ASE Summary (beliebig lang)

19. September, 2024; rev. 3. November 2024

Linda Riesen (rieselin)

1 Introduction and Overview

1.1 3 Kinds of Rquirements

- **Functional requirements** concern a result or behavior that shall be provided by a function of a system. This includes requirements for data or the interaction of a system with its environment.
- **Quality requirements** pertain to quality concerns that are not covered by functional requirements — for example, performance, availability, security, or reliability.
- **Constraints** are requirements that limit the solution space beyond what is necessary to meet the given functional requirements and quality requirements.

1.2 Goals of RE (Why)

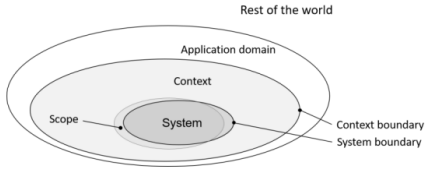
- minimize risk/ failure / costly modifications in later stages
- eases intellectual complexity
- basis for cost & effort calc.
- testing

1.3 Main Tasks

- elicitation
- documentation
- validation
- management

2 Fundamental Principles of RE

1. **Value orientation**: Requirements are a means to an end, not an end in itself (value = benefit - cost)
2. **Stakeholders**: RE is about satisfying the stakeholders' desires and needs, stakeholders have a role (client, user, etc.) & amount of influence (critical, major, minor)
3. **Shared understanding:** Successful systems development is impossible without a common basis (explicit & implicit shared understanding)
    - helpful: glossaries, prototypes, using existing system as reference
    - enablers: Domain knowledge, Domain-specific standards, Previous successful collaboration, Existence of reference systems known by all people involved, Shared culture and values, Informed (not blind!) mutual trust
    - obstacles: Geographic distance, Supplier-customer relationship guided by mutual distrust, Outsourcing, Regulatory constraints, Large and diverse teams, High turnover among the people involved
4. **Context**: Systems cannot be understood in isolation

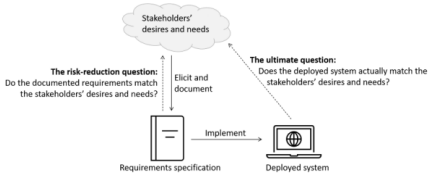


- If changes in the context may occur, how do they impact the requirements for the system?
- Which requirements in the real-world context are relevant for the system to be developed?
- How can such real-world requirements be mapped adequately to requirements for the system?
- Which assumptions about the context must hold such that the system will work properly and the requirements in the real world will be met?

- what rules and laws impact the context

5. **Problem**, requirement, solution: An inevitably intertwined triple, should try to separate will not be completely possible

6. **Validation**: Non-validated requirements are useless

- 

7. **Evolution**: Changing requirements are no accident, but the normal case

8. **Innovation**: More of the same is not enough: Giving stakeholders exactly what they want means missing out on the opportunity of doing things better than before.

9. **Systematic** and disciplined work: We can't do without in RE

3 Work Products and Documentation Practices

3.1 Work Product

- Lifespan (temporary, evolving, durable)
- can be:
  - single requirement (e.g. user story)
  - coherent sets of requirements (e.g. use case, task description epic, feature)
  - documents / documentation structure (e.g. product / sprint backlog, story map, vision)
  - other (e.g. glossary, prototype, note, sketch)
- **Abstraction Level** Requirements naturally occur at different levels of abstraction. Selecting work products that are adequate for a given level of abstraction and properly structuring work products that contain requirements at multiple abstraction levels is helpful.
- consider:
  - structure and data
  - function and flow
  - state and behavior
- Constraints:
  - Technical
  - Legal
  - Cultural
  - Environmental
  - Physical
  - specific for project

3.2 Natural Language

Problematic mostly in written form, could be helped by using (as long as not just syntactic exercise): phrase templates, form templates, and document templates

- Rules:
  - Write short and well-structured sentences
  - Create well-structured work products
  - Define and consistently use a uniform terminology (Glossary)
  - Avoid using vague or ambiguous terms and phrases.
- avoid:
  - Incomplete descriptions
  - Unspecific nouns (the data)

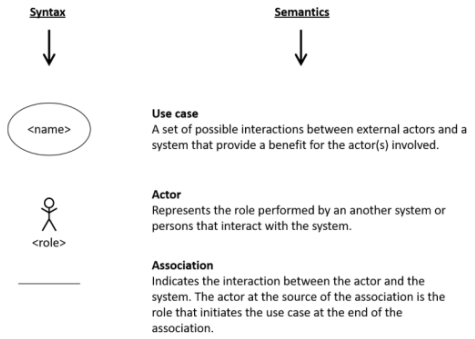


Abbildung 1: Modeling Language Syntax and Semantics

  - Incomplete conditions
  - Incomplete comparisons (much better)

- use with care:
  - Passive voice
  - Universal quantifiers (always, never, etc)
  - Nominalizations (often the verb describes a process that needs additional requirements)

3.3 Model Based Work Products

Model Types

- Modeling System Context: Data Flow Diagram, UML Use Case Diagram
- Modeling Structure and Data: UML Class Diagram
- Modeling Function and Flow: UML Activity Diagram
- Modeling State and Behavior: UML State Diagram
- Supplementary Models: Block Diagram, Domain Story Models, UML Sequence Diagram

Quality Assesment Model

- Syntactic quality
- Semantic quality
- Pragmatic quality

3.4 Glossaries

Creating, maintaining, and using a glossary consistently avoids errors and misunderstandings concerning the terminology used. Working with glossaries is a standard best practice in RE.

3.5 Prototypes

- **Exploratory Prototypes**
  - Clarify or validate requirements; discarded after use.
  - Types:
    - **Wireframes**: Low-fidelity, quick to build, for idea discussion.
    - **Mock-ups**: Medium-fidelity, realistic UI without functionality.
    - **Native Prototypes**: High-fidelity, tests critical functionality.
- **Experimental Prototypes (Breadboards)**: Explore technical feasibility; discarded.
- **Evolutionary Prototypes**: Core system evolves into final product; used in agile.
- **In Requirements Engineering**: Elicitation and validation tool to clarify unclear requirements.
- **Trade-offs**: Fidelity impacts cost vs. value in reducing risks.

3.6 Quality Criteria

Quality Criteria for single Requirements

- Adequate
- Necessary
- Unambiguous
- Complete
- Understandable
- Verifiable

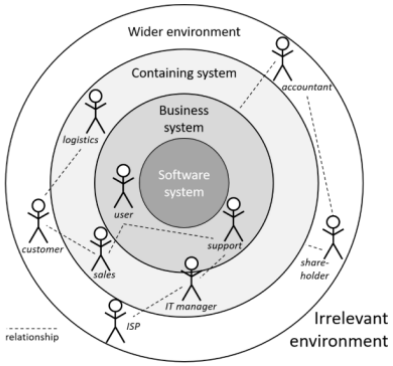
Quality Criteria for Work Products

- Consistent
- Non-redundant
- Complete
- Modifiable
- Traceable
- Conformant (follows mandatory structuring or formatting instructions)

4 Practices for Requirements Elaboration

4.1 Sources for Requirements

- Stakeholders: "Does a relevant relationship exist between the person/organization and the system?-> Create Stakeholder list
  - Stakeholder List: Info, Availability, Influence, Interest...
  - Alexander's Onion Model



  - Special Stakeholder: Internal & External User & Misuser => Personas
- Documents: company-, domain- and project-related documents, product and process descriptions, legal and regulatory documentation
- (Other) systems

4.2 Elicitation of Requirements

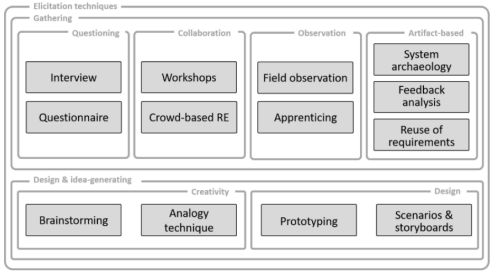


Abbildung 2: Gathering techniques vs. Design and idea-generating techniques

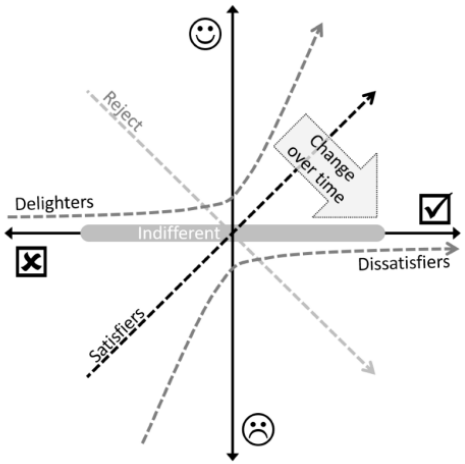


Abbildung 3: Kano Model

4.3 Resolving Conflicts regarding Requirements

If you do not resolve all requirements conflicts that you notice right away, they will pop up later in the development process.

Possible Types of conflict

- subject matter (different factual needs => different indended usage)
- Data conflict (same data, different interpretation)
- Interest conflict
- Value conflict
- Relationship Conflict (past negative experiences)
- Structural conflict (imbalance in power)

4.4 Validation of Requirements

Important Aspects for Validation

- Involving the correct stakeholders
- Separating the identification and the correction of defects
- Validation from different views
- Repeated validation

Validation Techniques

- Review techniques (static):
  - Formal: Walkthrough (explanation step by step), Inspection (by moderator, very formal)
  - Informal: Author-Reviewer Cycle: easy, cheap, and approachable means of validation
- Exploratory techniques (dynamic)
  - **Prototyping**
  - **Elicitation and validation go together:** while validating requirements elicited at an earlier point in time, you will almost certainly detect new requirements in the feedback from the participants
  - **Alpha testing and beta testing:** a fully featured, completely working pre-production version of the system is provided to end users for operation with the intended business processes in a realistic environment. (Alpha in Simulated enviroment, small group, beta in production)
  - **A/B testing:** system is offered to different (mostly randomly selected) groups of users in two variants that differ in design or functionality => measure reactions
- Sample development (static)

5 Process and Working Structure

5.1 Influencing Factors

- Overall process fit: How well does the chosen process fit

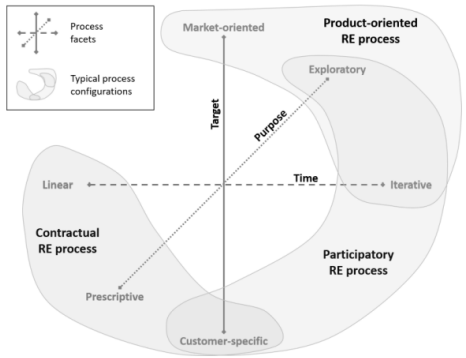


Abbildung 4: Process Facets

- Development context:
  - Customer-supplier-user relationship: Who are they, what company do they belong to
  - Development type: organizational Framework
  - Contract: Defined variables?
  - Trust: Do parties involved trust each other?
- Stakeholder availability and capability
- Shared understanding
- Complexity and criticality
- Constraints
- Time and budget available
- Volatility of Requirements
- Experience of Requirements Engineers

5.2 RE Process Facets

If possible: configure process from preexisting elements

5.2.1 Time Facet: linear vs iterative

Criteria for Linear RE linear RE process, requirements are specified up front in a single phase of the process.

- The development process for the system is plan-driven and mostly linear.
- The stakeholders are available, know their requirements, and can specify them up front.
- A comprehensive requirements specification is required as a contractual basis for outsourcing or tendering the design and implementation of the system.
- Regulatory authorities require a comprehensive, formally released requirements specification at an early stage of the development

Criteria for Iterative RE iterative RE process, requirements are specified incrementally, starting with general goals and some initial requirements and then adding or modifying requirements in every iteration.

- The development process for the system is iterative and agile.
- Many requirements are not known up front but will emerge and evolve during the development of the system.
- Stakeholders are available such that short feedback loops can be established as a means of mitigating the risk of developing the wrong system.
- The duration of the development allows for more than just one or two iterations.
- The ability to change requirements easily is important.

5.2.2 Purpose Facet Prescriptive vs. explorative

Criteria for prescriptive RE prescriptive RE process, the requirements specification constitutes a contract: all requirements are binding and must be implemented

- The customer requires a fixed contract for system development, often with fixed functionality, scope, price, and deadline
- Functionality and scope take precedence over cost and deadlines.
- The development of the specified system may be tendered or outsourced

Criteria for explorative RE explorative RE process, only the goals are known a priori, while the concrete requirements have to be elicited.

- Stakeholders initially have only a vague idea about their requirements.
- Stakeholders are strongly involved and provide continuous feedback.
- Deadlines and cost take precedence over functionality and scope.
- The customer is satisfied with a framework contract about goals, resources, and the price to be paid for a given period of time or number of iterations.
- It is not clear a priori which requirements shall actually be implemented and in which order they will be implemented.

5.2.3 Target Facet: Customer Specific vs. Market Oriented

Criteria for Customer Specific RE customer-specific RE process, the system is ordered by a customer and developed by a supplier for this customer

- The system will be used mainly by the organization that has ordered the system and pays for its development.
- The important stakeholders are mainly associated with the customer's organization.
- Individual persons can be identified for the stakeholder roles.
- The customer wants a requirements specification that can serve as a contract.

Criteria for Market-Oriented RE a market-oriented RE process, the system is developed as a product or service for a market, targeting specific user segments

- The developing organization or one of its clients intends to sell the system as a product or service in some market segment.
- Prospective users are not individually identifiable.
- The Requirements Engineers have to design the requirements so that they match the envisaged needs of the targeted users.
- Product owners, marketing people, digital designers, and system architects are primary stakeholders

6 Management Practices for Requirements

Requirements management can occur at different levels: (Use Version Control for Requirements)

- The individual requirements
- The work products that contain these requirements
- The system related to the work products and the requirements contained therein

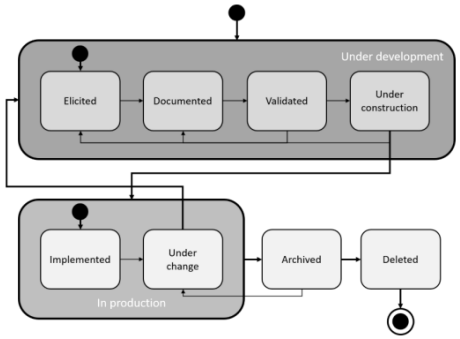


Abbildung 5: Requirements Life Cycle

6.1 Configuration

Has a product and a version dimension (which requirements in which version)

Properties of a correct Configuration

- Logically connected. The set of requirements in the configuration belongs together in view of a certain goal.
- Consistent. The set of requirements has no internal conflicts and can be integrated in a system.
- Unique. Both the configuration itself and its constituent requirements are clearly and uniquely identified.
- Unchangeable. The configuration is composed of selected requirements, each with a specific version that will never be changed in this configuration.
- Basis for reset. The configuration allows fallback to a previous configuration if any undesired changes appear to have occurred.

6.2 Attributes and Views

Requirement Attributes Use attributes to priorities / estimate effort / plan requirements

- Identification (e.g. number)
- Stakeholder priority. The (agreed) priority of the requirement from the viewpoint of the stakeholders.
- Dependency between Requirements
- Risk(s)
- Source (origin of the requirement)
- Rationale (why the requirement is needed)
- Type (functional or quality requirement)

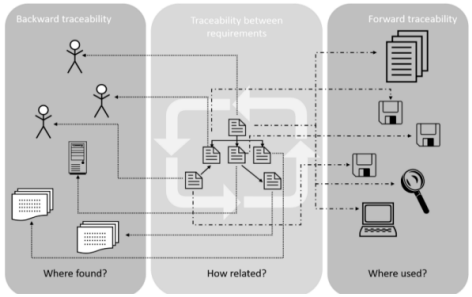


Abbildung 6: Traceability Types

Requirement Views

- Selective Views (deliberate selection of Requirements)
- Projective views (only a seleciton of all attributes)
- Aggregating views (summaries, totals, averages from a selection of Requirements)

6.3 Traceability

Necessary to:

- Provide evidence that a certain requirement is satisfied
- Prove that a requirement has been implemented and by what means
- Show product compliance with applicable laws and standards
- Look for missing work products (e.g., find out whether test cases exist for all requirements)
- Analyze the effects of a change to requirements

6.4 Prioritization

- Define major goals and constraints for the prioritization
- Define desired assessment criteria
- Define the stakeholders that have to be involved
- Define the requirements that have to be prioritized
- Select the prioritization technique (Ad Hoc vs Analytical)
- Perform Prioritization

7 Tool Support

7.1 All used Tools combined should support:

- Management of requirements
- Requirements Engineering process
- Documentation of knowledge about the requirements
- Modeling of requirements
- Collaboration
- Testing/ Simulation of Requirements

7.2 Aspects to take into account for new tools

- Consider All Life Cycle Costs beyond License Costs
- Consider Necessary Resources
- Avoid Risks by Running Pilot Projects
- Evaluate the Tool according to Defined Criteria
  - Project perspective
  - Process perspective
  - User perspective
  - Product perspective
  - Supplier perspective
  - Economic perspective
  - Architecture perspective
- Instruct Employees on the Use of the Tool

8 Questions

- Qualitätsanforderungen werden mit funktionalen Anforderungen ermittelt & können funktionale Anforderungen spezifizieren.
- Formalisierung von Anforderungen ist keine Aufgabe des RE.
- Grundlegende Prinzipien des RE:
  - Wertorientierung
  - Problem-Anforderung-Lösung
  - Systematische und disziplinierte Arbeit

9 Arbeitsprodukte

- Nur dann nicht redundant, wenn jede Anforderung nur einmal dokumentiert und keine Überschneidungen auftreten.
- Konsistent, wenn keine einzelne Anforderung im Widerspruch zu anderen Anforderungen steht.
- Auch vollständig, ohne alle relevanten Anforderungen zu enthalten.
- Vorlagenbasierte Arbeitsprodukte bieten eine Blaupause für die Strukturierung einzelner Anforderungen sowie ganzer Spezifikationen.

10 Basisanforderungen

Beste Erhebungstechnik: Feldbeobachtung

11 Diagrammtypen und Beschreibung

- **Use Case:**
  - Akteure einer Applikation
  - Grenze zwischen Applikation und Umgebung
  - Funktionalität einer Applikation
- BPMN
- Laus-Ohl
- Aktivitätsdiagramm
- Klassendiagramm

12 Lösung von Anforderungskonflikten

- Ober sticht Unter
- Variantenbildung
- Kompromiss

13 Validierungsverfahren

- A/B Tests
- Prototyp
- Walkthrough
- Inspektion

14 Praktiken für Requirements Management

- Es müssen nicht alle Anforderungen gelesen werden, um eine bestimmte Perspektive zu verstehen.
- Anforderungen können gruppiert werden.
- Anforderungen können für nicht autorisierte Stakeholder unsichtbar gemacht werden.