Use Cases

Software Requirements

- A **software requirement** is:
  1. A **condition or capability** needed by a **user** to **solve a problem** or **achieve an objective**.
  2. A **condition or capability** that must be **satisfied by a system**.
  3. A **documented representation** of a **condition or capability** as described in (1) and (2).
- A **functional requirement** is a requirement that **describes what a system must do** including **processes, interfaces, and data**.
  - **Function requirements** are described in **use case documents**, and are modelled in **OOA** with **use case diagrams, class diagrams, and interaction diagrams**.
- A **non-functional requirement** is a requirement that **specifies how the system must perform** including **response time, security considerations, and the volume of data**.
  - **Non-functional requirements** are documented in a **requirement list**.
- A **usability requirement** is a requirement that is concerned with **matching the system to the way people work**.
  - **Usability requirements** measure objectives, including **characteristics of users, tasks users undertake, situational factors, and the acceptance of criteria for the working system**.
  - **Usability requirements** are documented in the **list of requirements** and may be tested by **prototypes**.

Techniques for Finding Requirements

- There are **several techniques for finding requirements**.
- **Background reading** is a technique for finding **requirements** that aims at **understanding the organization** and it's **business objectives**.
  - Reading material includes **reports, charts, policies, job descriptions, and existing system documentation**.
  - This technique **works best** in the **initial stage of fact finding** and when the **analyst is not familia with the organization**.
- **Interviewing** is a technique for getting an **in-depth understanding** of the **organization's objectives, and user roles**.
  - Interview subjects include **managers, staff, and customers**.
  - This technique **works best** when **in-depth** information is required. The effectiveness of this technique depends on the skill of the interviewer.
- **Observation** is a technique for find out what **really happens**, **not what people say happens**.
  - Items to observe include **what happens to documents, how people carry out processes, quantitative data, a processes from end-to-end**.
- **Document sampling** is a technique for **providing statistical data** about **transaction volumes** and **activity patterns**.

  - Document sampling information includes **copies of empty and completed documents, screenshots of existing systems, numbers of forms filled in, and the lines on the forms**.
  - This technique **works best** when **error rates are high, large volumes of data are being processes**.

- **Questionnaires** are a technique for obtaining the **views of a large amount of people in a way that can be analyzed statistically**.
  - Questionnaires include **postal, web-based, and email questionnaires with open-ended and closed-ended questions**. They also **gather opinions and facts**.
  - This technique **works best** when **staff organizations are geographically dispersed, the system is going to be used by the general public, and when you need to obtain the views of a large amount of people**.

Use Case Descriptions

Use Case Modeling

- A **use case** is primarily an **action of writing text**.
- An **actor** is a **person or thing** that **interacts** with the software.
- A **use case** describes **what happens in the system** when an **actor uses the software**.
- **Use case modeling** may include a **use case diagram**; showing the **name, actors, and relationships** of **use cases**.
- **Use case development is a key characteristic** of the **Unified Process**. It serves to help discover **functional requirements, design construction, test plans, and maintenance to prepare user manuals**.

Types of Use Case Descriptions

- There are **3 types** of **use case descriptions**:
  1. A **brief use case description** consists of a **single paragraph** describing the **main success scenario**.
  2. A **casual use case description** consists of **multiple informal paragraphs** covering both the **main success scenario**, and **various alternatives**.
  3. A **fully dressed use case description** consists of a **detailed description of all steps involved in the main success, alternative, and exception scenarios**. This is usually accompanied by supporting sections, such as pre-conditions and post-conditions.
- The **fully dressed use case description** contains the following sections:
  1. **The primary actor** - The user who interacts with the system during this use case.
  2. **Stakeholders and their interests** - The use case covers the functionality that satisfies all the required stakeholder's interests.
  3. **Pre-conditions** - Conditions that must be true before the main scenario begins without any checking.
  4. **Post-conditions** - Conditions that must be true on the successful completion of a use case.
  5. **The main success scenario (detailed)** - The typical path to a successful outcome (describes what needs to happen not how).

6. **Alternative flows (detailed)**. - All other paths that may lead to a success or failure.
7. **Exceptions (detailed)**. Exceptions that may occur.
8. **Special requirements** - Non-functional requirements for the use case.
9. **Open issues** - Anything that hs an effect on the functionality of the use case.