defn
- Specification
Requirements Validation & Verification

Prior to submitting your RE document to designers, it is mandatory that

- The customer understands/knows our intents (i.e., Validate requirements definitions)
- Our intents are captured in the requirements document (i.e., Verify the requirements specifications)

Requirements Validation:

- Check that our requirements definitions accurately reflect all of the stakeholders’ needs (i.e., we build the system right)

Requirements Verification

- Verify that the requirements specification conforms to the requirements definition (i.e., we build the right system)
Desirable characteristics to check for include:

- **Are the Requirements correct?**
  - Ensure common understanding with the customer/stakeholders of the requirements definitions

- **Are the requirements consistent?**
  - Ensure there are no conflicting requirements

- **Are the requirements unambiguous?**
  - Multiple readers (reviewers) of the document should not walk away with different but valid interpretations of the document

- **Are the requirements complete?**
  - The requirements is considered to be complete if it specifies required behavior and output for all possible inputs in all possible states
Desirable characteristics to check for

- Are the requirements feasible?
  - Does a solution to the customer needs exist?
- Is every requirement relevant?
  - Check if the requirements include functions that are unrelated to the customers’ needs
- Are the requirements testable?
- Are the requirements traceable?
  - Requirements are organized and uniquely labeled (enumerated) for reference
  - Every entry in the requirements definition has a corresponding entry in the requirements specification, and vice versa
Validation Techniques

- **Reading**
  - Each stakeholder simply reads the document then reports errors and/or provides comments
  - The team signs off on the document after the corrections/comments are incorporated in the document
    - Team takes responsibility for subsequent errors found in the document

- **Walkthroughs**
  - One of the RE document authors presents the requirements to the stakeholders and ask for feedback
    - Useful for large number of varied stakeholders
Validation Techniques (continued)

- **Formal Inspection**
  - Reviewers take specific roles:
    - Presenter
    - Moderator
    - Scribe
  - Reviewers follow prescribed rules:
    - How to examine the requirements
    - When to meet
    - When to take breaks
    - Follow-up inspections
Review

- Requires representatives from the customers and the SDLC team to examine the document individually and then meet to discuss identified problems

- Customers’ team includes:
  - Employees who will operate the system
  - Employees who will prepare the systems’ input and those who will use the systems output respectively
  - Key managers of the employees

- SDLC team includes:
  - Systems architects
  - Software Designers
  - Test Team
  - Process Engineer
Requirements Validation
Review (continued)

- Review stated goals and objectives
- Compare requirements with goals and objectives
  - Isolate irrelevant requirements
- Customer’s rep review the flows, use-cases, MSCs to make sure they accurately customer’s needs
- SDLC team reviews the propose functions and constraints to confirm that they realistic (i.e., resources)
- Assess and document any risks (development/functioning) and agree on alternative approaches
- Discuss about testing the system
Requirements Validation
Other Techniques

- **Certain Validation Techniques can be automated**
  - Tools to check for consistency and completeness
  - Tools to check for traceability
- Interviews
- Checklists
- Models to check functions and relationships
- Scenarios
- Prototypes
- Simulations
Requirements Verification
Techniques

Our objective is to check that the requirement specification document corresponds to our requirements definition document

- Check that each requirement in the definition/stakeholders document is traceable to the requirements specifications
  - Cross-referencing
  - Simulation
  - Consistency checks
Case Study - 1
Systems Verification

- Analysis of Systems Verification Test Plans for
  - Number Portability Requirements Document
    - Stakeholders Requirements
    - Requirements Specification
Case Study - 2
Requirements Engineering

- Analysis of RE Document for
  - Autonomous Control of a Robot for rescue operations, Household Chores etc
    - http://www.youtube.com
    - Command control
    - Tongue Movement control
    - Joystick control
  - Hands free control of Wheelchair
    - http://www.youtube.com
Guidelines for Writing Requirements

Stable Requirements

- **Traceability**
  - Is the origin of your requirements clear and well understood
  - Can a part of the code (or change to the code) be identified with a unique set or sets of requirements
  - Alternatively, can a requirement (or change to a requirement) be traced to a unique set of code?

- **Ambiguity**
  - Can any of your requirements be interpreted in more than one way?

- **Open-Ended**
  - Do the requirements contain phrases such as “will support at least 20 online customers…”?
Guidelines for Writing Requirements

Stable Requirements

- Completeness
  - Are all expected and unexpected responses to an input defined?
    - Do not assume the sunny-day scenario
  - Do the requirements specify how frequently a feature will be used?
  - Do the requirements specify the peak hours of use for this feature?
  - Do the requirements differentiate between usage by occasional users of a feature and usage by frequent users?

- Consistency
  - How will you determine if there are any conflicting requirements?
  - What are the decision criteria for resolving conflicting requirements?
Guidelines for Writing Requirements

Stable Requirements

- Usage Scenarios
  - Is there a complete flow analysis from external stimulus through the system and out to external result?
    - If the answer is no, you may adopt the “divide and conquer” principle
  - An understanding of how the system will be used is essential if the system is to be architected and designed to meet the customer needs

What if usage scenarios are unavailable?
Guidelines for Writing Requirements

Stable Requirements

- **Performance Specification**
  - Are response time, throughput and other performance requirements clearly stated?
  - How were these determined?
  - Are the number and types of users documented?
  - Are transaction rates and/or data volumes known?
    - To determine CPU budgets
    - To determine Network bandwidth

- **Complexity**
  - How complex is the algorithm to implement a specific requirement?
  - Can a simpler algorithm be substituted?
Guidelines for Writing Requirements
Stable Requirements

- External Interfaces
  - Are external interfaces well defined and explicit?

- Error Control/Recovery
  - What provisions have been made to recover from lost transaction?
  - What provisions have been made to detect and recover from network failures
  - What provisions have been made to detect and recover from disk crashes of databases (*Opensong project*)?
Guidelines for Writing Requirements

Stable Requirements

- Operations, Administration and Maintenance (OA&M)
  - It is necessary to understand how (and by whom) the system will be operated
  - It is necessary to understand how (and by whom) the system will be administered
  - It is necessary to understand how (and by whom) the system will be maintained

- Availability/Reliability
  - Do the requirements specify system availability and reliability?
    - Downtime requirements
  - What are the MTBF and MTTR figures needed to meet the customer needs?
    - Note MTBF and MTTR preferred to “7 by 24” or “99.98%”
Guidelines for Writing Requirements

Review Questions

- Usage Scenario
  - Is there an understanding of how the customer will use this system?
- Has the requirements list be put in priority order?
  - What is the minimal list of features to achieve for the minimum first release?
- Are there any “requirements” (specifications) which are basically re-inventing functionality provided by the OS?
- What are the specific user response time requirements per transaction or transaction type?
- How many concurrent users will be accessing the system and what is the normal and abnormal mix of transactions they will be running?
Guidelines for Writing Requirements

Review Questions

- How many concurrent users will be accessing the system and what is the normal and abnormal mix of transactions they will be running?
- What is the OS for your system?
- What are the availability requirements by function/transaction/operation?
  - Is Mean Time Between Failures (MTBF) specified?
  - Is Mean Time to Repair (MTTR) specified?
- Do the requirements specify measurable system acceptance criteria?