



CS5002NP

Software Engineering



Agendas

- Requirements
- Types of Requirements
- Requirement Engineering
- Steps involved in Requirement Engineering



Requirement

- According to IEEE standards 729, a requirement is defined as:
 - A condition or capability needed by a user to solve a problem or achieve an objective
 - A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
 - A documented representation of a condition or capability as in 1 and 2



Requirement

- Description of features and functionalities of the target system
- Convey the expectations of users from the software product
- Can be obvious or hidden, known or unknown, expected or unexpected from client's point of view



Requirement - Example

- Let's say you're undertaking a project to build an e-commerce application for your client Steve



Deals and Promotions

Beats by Dre Studio 3

\$349.95 **\$279.99**

Shop More →

Explore Popular Categories



Computer & Laptop



Digital Cameras



Smart Phones



Televisions



Audio



Smart Watches



Requirement - Example

- You need to remember - "*The project belongs to the client and you are only responsible for delivering it*"
- Before carrying out the project, you must be clear about the
 - Functionalities that Steve wants in the application
 - The performance criteria and quality expectations of the application



Types of Requirements

- Functional Requirements
- Non-functional Requirements



Functional Requirements

- Requirements that the **end user specifically demands** as **basic facilities** that the system should offer
- Need to be **necessarily incorporated** into the system as a part of the contract
- Represented or stated in the form of **input** to be given to the system, the **operation performed** and the **output expected**
- Requirements stated by the user which **one can see directly in the final product**



Functional Requirements - Example

- Steve might say
 - The customer should be able to search for the products according to the category
 - The system should facilitate membership to the customers
 - The system should provide a platform to pay online or on delivery



Non-Functional Requirements

- Quality constraints that the system must satisfy according to the project contract
- The priority or extent to which these factors are implemented varies from one project to other
- Non-behavioral requirements.



Non-Functional Requirements

- They basically deal with issues like:

- Compatibility
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility



Non-Functional Requirements - Example

- Steve might say
 - The process time for each request should not be more than 6 seconds
 - User request to access data must be verified first
 - User layman is needed for the layman users



Requirement Understanding

- Understanding the requirements of problem is one of the most challenging tasks that Software Engineers have to face
- As there is a saying on *"Effective Requirement Practices"* by Ralph Young:
This happens most of the time when customers come to you and say: ***"I know you think you understand what I said, but what you don't understand is what I said is not what I meant"***
This means that what the customers means and what you understand might not always be the same



What's the Solution?

- A proper emphasis has to be given to while understanding requirements
- **REQUIREMENT ENGINEERING**



Requirement Engineering

- The process to **gather** the software requirements from client, **analyze** and **document** them
- Goal - **Develop** and **maintain** sophisticated and descriptive '***System Requirements Specification***' document





Inception

- Understanding the problem
- Main focus - carrying out ***feasibility study***




Feasibility Study

- **Decides** whether or not the proposed system is **worthwhile**
- A short focused study that checks:
 - If the system contributes to the organizational goals
 - If the system can be engineered using current technology and within budget
 - If the system can be integrated with other systems that are used



Elicitation

- Involves working with the **customer/end users** to **gather requirements, performance** and **hardware constraints**
- Process to **find out the requirements** for an intended software system by communicating with client, end users, system users and others who have stake in software development



Elicitation - Process

- **Requirement Discovery**

- Interacting with stakeholders to discover their requirements

- **Requirement Classification and Organization**

- Organizing the unstructured requirements and grouping them in terms of related requirements



Elicitation - Process

- **Requirement Prioritization and Negotiation**
 - Meeting the stakeholders to prioritize requirement and resolve conflicts if there are any
- **Requirement Specification**
 - Documenting the requirements



Elicitation - Techniques

- **Interview**

- Requirements are derived from formal and informal interactions with the system stakeholders



Elicitation - Techniques

- **Interview**

- 2 Types:

- Open Interview

- No predefined agenda
 - Things become clearer after exploring the issues with the stakeholders

- Closed Interview

- Predefined questions



Elicitation - Techniques

- **Interview**

- Advantage

- Clearer idea of what stakeholders do and interact with the new system

- Disadvantage

- Might not be helpful in understanding the requirements from the application domain



Elicitation - Techniques

- **Questionnaires**

- A document with predefined set of objective questions and respective options is handed over to all stakeholder to answer, which are collected and compiled



Elicitation - Techniques

- **Questionnaires**

- Advantage

- Very effective and easy to compile the feedback and results

- Disadvantage

- If the questions or options fail to represent the issue, the issue might be left out



Elicitation - Techniques

- **Observation**

- **Team of experts** visit the client's organization
- Observe **actual working environment**
- Observe **workflow at client's end** and how **execution problems are dealt**
- Team draws **conclusions** which **aids to form requirements** expected from the software



Elicitation - Techniques

- **Observation**

- Advantage

- First hand information gained by the requirement team or engineers

- Disadvantage

- People might behave differently when they know they are being observed



Elicitation - Techniques

- Studying existing documentation
- Scenarios
- Prototyping



Elaboration

- The information gathered from Inception and Elicitation are **refined** and **expanded**
- **Creating of user scenarios** on how they interact with the system



Negotiation

- Customers always try to ask more for the given resources and there might be conflicting requirements coming from different users
- Negotiate to resolve those conflicts



Negotiation

- Prioritizing the requirements, assessing the cost and risk
- Requirements are combined, modified or eliminated



THERE SHOULD BE ***NO WINNER OR
LOSER*** IN AN EFFECTIVE NEGOTIATION



Specification

- All requirements are documented i.e.
 - Functional requirements
 - Non-functional requirements
- This document is known as ***Software Requirements Specification (SRS)***
Document



Software Requirement Specification

- Description of a software system to be developed
- Contains
 - All requirements that reduce the development time and cost
 - All requirements that minimizes the risk
 - All functional and non-functional requirements



Software Requirement Specification - Template

- Introduction
- Overall Description
- System Features
- External Interface Requirements
- Other Non-functional Requirements
- Other Requirements



Validation

- Examination of specification to ensure
 - All requirements have been stated
 - All inconsistencies, omissions and errors have been detected and corrected
- A technical review is carried out involving review team, customers, end-users and other stakeholders



Requirements Management

- Ensure product development goals are successfully met
- Technique for documenting, analyzing, prioritizing, and agreeing on requirements - engineering teams can have current and approved requirements
- Avoid errors by tracking changes in requirements and fostering communication with stakeholders from the start to the end



THANK YOU!