

ASSIGNMENT 1



Unit - 1 & 2. Introduction to Software Engineering and Software Processes & software Modeling

Q.1

Explain dual role of software.

The dual role of software are the two primary aspects that software serves in the world of computing and technology, which are:

i) User - Facing Functionality

Software provides a user interface through which end - users can interact with computers, devices or applications through GUIs, CLI or even voice commands. It is responsible for delivering specific functionalities and features to end - users , from word processing , calculation to playing video games , managing emails.

ii) Machine - Facing Functionality

Software also plays a crucial role in controlling and coordinating hardware components and system resources like CPU, memory , storage and input/output devices to ensure that they work together effectively efficiently. It often automates tasks and processes , reducing the need for manual intervention. Also for processing and manipulating data . (sorting , filtering , analyzing).

Q.2

What is Software ? Describe Software Engineering
Software refers to a collection of logically organized instructions , data and programs that enable a computer or other devices to

I. INTRODUCTION

perform specific task or functions. There are broadly two ^{main} types, a software can be divided into:

- ① System Software : Responsible for managing and controlling the computer hardware
- ② Application Software : Designed for specific tasks or applications.

-4 Software Engineering is a discipline or field of study focused on the systematic design, development, testing and maintenance of software. It encompasses a set of principles, methods and practices that aim to create high-quality software efficiently and reliably. It also involves the best practices to manage the entire Software Development Life Cycle, from requirement gathering to deployment.

Q.3 Enlist and Explain process framework activities.

-4 Process framework activities refer to the fundamental stages that are typically involved in software development processes. It provides a structured approach to managing and executing software projects.

i) Communication

This initial activity involves establishing effective communication among project stakeholders for identifying and understanding

the needs of and requirements of the project as well as setting up clear lines of communication.

ii) Planning

In the planning phase, project managers and teams define project objectives, scope, budget and timelines. They create a detailed project plan that outlines the tasks, resources and schedules required to achieve project goals.

iii) Modeling

Modeling involves creating various models or representations of the software to understand its structure, behavior and design which may include creating diagrams, prototypes and specifications.

iv) Construction

This is the phase where actual coding and development of the software take place. Based on design specifications and build the software according to the project plan.

v) Testing

Here, the software is systematically tested to identify and fix defects, verify that it meets the specified requirements and ensure its quality and reliability.

vi) Deployment

This last part involves installing and configuring the software in the target environment. It may also include data migration, user training and setting up production servers.

Q.4
—4

Enlist and Explain Umbrella Activities.

Umbrella Activities are high-level activities that span across multiple phases of a software development process or project. They help in managing and ensuring the quality of the project.

i) Project Management

It involves planning, organizing and controlling various aspects of the project including scope, schedule, budget, resources and risk. This ensures that the project stays on track, meets its objectives and adheres to the planned timelines and budget.

ii) Quality Assurance and Control.

Quality assurance activities focus on defining quality standards and processes, while quality control activities involve monitoring and verifying that the project adheres to these standards.



iii) Configuration Management

It involves tracking and controlling changes to project artifacts, including source code, documentation and other assets. It ensures that all project elements are versioned, documented and properly managed.

iv) Risk Management

It identifies potential risks and uncertainties that could affect the project's success and minimizes the impact of it. These risks are analyzed, assessed and strategies are developed to mitigate or respond to them.

v) Measurement and Metrics

It involves defining and collecting project metrics and measurements to track progress, assess quality and make data driven decisions. Metrics can include lines of code written, defect counts and progress against milestones.

vi) Documentation

It involves creating, organizing and maintaining project documentation. This includes design specifications, user manuals and other project-related documents.

vii) Training and Knowledge Transfer

It focuses on providing training & knowledge

transfer sessions to team members, users and stakeholders. Training ensures that individuals have the skills and knowledge needed to use and maintain the software.

viii) Reviews and Audits.

They are formal assessments of project artifacts and processes to identify issues, ensure compliance with standards and validate that project objectives are being met.

Q.5 Difference between Software process and product.

-4	Software Process	Software Product
>	The focus is on how software is developed and managed.	The focus is on what the software does and how it delivers value to users.
>	The process tends to be long-term	A product tends to be short-term.
>	The purpose of the process is to make the quality of the project better.	The main goal of the product is to complete the work successfully.
>	Generally only one person uses, hence there is a lack of user interface.	Good GUI is must required by any software product.

- > The output of process is product Product development occurs by following process.
- > eg: Program developed for parsing the input eg: A word processing software

Q.6 Outline the software development life cycle. Briefly describe each of the stages, its relation to other stages and its overall importance.

—> The SDLC is a structured framework that describes the processes and stages involved in developing software applications or systems.

i) Requirements Gathering and Analysis.

In this stage, the development team works closely with stakeholders to gather and analyze requirements. The goal is to understand what the software needs to accomplish, its functional and non-functional requirements and any constraints or limitations.

ii) Planning and Feasibility

During planning, project managers create a detailed project plan defining project scope, estimating resources, scheduling tasks and assessing the project's feasibility in terms of budget, technology and time. It helps in

maintaining the project well-organized, resources are allocated efficiently and potential risks are identified.

iii) System Design (Modeling).

In this stage, architects and designers create a high-level design of the software system which defines the system's architecture components, data structures and interfaces.

iv) Implementation (Coding and Testing).

Here, developers write code based on the design specifications where the software is built and programming languages are used to translate the design into functional code.

Testing involves systematically verifying and validating the software to identify and fix defects, ensure it meets requirements and maintains quality standards.

v) Deployment

It includes installing and ~~configuration~~ the software in the target environment. It may also involve data migration, user training and setting up production servers. It ensures that the software is accessible to users and functions correctly in the production environment.

Q.7 Compare Prototype and RAD process Model.

→

Prototype Model

RAD Model

<ul style="list-style-type: none"> > A process that involves building a working model of the software to get feedback from users before the final product is developed. > Focuses on user's needs and requirements > Build a prototype, get feedback, refine the prototype and repeat until the user is satisfied > Higher risk, as the final product may not meet the user's needs > Suitable for projects where requirements are not well-defined or there is a high risk of changes to the requirements 	<ul style="list-style-type: none"> > A process that involves combines prototyping and iterative development to deliver software quickly and efficiently. > Focuses on delivering the software quickly & efficiently > Develop in iterations, with each iteration delivering a working version of the software. > Lower risk, as the software is developed iteratively and the user's feedback is incorporated > Suitable for projects where the requirements are well-defined and there is a need to deliver the software quickly

Q.8

Explain Spiral model and describe its advantages over waterfall model.

→

The Spiral model is a software development process model that combines elements of iterative development and risk management. It is characterized by its iterative approach, which allows for incremental development and frequent opportunities for stakeholder feedback. It is well-suited for large, complex projects where requirements may be uncertain or subject to change.

The Spiral Model is structured around a series of iterative cycles or "spirals", which consist of planning, risk analysis, engineering and evaluation.

→

Advantages of the Spiral Model Over the Waterfall Model:

i) Iterative and Incremental.

It allows for the development of smaller, manageable portions of the software in each cycle. This contrasts with the waterfall model, where it follows a sequential process.

ii) Risk Management

It places a strong emphasis on risk analysis and management. It identifies and address risk in each cycle, and it handle uncertainties, changing requirements. In contrast, waterfall model assumes that requirements are



stable from the outset.

iii) Client Involvement.

Frequent evaluations and feedback gathering from ~~stakeholders~~ stakeholders occur in the Spiral Model. In contrast, waterfall model, client feedback often comes late in the process, making it more challenging to address changes.

iv) Flexibility

The Spiral Model is adaptable and can accommodate changes more readily compared to the Waterfall Model, which can be rigid when requirements evolve.

Q.9 Explain Concurrent Development Model.

→ The Concurrent Development Model, is a software development approach that emphasizes parallel and simultaneous activities across various phases of the SDLC. It is designed to accelerate the software development process and improve efficiency by allowing multiple tasks to occur concurrently, rather than sequentially.

• Key Characteristics of this model are:

- i) Parallel Activities : Multiple phases of the SDLC can occur in parallel, rather than one after the other.
- ii) Cross-Functional Teams : Consisting of members with

diverse skills, collaborate closely to perform simultaneously.

- iii) Client Involvement : Involvement is encouraged throughout the development process. Feedback is solicited and incorporated into ongoing development.
- iv) Risk Management : Teams work to identify potential risk early in the process and take proactive measures to mitigate them.
- v) Iterative and Incremental : It allows for the development of smaller increments of the software in parallel, with each increment building upon the previous one.

Q.10

Explain Agile Process Model. State the significance to use agile framework.

→ 4

The Agile Process Model is a set of principles and values for software development that emphasizes flexibility, collaboration, customer-centricity and iterative progress. Key principles of this model are :

- i) Customer Collaboration : Agile places a high value on collaborating closely with customers and stakeholders to understand their needs and preferences to continuously adapt the product to meet those needs.

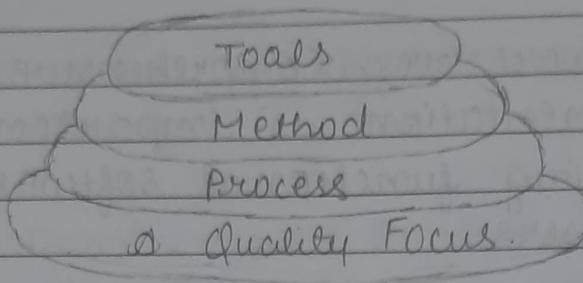


- ii) Working Software over Comprehensive Documentation: While documentation is important, agile focuses on delivering functional software that adds value to users.
- iii) Individuals and Interactions over Processes and Tools: Agile values people and their interaction more than the processes and tools they use.
- iv) ^{Sustain} Sustainable Development: Agile promotes a sustainable pace of work for the development team to ensure long-term productivity and quality.
- v) ^{Team} ~~base~~ collaboration: Agile teams are often self-organizing and cross-functional, with members collaborating closely to deliver working software.

- 4 Significance of Using Agile Framework:
- Adaptability to Change
 - Reduced Risk
 - Continuous Improvement
 - Efficiency and Productivity
 - Flexibility

Q.11 Explain Software as a layered technology.

- 4 Software as a layered technology refers to the concept of organizing and structuring software system into distinct layers, each with a specific set of responsibilities and functionalities.



i) Quality Focus

It defines the continuous process improvement principles of software. It provides integrity that means providing security to the software so that data can be accessed by only an authorized person.

ii) Process

It is the foundation of software engineering. It is ^{the} key that binds all the layers together which enables the development of software before the deadline. It defines a framework that must be established for the effective delivery of software.

iii) Method

All the answers of "how-to-do" questions are given by method. It has the information of all tasks which includes communication, requirement analysis, design modeling, program construction, testing and support.

iv) Tools

It provide a self-operating system for processes and method.

Q.12 Define Requirement Engineering. List and explain requirements engineering tasks.
→ Requirement Engineering is a systematic and disciplined approach to defining, documenting and managing software requirements throughout the entire SDLC. It is a critical phase that focuses on understanding, capturing and specifying the needs and expectations of stakeholders to ensure that the final product meet their objectives.

i) Elicitation.

It is the process of gathering requirement from stakeholders through interviews, surveys and observations, are used to extract information about their needs and expectations.

ii) Analysis and Documentation.

The requirements are then documented in a clear and unambiguous manner which may include use cases, user stories, requirement specifications or more formats.

iii) Requirements Prioritization.

It involves ranking requirements based on their importance, ~~difficulty~~ criticality and relevance to project objectives.

iv) Communication

Effective communication with stakeholders,

including regular updates and feedback sessions, is a continuous task in requirements engineering. It ensures that all parties remain informed and aligned throughout the project.

Q.13 Explain Functional and Non-Functional Requirements for Hotel Management System.

- A Hotel Management System is a software application designed to streamline the operations and management of a hotel establishment.

- Functional Requirements : It describes what the system should do in terms of specific functions, features and interactions.
 - > User Registration and Authentication
 - > Room Reservation
 - > Check-In and Check-Out
 - > Billing and Invoicing
 - > Reservation Modifications and Cancellations.

- Non-Functional Requirements : It defines the qualities or characteristics of the system. They describe how well the system should perform its functions.
 - > Performance
 - > Security
 - > Reliability
 - > Maintenance and Support



Q.14 What is Software Requirement Specification (SRS)? Why is it important? List the characteristics of a good quality SRS? What contents can we include in it?

→ Software Requirement Specification (SRS), is a comprehensive document that serves as a formal, written representation of the software system's requirements. It outlines what the software is expected to do, its functionalities, constraints, and the quality attributes it should possess.

→ Importance of an SRS:

- i) Communication: It acts as a common point of reference for all stakeholders which ensures that everyone has a shared understanding of what the software should achieve.
- ii) Scope Definition: The SRS helps define the project's scope by detailing the functionalities and features of the software.
- iii) Testing: It creates test cases and verify that the software meets the specified requirements.
- iv) Client Expectations: Clients and stakeholders can use the SRS to validate that the software aligns with their expectations.

→ Characteristics of a Good Quality SRS:
➢ Clear and Unambiguous

- > Consistent
- > Verifiable
- > Feasible
- > Prioritized.

-4) Contents of an SRS:

- > Introduction
- > Functional Requirements
- > Non - Functional Requirements
- > Constraints.
- > System Overview

Q.15 Explain Requirement Analysis with example.

-4) Requirement Analysis is a crucial phase in software engineering and system development. It involves a systematic process of gathering, documenting, validating and prioritizing user needs and system requirements. The goal of requirement analysis is to ensure a clear and complete understanding of what the software should accomplish and what stakeholders expect from it.

• Example : Online Shopping System.

i) Identify stakeholders.

Identify all relevant stakeholders who will interact with or be affected by the online shopping system.



ii) Gather initial Requirements

Conduct interviews and surveys with stakeholders to understand their needs and expectations.

iii) Create use Cases.

Create use cases that describe specific interactions or scenarios involving the system.

iv) Analyze and Prioritize Requirements

Work with stakeholders to prioritize requirements. Some may be critical, while others may be nice-to-have.

v) Identify Constraints

Identify any constraints that the project must adhere to, such as budget, timeline or compliance with legal regulations.

vi) Validate and Verify Requirements

Create a prototype of the user interface and share requirement specification with stakeholders for review and feedback.

vii) Maintain Traceability

Maintain a requirement traceability matrix that links requirements to design elements, test cases and other project artifacts.

Q.16

Enlist characteristic of SRS. write a SRS for college management system.

→ Characteristic of a SRS document are:

- > Clarity
- > Consistency
- > Completeness
- > Traceability
- > Testability
- > Verifiability
- > Change Management

→ A SRS for College Management System :

i) Introduction

> Purpose : ~~Designed to streamline and automate various administrative and academic processes.~~

> Stakeholders : Students, Faculty, Administrative staff and college management

ii) System Overview

> Functional : Provide student registration, grading, course scheduling and report.

> Non-Functional : ^{Prioritize} Security, scalability and user-friendliness

iii) Functional Requirements

> User registration and authentication, student management, course management, attendance tracking, examination management.



(iv) Non-Functional Requirements

- > Security, Performance, Scalability

(v) Constraints

- > Should Budget is limited to \$100,000
- > Development timeline is six months.

Q.17

Draw use case diagram for Hospital Management System.

→

use case Diagram for HMS.

