UNIT-4

Requirements Engineering

⊛. User Requirement:

User requirements are the statements in natural language plus diagram of what services that the system is expected to provide and the constraints under which it must operate. These are often referred to as user needs, describe what the user does with the system. User requirements are generally documented in user requirement document using narrative text.

An important and difficult step of designing a software product is determining what the user actually wants it to do. This is because the user is often not able to communicate the entirety of their needs and the information they provide may also be incomplete, inaccurate and self-conflicting. This is why user requirements are generally considered seperate from system requirements.

⊛ System Requirement:

System requirements are more detailed descriptions of the software system's functions, services and operational constraints. The system requirements document should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.

Readers of user requirement are Client Manager, System and user, Contractor manager, and System Architects. But readers of system requirement are System and user, Client engineer, Software developers, and System Architect.

⊛. Functional Requirement: Functional requirements are the statement of the services that the system must provide. Functional requirements define the basic system behaviour. They are what the

system does or must not do, and can be thought in terms of how the system responds to inputs. Functional requirements are features that allow the system to function as it was intended. If the functional requirements are not met, the system will not work.

Functional requirements are the main things that the user expects from the software. For e.g, if the application is banking application, that application should be able to create new account, update account, delete a account etc. The functional requirement for the system should be both complete and consistent. Completeness means that all services required by the user should be defined and consistency means that requirement should not have contradictory meaning.

Examples of Functional Requirements:

→ The software automatically validates customers against the Contact Management System.
→ The Sales system should allow users to record customers sales.
→ Only Managerial level employees have the right to view revenue data.
→ The software system should be integrated with banking API.

Advantages of functional Requirement:

→ It helps us to check whether the application is providing all the functionalities that were mentioned by user.
→ It helps us to define the functionality of a system or it's sub-system.
→ Errors caught in the functional requirement gathering stage are the cheapest to fix.
→ Support user goals, ~~but~~ tasks or activities for easy project management.

⊛ Non-Functional Requirements:

Non-functional requirement are requirements that are not directly concerned with specific functions delivered by the system but they are concerned with developing system properties, such as reliability, response time, security, safety etc. Non-functional requirements specify how the system do it. Non-functional requirements do not affect the basic functionality of the system. Even if the non-functional requirements are not met, the system will still perform its basic purpose.

Examples of Non-Functional Requirements:

→ Users must change the initially assigned login password immediately after the first successful login.
→ Employees never allowed updating their salary information. Such attempt should be reported to the security administrator.
→ A website should be capable enough to handle millions of users without affecting its performance.
→ The software should be portable. So moving from one OS to other OS does not create any problem.

Advantages of Non-Functional Requirement:

→ The non-functional requirements ensure the software system follow legal and compliance rules.
→ They ensure the reliability, availability, and performance of the software system.
→ They ensure good user experience and ease of operating the software.
→ They help in formulating security policy of the software system.

Types of Non-Functional Requirements:

1). **Product Requirement:** The requirements which specify the product behaviours are called product requirements. The product's requirements include all functions, features and behaviours that the product must possess, so that it will be efficient, ease to use, safe, low cost etc.

2). **Organization Requirement:** The requirements which are derived from politics and procedures in the customer's and developers organization are called organizational requirement. It includes the process standard used, implementation such as programming language, design methods etc.

3) **External Requirement:** The requirements that are derived from the factors external to the system and its development process. It ensures that whether the system operates with law or not, whether it will be acceptable to its users and general public or not.

⊛ Differences between Functional and Non-Functional Requirements:

Functional Requirements	Non-functional Requirements
i) A functional requirement defines a system or its component.	i) A non-functional requirement defines the quality attribute of a software system.
ii) It specifies "What should the software system do?"	ii) It specifies "How should the software system fulfill the functional requirements?"
iii) It is specified by user.	iii) It is specified by technical peoples.
iv) It is mandatory	iv) It is not mandatory.
v) It helps us to verify the functionality of software.	v) It helps to verify the performance of the software.
vi) Usually easy to define	vi) Usually more difficult to define.

✲ Requirement Engineering Process:

It is a process that involves all of the activities required to create and maintain a system requirement document. The purpose of requirements engineering is to make the problem that is being stated clear and complete, and to ensure that the solution is correct, reasonable, and effective. The process used for requirement engineering varies widely depending on the application domain, the people involved and the organization developing the requirement. However, following are common to all processes:

1) Feasibility Study: Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on it's maintenance are considered during the feasibility study.

(10 marks मा सोधे types पनि लेख्ने 5 मा skip गरे short मा लेख्ने)

Technical Feasibility: It includes the hardware and software required to complete user requirements in the software within allocated time and budget.

Economic Feasibility: Economic feasibility determines whether the required software is capable of generating financial gains for an organization.

Operational Feasibility: It involves the extent to which the required software performs a series of steps to solve user requirements.

2) Requirement Elicitation and Analysis:

Elicitation is gathering of all the system requirements from stakeholders. Requirement Elicitation is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders. Following are four main activities of requirements elicitation and analysis:

Requirements Discovery: Requirement discovery is the process of interacting with, and gathering the requirements from, the stakeholders about the required system and the existing system.

Requirement classification and Organization: It is the process of putting related requirements together, and decomposing the system into sub components of related requirements.

Requirement Prioritization and Negotiation: This activity is concerned with prioritizing requirements and finding and resolving requirements conflicts through negotiations until we reach a situation where some of the stakeholders can compromise.

Requirement specification: In this step, requirements are checked to discover whether they are complete, consistent or not and they are documented and input in to the next round of spiral.

3) Requirement Specification: The management of the organization studies feasibility report and suggests the modifications in the requirement if any. Knowing the constraints on available resources and modified requirements specified by organization, final specification of the system to be developed is drawn up by the system analyst.

4) Requirement Validation: Requirement validation is the process of checking the requirements for validity, consistency, completeness, realism and verifiability. It is concerned with showing that the requirement actually define the system which the customer wants. Requirement validation process consists of different types of checks like validity check, completeness check, consistency check etc.

5) Requirement Change Management: Requirement management is the process of managing changes to a system requirement. It is the process of understanding and controlling changes to system requirement. It is essential because we need to decide if the benefits of implementing new requirements are justified by the costs of implementation.

Q. What are the different types of requirement elicitation techniques? Explain in brief.

Ans: Requirement elicitation techniques are:

1) Document analysis: Document analysis is one of the most important technique in understanding the current process. Documents like user manuals, software vendor manuals, process documents about the current system can provide the inputs for the new system requirements.

2) Observation: The elicitation technique helps in collecting requirements by observing users or stakeholders. This can provide information about the existing process, inputs and outputs.

3) Interview: An interview is a systematic approach to elicit information from a person or group of people. In this case, the business analyst acts as an interviewer.

4) Questionaries: Questionaries are useful when there is a lot of information to be gathered from a larger group of stakeholders. This enables the business team to gather requirements from stakeholders remotely.

5) Prototyping: Prototyping is building user interface without adding detail functionality for user to interpret the features of intended software product. It helps in giving better idea of requirements.