Requirements Engineering

Tags

Learning Objective:

- What is System requirements
- Stakeholder identification
- Functional and Non-Functional Requirements
- Describe system behaviour via user model and user cases
- Requirements Quality

Requirements:

- **System Requirements:** specify a systems requirements not in terms of its implementation but in terms of user observations
- **Functional requirements:**
  - specify how the systems should function according to user specifications

    - How it should act with specific inputs
    - and how it should also **not** act
- Non-Specific requirements:
    - Constraints on service, i.e what is required for the functional requirements to be realised.

Analysing requirements:

1. **Identify stakeholders** :
    - Stakeholders are individuals or groups who have a vested interest in the completion of the project
    - They can influence the project's completion and can be categorised using the onion model:

- The first layer (most inner circle) is the product of service
- The second layer is the the system, this layer contains people that will interact with the product directly, e.g. users, programmers, testers, project managers.

- The third layer: Supportive. Are the people who do not interact directly with the product but benefit from it i.e. the **functional beneficiaries.** For example, marketing, sales, customer support staff, legal staff.
- The 4th layer represents the wider environment. This layer is populated by stakeholders who are outside the firm. for example, the public, media for a game this will be the reviewers, parents who are not going to interact with the game directly but will have a say if their children buys the game.
- You also have the negative stakeholders such as regulators who will put constraints on the project. For a game this will be the people who give the game age rating.

> 💡 In which group are the users in the onion model?

▼ Ans:

> 💡 What are the functional Beneficiaries

▼ Ans:

> 💡 In a online shopping store website who would the negative stakeholder be?

2. **Identify the top level user needs using User Stories :**

- User Stories are a method of describing how a user or other systems may interact with the system, a popular way to record this would be: As a <type of user>, I want to <some goal> so that <some reason> E.g.: A racing game

    - As a gamer I want to race other players / bots around various game maps so that I can achieve a faster time and compare myself to the leader board.
- A use case model, has **Actors** that represent the various methods a user can interact with the system for a specific goal. A use case describes the sequence of events, preformed by the system that yields an observable result

- Let’s take a look at an example: Online shopping system:

Actors:

- Customers: A person who is going to use the site to buy products
- Guests: A non-registered person who is browsing
- Seller: A registered user or company that sells the products on the platform
- System Administrator: A person who manages the systems

Use Cases:

- Browse Products
- Register for an account
- Login
- Add to cart
- Checkout
- Manage Listings
- Process Orders
- Leave Reviews
- Generate Reports
- Add products to sell

3. **Non-functional Requirements:**

- Non-functional requirements are those requirements of the product that do not have to with the exact functionality of the system but are equally important.
- In the online shopping website example above, the non-functionla requirements would be :
    - Performance requirements: The website should be able to load a sufficient amount of products at a time, it shouldn't be just one product per page. The webpage should load in a reasonable amount of time.
    - Scalability: The website should be able to support a growing list of products
    - Security: credit information about customers should be

Flow of events:

- A use case scenario is how an actor can interact with the use case, and each step is broken down to its simplest components.
- A flow of events is created where potential issues are identified
- As well as alternate flow for these alternate use cases

- For examples: In the example above a flow of event for a registered user would be :

1. Go onto the website
2. The user logs in via the log in button - > Here the login button must be readily apparent and at the same place every time the user logs in
3. Alternative flow: The user wants to log in but cannot remember their password, — > Have a button that allows the user to reset their password

4. The user clicks on the product
5. The specific product page opens

and so on....