Requirements Validation and Negotiation
AGENDA

- Fundamentals of Requirements Validation
- Fundamentals of Requirements Negotiation
- Quality Aspects of Requirements
- Principles of Requirements Validation
- Requirements Validation Techniques
- Requirements Negotiation
REQUIREMENTS VALIDATION AND NEGOTIATION

Fundamentals of Requirements Validation
Purpose of Requirements Validation

- Reviewing requirements in order to discover errors or quality problems
- Early assurance that (documented) requirements:
  - represent the actual needs and expectations of the stakeholders
  - have the necessary level of quality
  - can be approved for further development activities
    - Design, implementation, test, ...
    - Especially important in client–contractor relationships

! Remember: the later an error is found, the more expensive it is to correct
REQUIREMENTS VALIDATION AND NEGOTIATION

Fundamentals of Requirements Negotiation
Purpose of Requirements Negotiation

- Unresolved conflicts reduce system acceptance
  - E.g., when only one of the contradictory requirements is implemented, the other party is demotivated

- Overall goal of negotiation
  - Gain a common and agreed-upon understanding of the requirements among all stakeholders
  - Resolving contradictory requirements of different stakeholders
    - “Shut down in case of failure” vs. “Restart in case of failure”

- Negotiation needs to be done throughout the entire requirements engineering process (not as a single step at the end!)
REQUIREMENTS VALIDATION AND NEGOTIATION

Quality Aspects of Requirements
Relevant Quality Aspects for Validation

- **Content**
  - Have all relevant requirements been elicited and documented at the appropriate level of detail?

- **Documentation**
  - Do all documented requirements respect the predetermined guidelines for documentation and specification?

- **Agreement**
  - Do all stakeholders concur with the documented requirements and have all known conflicts been resolved?

Approval for further development activities should only be given if all three aspects are being fulfilled.
Quality Aspect: Content

Content errors are present when the quality criteria for requirements are violated.

Fulfilled if no shortcoming have been detected with regard to:
- completeness (set of all requirements)
- completeness (individual requirements)
- traceability
- correctness / adequacy
- consistency
- no premature design decisions
- verifiability
- necessity
Quality Aspect: Documentation (1/2)

- Documentation errors are present when specification guidelines have been violated
- Risks of documentation errors
  - Impairment of development activities
    - Format not usable for processing
  - Misunderstanding
    - Requirements consumers do not understand the notation properly
  - Incompleteness
    - Used format does not allow representing all information
  - Overlooking requirements
    - Requirements consumers do not find information where expected
Quality Aspect: Documentation (2/2)

Fulfilled if no shortcoming have been detected with regard to:
- conformity to documentation format (e.g., template, notation)
- conformity to documentations structure (e.g., correct section)
- understandability (e.g., defined terminology)
- unambiguity (e.g., only one possible interpretation)
- conformity to documentation rules (e.g., correct use of notation syntax)
Quality Aspect: Agreement

Agreement errors are present when the set of requirements does not or no longer represent the stakeholders’ actual needs and expectations.

Fulfilled if no shortcoming have been detected with regard to:
- agreement (all requirements agreed upon by the stakeholders)
- agreement after changes
- conflict resolution
REQUIREMENTS VALIDATION AND NEGOTIATION

Principles of Requirements Validation
Essential Validation Principles

1. Involvement of correct stakeholders
2. Separation of the identification and correction of errors
3. Validation from different views
4. Adequate change of document type
5. Construction of development artifacts
6. Repeated validation

- These principles are independent of how the validation is performed
Principle 1: Involvement of Correct Stakeholders

- Which stakeholder is “correct” depends on the goal of the validation
- Validation should never be carried out by the author of a requirement / requirements document
- External vs. internal validation
  - **External**: validators from outside the development team or even the organization (e.g., experts from Fraunhofer)
    - More independent and neutral view
    - Higher effort and costs
  - **Internal**: validators from inside the development team or the organization
    - More influence from prior knowledge
    - Easy to coordinate
Principle 2: Separation of the Identification and Correction of Errors

- The principle of separating error identification from error correction is well-established in today’s software development practice.

- Applied to requirements validation, this separation:
  - enables validators to concentrate on identification
  - enables authors to concentrate on correcting / fixing
  - supports double-checking
Principle 3: Validation from Different Views

- Requirements should always be validated by several people who take the perspectives of different roles, e.g.:
  - the customer (who pays for the system)
  - the user (who uses the system)
  - the tester (who tests the system)
  - the developer (who develops the system)
  - the architect (who designs the system)
  - …

- Taking into account multiple perspectives increases finding more and different errors and reduces finding duplicate errors
Principle 4: Adequate Change of Documentation Type

- Validation uses the strengths of each documentation type (e.g., natural language, graphical models) in order to compensate for their weaknesses.
- Transcribing a requirements from one documentation format into another allows for easier validation of this requirement.
  - E.g., written text on a process flow can be validated easier if a corresponding model is drawn.
Principle 5: Construction of Development Artifacts

- Requirements are used as the basis for further development artifacts
- It makes sense to directly derive these artifacts (e.g., test cases, design sketches) during validation
- Developing artifacts allows for checking whether a requirement is really a suitable basis for development (beyond the assumption that it is)
Principle 6: Repeated Validation

- Validation should therefore be repeated multiple times, because stakeholders gain additional knowledge during a project.
  - A validated requirement may not be valid at a later point in time.
- Typical situations in which validation is crucial:
  - (Many) Innovative ideas
  - Significant gain of knowledge during requirements engineering
  - Long-lasting projects
  - First validation was conducted very early
  - Unknown domain
  - Reuse of requirements
REQUIREMENTS VALIDATION AND NEGOTIATION
Requirements Validation Techniques
Techniques Overview

- **Core techniques** (review techniques)
  - Commenting
  - Inspections
  - Walkthroughs

- **Supporting techniques**
  - Perspective-based reading
  - Prototyping
  - Validation checklists

- Each technique requires upfront preparation (e.g., identification and involvement of correct stakeholders)
Commenting / Ad Hoc Review (One Reviewer)

- Receiving opinions on the quality of the requirements

- Process
  1. The author hands out the document to an expert.
  2. The expert checks for errors and ambiguities
  3. The flaws are marked and briefly explained

- Advantages
  - Easy to apply
  - Low costs

- Disadvantages
  - Unsystematic; results depend on expert’s knowledge
  - No group effects
Inspection

- Finding errors in a systematic way
- Advantages
  - Very effective and efficient
  - Structured process
  - Systematic support through reading techniques
- Disadvantages
  - Costly
  - Learning effort
1. Planning

- Determining the goal of the validation, the work results that are validated, and which roles and participants are included

- Six roles
  - Author → explains the requirements, implements corrections
  - Validators (inspectors) → search for errors, communicate findings
  - Organizer → plans and supervises
  - Moderator → leads the session, follows the inspection process
  - Reader → guides the inspectors through the requirements
  - Minute-taker → reports on the results of the inspection
Inspection Process (2/2)

2. Overview
   - The author explains the requirements to the validators

3. Error detection
   - Validators (inspectors) search through the requirements for errors and document their findings
   - Recommendation: alternate between collaborative and individual work

4. Error collection and consolidation (technical review)
   - Discussing the findings, detecting false positives
   - An error list contains the consolidated errors and correcting measures

5. Correction
   - The author corrects the detected errors
Walkthrough

- Gaining a shared understanding and identifying quality flaws

Process

1. The author hands out requirements to validators prior a workshop
2. Validators discuss the requirements to be validated one by one

Advantages

- Stakeholders gain a better understanding
- Active discussions among stakeholders

Disadvantages

- No formal process
- Risk of the author presenting the requirements too positively
Checklists

- Supporting technique, to be applied in conjunction with review techniques
- Ideal for complex environments in which no aspect may be omitted
- Precise questions ease the detection of errors, according to the:
  - three quality aspects (content, documentation, agreement)
  - principles of validation
  - quality criteria for requirements documents
  - quality criteria for (single) requirements
  - experiences from prior projects
  - error statistics from prior projects
- The checklist should not be longer than one page
- The checklist can range from generic to perspective-specific
Perspective-Based Reading

- Supporting technique, to be applied in conjunction with review techniques (ideally inspections)
- Reviewing the document based on perspective-specific checklists from just one perspective, e.g.:
  - user perspective
  - architect perspective
  - tester perspective
Benefit of Perspective-Based Reading

Non-Perspective-Based Reading

High Overlap, Low Coverage

Document with Errors

Perspective-Based Reading

Low Overlap, High Coverage
Prototypes

- Supporting technique for requirements validation
- Enables stakeholders to experience what they will get
- Are considered the most effective means for validation

- Degree of detail
  - High-fidelity
  - Low-fidelity

- Prototype lifetime
  - Evolutionary
  - Throw-away
Using Prototypes

Prototypes are useful if:

- stakeholders do not have any experience regarding the system to be built or do not know in detail what they really need
- “I know it when I see it” (IKIWISI) phenomenon
- a risk for misunderstandings exists, because:
  - stakeholders are not able to describe their requirements properly
  - requirements engineers do not understand the stakeholders correctly
- a similar product did not exist so far (feasibility unclear)
- the system is interactive
Low-Fidelity Prototypes

- Usually first sketches on paper
  - Intentionally not similar to the final product

Advantages
- Easy to create
- Supports communication
- Change requests can be "implemented" directly

Disadvantages
- No prototyping of functionality
- Throw-away prototype
- No assurance that every concept is technically feasible
Example: Lo-Fi BTB Prototype with Changes
High-Fidelity Prototype

- Usually realistic screens developed using software
  - High similarity to the final system

Advantages
- Functionality can be validated as well
- Users can get a feel what the final system is like

Disadvantages
- Costly development
- Change requests cannot be incorporated directly
- Stakeholders may get the impression that the system is already there
Example: Hi-Fi BTB Prototype

Business Trip Booker 2.0 - New trip

View   Edit   Help
New Trip  Drafts  My Trips  Log out

User: Tom Traveler

Outbound trip
Date
Time
Location (please select)

Inbound trip
Date
Time
Location (please select)

Departure
Arrival

Save as draft  Next
Validation Process with Prototypes

1. Selection of requirements that should be validated through prototypes
2. Selection of a suitable class of prototypes
3. Development of the prototype
4. Preparation of a manual or set of instructions for the users, a prototyping scenario, and a checklist of validation criteria
5. Performing the validation with stakeholders
6. Documentation of the validation results (i.e., own observations and statements of the stakeholders)
7. Analysis of the results and decision about adaptation
8. (Repetition)
REQUIREMENTS VALIDATION AND NEGOTIATION

Requirements Negotiation
Performing Requirements Negotiation

- Negotiation requires systematic conflict management, which includes:
  - conflict identification
  - conflict analysis
  - conflict resolution
  - documentation of the resolution
Conflict Identification

- Conflicts are often not obvious at first glance
- Conflicts need to be identified systematically by:
  - directly analyzing elicited requirements
  - analyzing the requirements while documenting them
  - reviewing the requirements explicitly during validation
Conflict Analysis (1/3)

- To resolve contradicting requirements, the underlying reason for the conflict must be understood.
- Main conflict types
  1. Subject conflict
  2. Conflict of interest
  3. Conflict of value
  4. Relationship conflict
  5. Structural conflict

Note: In practice, most contradictions have multiple underlying reasons (hybrid conflicts). The above classification helps to analyze these reasons.
Conflict Analysis (2/3)

**Subject conflict**
- Stakeholders have different interpretation of the consequences
  - E.g., “Is a response time of 1 second fast enough?”

**Conflict of interest**
- Stakeholders have different goals
  - E.g., “Should the requirements on integrating the system with the central human resources system be rejected due to its implementation costs?”

**Conflict of value**
- Stakeholders have different cultural and personal preferences
  - E.g., “Should the BTB system be developed using open source or commercial components?”
Conflict Analysis (3/3)

- **Relationship conflict**
  - Negative interpersonal behavior between stakeholders of the same organizational rank (→ “power & politics”)
  - E.g., “Should the requirement of the Payroll Director be replaced by the requirement the Human Resources Director?“

- **Structural conflict**
  - Similar to relationship conflict, but between stakeholders on different levels of the hierarchy
  - E.g., “Does the Human Resources Director agrees with the requirements of Tina Travelmanager?”
Conflict Resolution and Documentation

- Conflict resolution is a success factor for a project
  - It can motivate or demotivate stakeholders to cooperate further

- It is essential to involve all relevant stakeholders in conflict resolution
  - Otherwise: some opinions or viewpoints will be neglected

- Conflicts and their resolution must be documented, otherwise:
  - the same conflicts may arise again and need to be handled anew
  - the rationale for the requirement is lost and one group of stakeholders will complain when the system is later in place
Conflict Resolution Techniques

- **Agreement** → one party convinces the other party and both parties agree on the initial requirement

- **Compromise** → The parties find a compromise between the contradicting, alternative requirements or create a new requirement to which all parties agree

- **Voting** → all stakeholder vote for one of the contradicting requirement, and the requirement with most votes is selected

- **Variants** → the conflicting requirements are not resolved, but the system is developed in different variants (or configurations)

- **Overruling** → a party with a higher organizational rank selects one of the conflicting requirements (only if all other techniques have failed)

- ...and some more, e.g., decision matrix, plus-minus-interesting, and consider-all-facts
Summary

- Validation assures the quality of elicited and documented requirements
- Quality aspects: content, documentation, agreement
- Validation techniques
  - Main techniques: commenting, inspections, walkthroughs
  - Supporting: perspective-based reading, prototypes, checklists
- Negotiation of requirement conflicts
  - Identification, analysis, resolution, documentation
Questions