Week 5 – Documenting Requirements

Requirement Engineering

Book Pohl, Klaus. *Requirements engineering: fundamentals, principles, and techniques*.
Section 04

Fast-National University of computer & Emerging Sciences

Document Design

Requirements Document / Requirements Specification

A requirements specification is a systematically represented collection of requirements, typically for a system or component, that satisfies given criteria.

Document Design

Reasons for the documentation
During the life cycle of a requirements document, many people are trusted with the documentation. During communication, the documentation has a goal-oriented and supporting role. Main reasons are:

- Central role of requirements
- Legal relevance
- Complexity
- Accessibility

Document Design

Central role of requirements

Requirements are the basis of the system development. Requirements of any kind influence the analysis, design, implementation, and test phases directly and indirectly.

The quality of a requirement or of a requirements document has a strong impact on the progress of the project and therefore on its success.

Legal relevance

Requirements have a legal relevance. Requirements are legally binding for the contractor and the client, and the client can sue for their fulfillment.

Documenting the requirements can help to quickly overcome legal conflicts between two or more parties.

Document Design

Complexity

- *Requirements documents are complex.* Systems that possess thousands of requirements that in turn have complex interdependencies on multiple layers are not unheard of in practice.

- Without suitable documentation, keeping on top of things can become very difficult for anyone involved.

Accessibility

- *Requirements must be accessible to all involved parties.* Projects undergo certain "development" as time goes by—with regard to the subject as well as the staff.

- When requirements can be permanently accessed, uncertainty and obscurities can be avoided and staff that has recently joined the project can quickly get up to speed.

Document Design

Another argument for a good documentation, supportive of the project, is that employees almost never share the same understanding of a subject matter. Therefore, requirements should be documented in a way that they meet the quality demands of all involved.

Types of Documentation

The Three Perspectives of Requirements

1. ***Data perspective***

Data perspective: In the data perspective, a **static-structural perspective** on the requirements of the system is adopted.

For example, the structure of input and output data as well as static-structural aspects of usage and dependency relations of the system and the system context can be documented (e.g., the services of an external system).

Types of Documentation

2. Functional perspective

The functional perspective documents which information (data) is received from the system context and manipulated by the system or one of its functions.

This perspective also documents which data flows back into the system context. The order in which functions processing the input data are executed is also documented.

Types of Documentation

3. *Behavioral* perspective

In the behavioral perspective, information *Behavioral perspective* about the system and how it is embedded into the system context is documented in a state-oriented manner.

This is done by documenting the reactions of the system upon events in the system context, the conditions that warrant a state transition, and the effects that the system shall have on its environment (e.g., effects of the system analyzed that represent events in the system context of a different system).

Types of Documentation

Requirements Documentation using Natural Language

Advantages of using natural language

- Natural language, particularly prose, is the most commonly applied documentation form for requirements in practice.

- In contrast to other documentation forms, prose has a striking advantage: No stakeholder has to learn a new notation. In addition, language can be used for miscellaneous purposes—the requirements engineer can use natural language to express any kind of requirement.

Types of Documentation

Requirements Documentation using Natural Language

Disadvantages of using natural language

- Natural-language-based documentation is well suited to document requirements in any of the three perspectives.

- However, natural language can allow requirements to be ambiguous, and requirements of different types and perspectives are in danger of being unintentionally mixed up during documentation.

- In that case, it is difficult to isolate information pertaining to a certain perspective amidst all of the requirements in natural language.

Types of Documentation

Requirements Documentation using Conceptual Models

- When documenting requirements by means of models, special modeling languages must be used that pertain to the appropriate perspective. Assuming the modeling language selected for a documentation task is applied correctly, its use constructively guarantees that the models created depict information pertaining to the respective perspective only.

- The models depict the documented requirements much more compactly and they therefore are easier for a trained reader to understand than is natural language. In addition, conceptual models offer a decreased degree of ambiguity (i.e., fewer ways to be interpreted) than natural language due to their higher degree of formality.

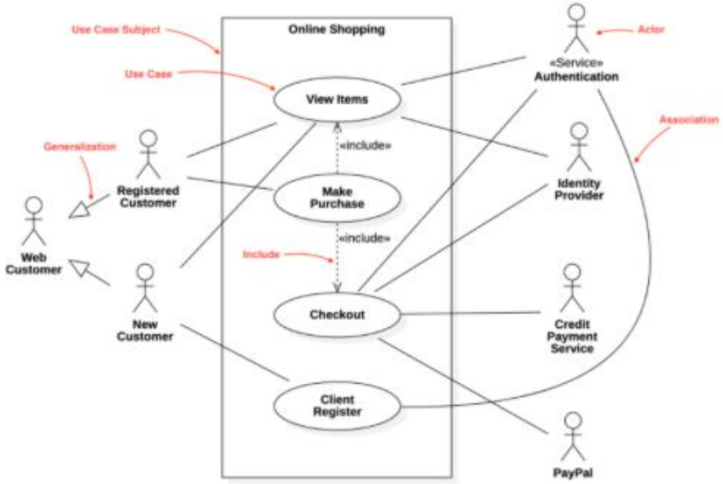
- However, using conceptual modeling languages for requirements documentation requires specific knowledge of modeling.

Types of Documentation

List includes short descriptions of the most important diagrams:

- ***Overview of system functions***

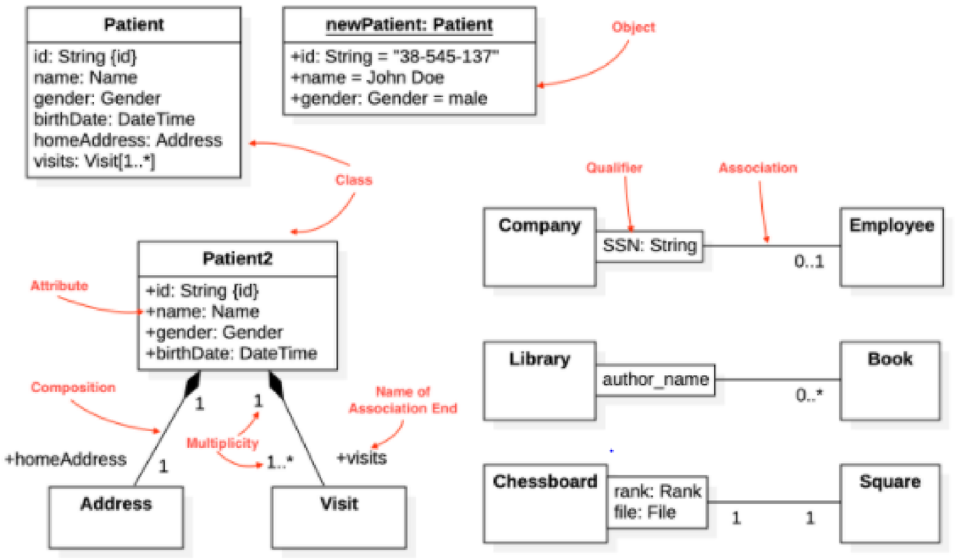
Use case diagram: A use case diagram allows you to gain a quick overview of the functionalities of the specified system. A use case describes which functions are offered to the user by the system and how these functions relate to other external interacting entities.



Types of Documentation

- ***Structural data modeling and structuring of terms***

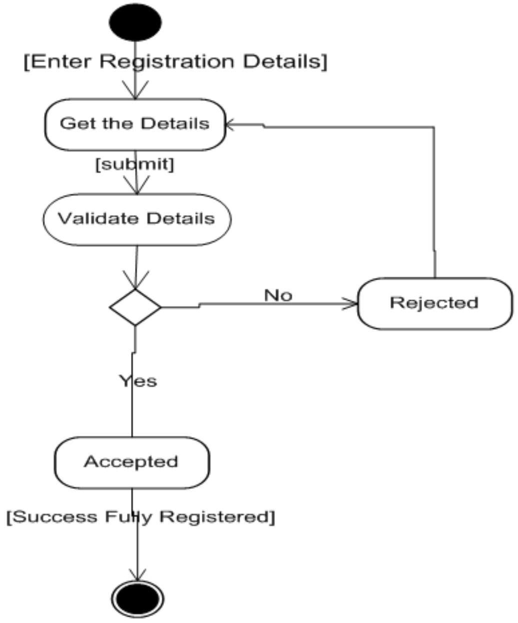
Class diagram: are used in requirements engineering to document requirements with regard to the static structure of data, to document static-structural dependencies between the system and the system context, or to document complex domain terms in a structured manner



Types of Documentation

- ***Sequence modeling***

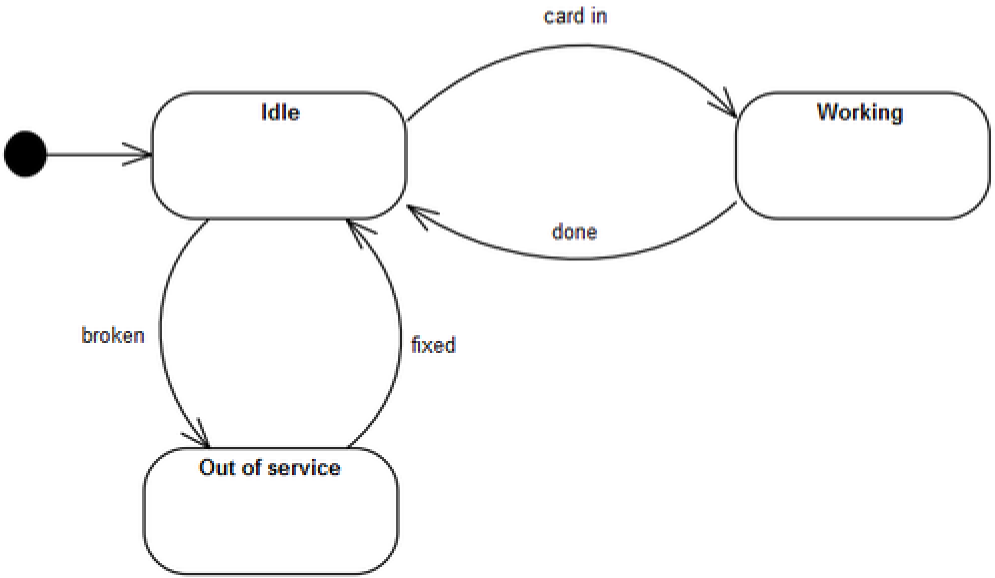
Activity diagram: Using activity diagrams, business processes, or sequence-oriented dependencies of the system in regard to processes within the system context can be documented. Activity diagrams are also well suited to model the sequential character of use cases or to model a detailed specification of the interaction of functions that process data



Types of Documentation

- ***Event-driven behavior***

State diagram are used in requirements engineering to document event-driven behavior of a system. The focus of this type of model is on the individual states the system can be in, events and their respective conditions that trigger a state transition, and effects of the system in its environment.



Types of Documentation

- **Hybrid Requirements Documents are** *Combined use of documentation types*
- Requirements documents first and foremost contain requirements. In addition, in many situations it is sensible to document decisions, important explanations, and other relevant information as well.

- Depending on the target audience of the document, the perspective on the system, and the documented knowledge, suitable documentation types are selected.

- Typically, documents contain a combination of natural language and conceptual models. The combination allows the disadvantages of both documentation types to be decreased by means of the strengths of the other documentation type, and combining documentation types exploits the advantages of both.

- For instance, models can be amended or complemented by natural language comments and natural language requirements and natural language glossaries can be summarized and their dependencies can be depicted clearly by making use of models.

Document Structure

Influence of the requirements on satisfaction

- Requirements documents contain a magnitude of different information.

- These must be well structured for the reader. In order to do that, one can make use of standardized document structures or individually define a custom document structure.

Document Structure

Standardized Document Structure is *Adaptation of existing standard outlines* offer a predefined structure, i.e., predefined stereotypes according to which the information can be classified.

Advantages of using standard Standard outlines

- simplify incorporating new staff members.
- Standard outlines allow for quickly finding desired contents.
- Standard outlines allow for selective reading and validation of requirements documents.
- Standard outlines allow for automatic verification of requirements documents (e.g., with regard to completeness).
- Standard outlines allow for simplified reuse of the contents of requirements documents.

Document Structure

Three of the most widely used standardized document structures are

1. ***Rational Unified Process***

- The *Rational Unified Process (RUP)* [Kruchten 2001]is usually used for *Rational Unified Process* software systems that are developed using object-oriented methods.

- The client creates a *business model* that contains different artifacts from the business environment (e.g., business rules, business use cases, business goals), which serve as the basis for requirements of the system over the course of development.

- The contractor uses the structures of the *software requirements specification (SRS)* to document all software requirements.

- These structures are closely related to the IEEE standard 830.

Document Structure

2. IEEE standard 830-1998

The *IEEE standard 830-1998* [IEEE Std. 830-1998.] (Recommended Practice for Software Requirements Specifications) contains an outline that was specially designed for the documentation of software requirements. Three main chapters are :

- Main chapter with introductory information (e.g., system goal, system bounding)

- Main chapter for general descriptions of the software (e.g., perspective of the system, properties of the future users, constraints for development)

- Main chapter for specific requirements (e.g., functional requirements, performance, interfaces)

Document Structure

3. V Model

The *V-Model* [V-Modell 2004] of the German Federal Ministry of the Interior (BMI) defines different structures, depending on the creator of the requirements document:

- The *Customer Requirements Specification :* is created by the customer and describes all of the demands to the contractor regarding the subject of the contract. the Customer Requirements Specification usually describes *what* is made *for what*.

- The *System Requirements Specification*, known in the German original as Pflichtenheft, is based on the Customer Requirements Specification and contains the implementation suggestions that the contractor has elaborated. It is a refinement of the requirements and constraints of the Customer Requirements Specification.

Document Structure

Customized Standard Contents

The minimum content that standardized document structures are adapted with regard to the specific project conditions.

Introduction

The introduction contains information about the entire document. This information allows gaining a quick overview of the system.

- Purpose:
- System coverage
- Stakeholder
- Definitions, Acronyms, and Abbreviations
- References
- Overview

Document Structure

General Overview

Information is documented that increases the understandability of the requirements. In contrast to the introduction, this is merely operational information that does not pertain to administration, management, or organizational aspects of the requirements document.

- System environment
- Architecture description
- System functionality
- User and target audience
- Constraints:
- Assumptions

Document Structure

Requirements

This part contains functional requirements as well as quality requirements.

Appendices

additional information that completes the document can be documented e.g. user characteristics, standards, conventions, or background information regarding the requirements document.

Index

The index typically contains a table of contents (i.e., a structure of the chapters) and an index directory. Keep it updated.

Using Requirements Documents

Requirements documents as the basis for development

- **Planning:** Based on the requirements document, concrete work packages and milestones for the implementation of the system can be defined.

- **Architectural design:** The detailed documented requirements (along with constraints) serve as the basis for the design of the system architecture.

- **Implementation:** Based on the architectural design, the system is implemented by making use of the requirements.

- **Test:** On the basis of requirements that have been documented in the requirements document, test cases can be developed that can be used for system validation later on.

Using Requirements Documents

Change management: When requirements change, the requirements document can serve as the basis to analyze the extent to which other parts of the system are influenced. The change effort can thus be estimated.

System usage and system maintenance: After the system is developed, the requirements document is used for maintenance and support. This way, the requirements document can be used to analyze concrete defects and shortcomings that surface during system use. For example, one can deduct if a defect is a result of using the system incorrectly, a result of an error in requirements, or a result of an error in implementation.

Contract management: The requirements document is the prime subject of a contract between a client and a contractor in many cases.

Quality Criteria for Requirements Documents

Along with the quality criteria that are suggested in the IEEE standard [IEEE Std. 830-1998], the requirements document should possess a clear structure and be reasonably comprehensive. Thus, for requirements documents, the following criteria must hold:

- Unambiguity and consistency [IEEE Std. 830-1998]
- Clear structure
- Modifiability and extendibility [IEEE Std. 830-1998]
- Completeness [IEEE Std. 830-1998]
- Traceability [IEEE Std. 830-1998]

Quality Criteria for Requirements Documents

Unambiguity and consistency

Quality of individual requirements is a prerequisite. The individual requirements are consistent and unambiguous.

Individual requirements do not contradict one another. Make use of conceptual models.

Another aspect of unambiguity pertains to the unique identification of a requirements document or a requirement among the set of all requirements documents or requirements in a development project

Quality Criteria for Requirements Documents

Clear Structure

- *Allows for selective reading.* Requirements document is readable by any stakeholder, it should be appropriately comprehensive and clearly structured.

- A very comprehensive requirements document with a good structure can be just as appropriate as a less comprehensive document because a clear structure will allow the reader to skip parts that are not relevant to him.

- An unstructured or badly structured requirements document of the same high level of comprehensiveness would not be appropriate because the document must be read in its entirety in order for a stakeholder to be able to identify parts that are relevant to her.

Quality Criteria for Requirements Documents

Modifiability and Extendibility

- *Content and structure should support changeability.* Requirements documents must be easy to extend. There are always requirements that are changed, altered, added, or removed as a project progresses.

- The requirements documents of a project should be subject to the project's version control management

Quality Criteria for Requirements Documents

Completeness

- *Two types of completeness in requirements documents* All possible inputs, influential factors, and required reactions of the system must be described for each desired system function. This comprises describing error and exception cases in particular. Also, quality requirements, such as requirements pertaining to reaction times or availability and usability of the system, must be noted.

- *Evidence, reference, and sources are formal necessities.* Formal factors also contribute to completeness. Graphs, diagrams, and tables should be appropriately labeled.

Quality Criteria for Requirements Documents

Traceability

- *Relationship to other development documents.* An important quality criterion is traceability of relationships between requirements documents and other documents (e.g., business process model, test plans, or design plans).

- These documents could have been created in previous development phases, in subsequent development phases, or concurrently with the requirements documents. Among other things, traceability supports change management

Quality Criteria for Requirements

The formal criteria according to [IEEE Std. 830-1998] as well as criteria that increase the readers' acceptance of the requirements are :

- *Agreed:* A requirement is agreed upon if it is correct in the opinion of all stakeholders and all stakeholders accept it as valid.

- *Ranked:* When the complexity or comprehensiveness of a system reaches a certain threshold, it is important to rate the requirements, for example, according to their importance, legal obligation, or priority.

- *Unambiguous:* A requirement that is unambiguously documented can be understood in only one way. It must not be possible to interpret the requirement in a different way. All readers of the requirement must arrive at the same understanding of the requirement.

Quality Criteria for Requirements

- *Valid and up-to-date:* A documented requirement must represent the facts and conditions of the system context in a way that it is valid with regard to the actualities of the system context. These actualities may be the different stakeholders' ideas, relevant standards, or interfaces to external systems.

- *Correct:* A requirement is correct if it adequately represents the idea of the stakeholder. This also means that the requirement may not express more than what the stakeholder was trying to say. In order to verify that, it is necessary for the stakeholders to be able to read and understand the requirements documentation. Therefore, correctness of the specification can only be verified if the criterion of understandability is met.

- *Consistent:* Requirements must be consistent with regard to all other requirements, i.e., the requirements must not contradict one another, regardless of their level of detail or documentation type. In addition, a requirement must be formulated in a way that allows for consistency with itself, i.e., the requirement may not contradict itself.

Quality Criteria for Requirements

- *Verifiable:* A requirement must be described in a way that allows for verification. That means that tests or measurements can be carried out that provide evidence of the functionality demanded by the requirement.

- *Realizable:* It must be possible to implement each requirement given the organizational, legal, technical, or financial constraints. This means that a member of the development team ought to be involved in rating the goals and requirements so that he can show the technical limits of the implementation of a particular requirement. In addition, the costs for the implementation must be incorporated into the rating. Occasionally, stakeholders withdraw a requirement if the costs for its realization become apparent.

- *Traceable* A requirement is traceable if its origin as well as realization and its relation to other documents can be retraced. This can be done by means of unique requirement identifiers. A system goal can be traced through all levels of abstraction, from design to implementation and test.

Quality Criteria for Requirements

- *Complete:* Each individual requirement must completely describe the functionality it specifies. Requirements that are yet incomplete must be specially marked, for example by inserting “tbd” (“to be determined”) into the respective text field or by setting a corresponding status.

- *Understandability:* Requirements must be comprehensible to each stakeholder. Therefore, the type of requirements documentation can vary significantly, depending on the development phase (and therefore, depending on the involved staff). In requirements engineering, it is important to strictly define the terms used.

Quality Criteria for Requirements

Fundamental principles of understandability

there are two fundamental rules that enhance the readability of requirements:

- *Short sentences and short paragraphs:* As human short-term memory is very limited, circumstances that belong together should be described in no more than seven sentences.

- *Formulate only one requirement per sentence:* Formulate requirements using active voice and use only one process verb. Long, complicated interlaced sentences must be avoided.

Glossary

A glossary is a collection of term definitions and contains the following elements:

- Context-specific technical terms
- Abbreviations and acronyms
- Everyday concepts that have a special meaning in the given context
- Synonyms, i.e., different terms with the same meaning
- Homonyms, i.e., identical terms with different meanings

Glossary

Consistent definitions

- By defining the meaning of terms, you can increase the understandability of requirements considerably. Misunderstandings and different interpretations of terms that might lead to conflicts can be avoided from the beginning.

Reuse of glossary entries

- In different projects, terms are used that are similar to one another or in fact identical. This may be the case, for example, when one system is developed for different customers but within the same domain.

- It may even be feasible to define such terms in a universal, inter-project glossary. The additional effort of creating such a glossary will pay off in future projects.

Rules for Using a Glossary

1. *The glossary must be centrally managed*
2. *Responsibility must be assigned*
3. *The glossary must be maintained over the course of the project*
4. *The glossary must be commonly accessible*
5. *Using the glossary must be obligatory*
6. *The glossary should contain the sources of the terms*
7. *The stakeholders should agree upon the glossary*
8. *The entries in the glossary should have a consistent structure*

To reduce the effort of aligning terms with one another, it is advisable to start with the creation of the glossary early on in the project.

Summary

The documentation of requirements plays a central role in requirements engineering. As the amount of requirements is often vast, it is very important to clearly structure the requirements so that personnel not involved with the project also understand them.

Looking up and changing requirements is simplified and accelerated in this way. This makes meeting the quality criteria for requirements documents much easier.

Using customized documentation structures has proven to be suitable for that purpose. These are completed by inserting project-specific requirements written in natural language in conjunction with conceptual requirements models.

References

- Pohl, Klaus. *Requirements engineering: fundamentals, principles, and techniques*.(Chapter 4).