Software Engineering

Prof. Dr. Christoph Schober

Department of Applied Computer Science,
Deggendorf Institute of Technology

christoph.schober@th-deg.de

Requirements Engineering

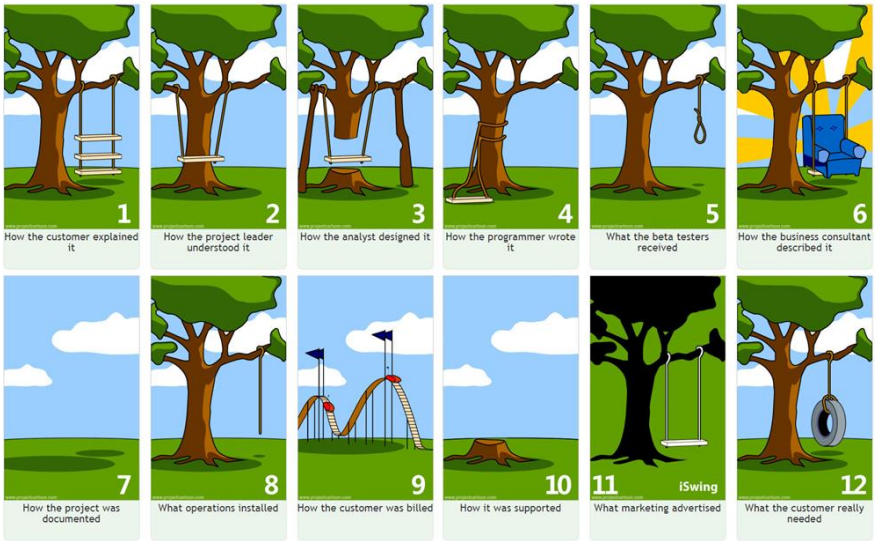
Requirements Engineering

1. Definitions

Definitions

https://www.youtube.com/watch?v=BKorP55Aqvg

Definitions



Stakeholder

Any person or group that is affected by, or has interest in, the development of a software system.

Examples:

- Customers
- Business owners
- Project managers
- Regulatory bodies
- Software developers
- Software testers
- System operators
- Competitors

→ Effective stakeholder management is critical to the success of a project.

Requirement

A description of a functionality, capability, or constraint that a software system must possess to meet the needs of its stakeholders.

- May range from a high-level abstract statement of a service to a detailed mathematical function specification.
- Requirements may serve a dual function:
  - May be the basis for a bid for a contract - must be open for interpretation.
  - May be the basis for the contract itself - therefore must be defined in detail.

Types of requirements:

- *Functional:* Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- *Non-Functional*: Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- *Domain:* Constraints on the system from the domain of operation.

Examples: See the document ”Task Assignment” in iLearn.

Requirements Engineering

Process of eliciting, analyzing, documenting, validating, and managing the requirements for a system or software development project.

- In other words: Understand what needs to be developed.
- Involves identifying stakeholders needs, defining system capabilities, and ensuring that the requirements are complete, consistent, and unambiguous.

Agile methods and requirements engineering?

- Many agile methods argue that producing detailed system requirements is a waste of time as requirements change so quickly.
- A requirement document is therefore always out of date.

→ Agile methods usually use incremental requirements engineering and may express requirements as *user stories*.

Requirements Engineering

2. Requirements Elicitation

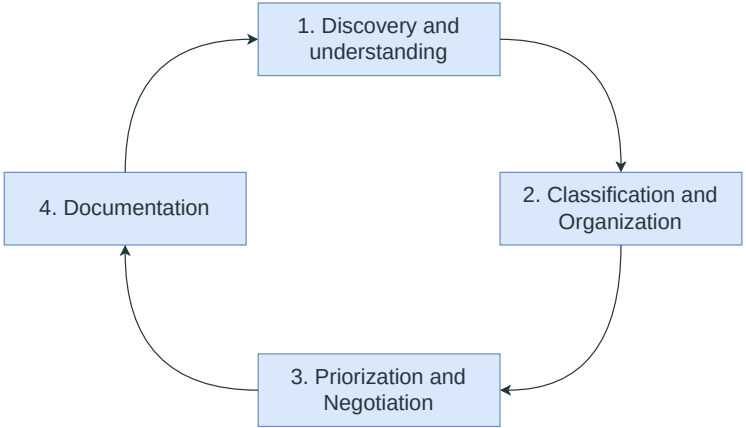
Requirements Elicitation

The process of discovering and gathering information about the needs, expectations, and preferences of stakeholders.

- The goal is to ensure that the system is designed and developed according to the expectations of the stakeholders.
- Collaborative process that involves the participation of various stakeholders, including users, customers, project managers, and developers.
- Helps to minimize the risk of project delays, cost overruns, and unsatisfied stakeholders.

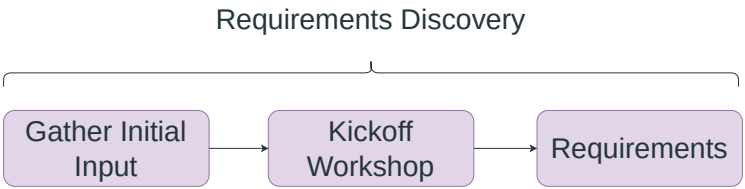
Challenges:

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organizational and political factors may influence the system requirements.
- The requirements **change during the analysis process**. New stakeholders may emerge and the business environment may change.


1. Discovery and understanding
2. Classification and Organization
3. Priorization and Negotiation
4. Documentation

Process activities:

- *Discovery*: Interacting with stakeholders to discover their requirements.
- *Classification & Organization:* Group related requirements and organize them into coherent clusters.
- *Prioritization & Negotiation:* Prioritize and resolve requirements conflicts.
- *Documentation:* Requirements are documented and input into the next round of the spiral.


Requirements Discovery
Gather Initial Input
Kickoff Workshop
Requirements

Interview as technique for discovery: Gather & fixate first input.

1. *From your perspective, what are important jobs that need to be done?*
   → I need to hang a painting on the wall.
2. *Which pain points are currently existing in this context?*
   → I do not have a drilling machine available.
3. *Which gains can be expected?*
   → When I did this, my flat is nice looking and cozy.
4. *Which alternatives can be imagined to also solve this topic?*
   → Until now, I used Tesa Power Strips.
5. *Which people should be in the kickoff workshop?*
   → My wife and the kids.

Workshops as technique for discovery: Collaborative and interactive sessions.

1. *Icebreakers*: Help the participants to feel more comfortable working together.
2. *Project overview*: Discuss project goals, scope, timeline, and budget.
3. *Stakeholder analysis*: Identify stakeholders, their roles, and their needs.
4. *Scenarios*: Create user personas and scenarios to understand the end-user.
5. *Requirements gathering*: Brainstorm to identify the project requirements.
6. *Refinement*: Ensure that the requirements are clear, concise & complete.

Requirements gathering technique: Mind Mapping

- Visual brainstorming technique that involves creating a diagram of related ideas and concepts, using lines and keywords to connect them.
- Write the central idea or topic in the center of the page, and then draw lines branching out from it to represent different subtopics or related ideas.
- Each subtopic is represented by a keyword or short phrase.

Requirements gathering technique: Six Thinking Hats

- Involves six different colored ”hats” that represent different ways of thinking.
- By wearing and switching hats, individuals can explore a problem or situation from multiple perspectives → make better decisions.

Hat	Thinking	Description
White	Objective	Focus on facts and information.
Red	Emotional	Express feelings, make gut-level judgments.
Black	Critical	Identify problems and risks, be skeptic.
Yellow	Positive	See benefits and opportunities, can-do-mentality.
Green	Innovative	Generate ideas, explore different approaches.
Blue	Strategic	Step back and analyze the thinking process itself.

Requirements Engineering

3. Requirements Specification

Requirements Specification

The process of writing down the requirements in a requirements document.

Ways of writing a requirements' specification:

- Natural language sentences (e.g., prosa text).
- Structured natural language (e.g., user stories).
- Graphical notations (e.g., UML diagrams).
- Mathematical specifications (e.g., finite-state machines).

→ Agile methods prefer structured natural language in the form of **user stories**.

User Story

A brief, simple description of a feature or functionality of a software system, written from the perspective of the end-user or customer.

Structured format:

```
1 As a [user], I want to [action], so that [goal].
```

→ User stories help to ensure that software development remains focused on delivering value to users by keeping the user's perspective at the forefront.

Examples:

- *As a customer, I want to be able to add items to my shopping cart, so that I can easily purchase multiple products at once.*
- *As a social media user, I want to be able to block other users, so that I can avoid interactions with people I do not want to communicate with.*
- *As a mobile app user, I want to be able to set reminders for tasks, so that I can stay on top of my to-do list.*
- *As a manager, I want to be able to view sales reports, so that I can track performance and make data-driven decisions for my business.*

Requirements Engineering

4. Summary

Requirements engineering in context of Scrum:

- Integrated into the development process through the Product Backlog.
- User stories are typically part of the description of Product Backlog items.
- Continuous refinement of requirements as the development progresses.
- In addition to the Product Backlog, the Scrum framework includes other practices that support requirements engineering, such as Sprint Reviews, Sprint Retrospectives, and Daily Scrums.

→ Welcome change and identify issues early to take corrective actions.

Summary

You should have acquired the following competencies:

- Understand that requirements for a software system set out what the system should do and define constraints on its operation and implementation.
- Be able to classify requirements in functional, and non-functional ones.
- Know the requirement engineering activities.
- Write user stories.
- Apply requirements engineering practices in context of agile methods.