Date .....

Software Engineering

Q-1
Ans

1) SDLC model : Agile
=> Agile offers flexibility, iterative development, stakeholder collaboration, and is suitable for hybrid development technology

2) Use Case Diagram:
- Primary Actors - Estheticians, clients
- Supporting Actors - Rating and Review system, Payment System, Account Management System
- Actor goals - Estheticians - offer beauty services, manage services and account; clients - confirm estheticians, book ~~user~~ services, rate and review ~~user~~ services, select payment methods.

3) a) Differentiation between system and software Engineering
- System Engineering - Involves designing and managing complex systems and processes to meet specified requirements. It encompasses the entire system lifecycle, including hardware, software and human factors. For example, designing a traffic management system involves considering traffic flow, road infrastructure and human behaviour.

Spiral

Date .....

- Software Engineering: focuses on designing, developing, and maintaining software systems. It deals with software-specific aspects such as requirement analysis, design, coding, testing, and maintenance. For example, developing an e-commerce application involves designing user interfaces, implementing payment systems.

b) Software Process model for WMZU system: Agile

- Incremental development allows for early delivery of essential features and customer collaboration ensures that the system meets stakeholders needs.

c) Requirement Elicitation Techniques:

- Interviews: Direct interaction with stakeholders to gather needs and preferences

- Surveys - Collecting feedback from a larger audience to understand preferences and habits.

Spiral

Date .....

Q-2

Ans 2

i) SDLC for each case study:

a) Case Study 1:-

- SDLC Model : Agile
- Agile allows for iterative development and accommodates changes in requirements, which is suitable for a project where requirements may evolve as stakeholders gain a clearer understanding of the system's need.

b) Case Study 2 -

- SDLC :- Incremental or Iterative
- Incremental or Iterative models allow for early and periodic deliverables, while also accomodating changes to scope / requirements. They enable the customer to be involved throughout the project and provide feedback and evaluation.

c) Case Study 3 -

- SDLC : Waterfall
- Waterfall model ensures thorough planning and documentation, which is important for a critical system like a central pharmacy. It follows a structured and sequential approach, suitable for the well-defined process and stringent requirements of the healthcare industry.

Date .....

Q-3

Ans 3 → CMM Framework in Software Engineering:

The Capability Maturity Model (CMM) framework is a process improvement approach that defines five maturity levels for software development organizations, ranging from Initial (Level 1) to Optimized (Level 5). It focuses on improving processes to enhance product quality, reduce risks, and increase efficiency.

→ Use of CMM Framework:

- CMM provides a structured framework for assessing and improving an organization's software development process.
- By following the CMM framework, organizations can identify areas of improvement, ~~est~~ establish best practices, and standardize process.
- It helps organizations in setting realistic goals, measuring progress, and achieving higher levels of maturity over time.
- CMM encourages continuous improvement and learning, leading to enhanced productivity and customer satisfaction.

Date .....

=> Example: Consider a software development company that adopts the CMM framework. Initially, the company operates at level 1, where processes are ad-hoc and chaotic. As the company implements CMM practices and progresses to higher maturity levels, such as level 3 or level 4, it experiences significant improvements in process consistency, project management and product quality.

Q-4

Ans => SEI CMM and Increasing levels of Maturity:

The SEI CMM specifies five levels of maturity for software development organizations, ranging from Initial to Optimized. As the maturity levels increase, the organizations software development processes become more structured, disciplined, and optimized, leading to improved product quality and project success rate.

=> CMM level 5 without level 1:

It's theoretically possible for an organization to achieve CMM level 5 without going through the lower maturity levels, but it's highly unlikely and not recommend. The lower maturity

Spiral

Date .....

levels provide the ~~foundation~~ foundational elements necessary for establishing disciplined processes and organizational capabilities. Attempting to reach levels without addressing the fundamental process deficiencies at lower levels would likely result in inefficiencies, inconsistencies and limited sustainability.

» Human factors for Agile Process:
Two human factors that contribute to the success of Agile process are:

1. Collaborative Culture - Agile methodologies emphasize teamwork, open communication, and collaboration among cross-functional teams, stakeholders, and customers. A collaborative culture fosters creativity, innovation and collective ownership of project outcomes.

2. Adaptability and Empowerment: Agile empowers team members to take decisions, take ownership of tasks, and adapt to changing requirements and priorities. It encourages individuals to embrace change, experiment with new ideas, and continuously improve their work processes.

→ Problem Faced During Requirement Elicitation:

One common problem faced during requirement elicitation is the uncertainty or ambiguity in requirements. Stakeholders may have differing perspectives, priorities, and expectations, leading to inconsistent or unclear requirements. Additionally, stakeholders may struggle to articulate their needs effectively, resulting in incomplete or inaccurate requirements documentation.

Q.5 → Software Process Model Effectiveness -

Ans The effectiveness of software process models depends on various factors, including project characteristic, organizational culture, and stakeholder requirements. Both incremental and linear sequential (Waterfall) models have their advantages and limitations, making them suitable for different contexts.

→ Reasons for Effectiveness -

The incremental model is more effective in scenarios where requirements are not fully understood upfront and iterative development is preferred to accomodate changes and uncertainties. It allows for the incremental delivery of functionality, enabling early feedback from stakeholders and mitigating risks associated with changing requirements.

Date .....

Q-6

```
#include <stdio.h>

int main() {
    double first, second, temp;
    printf("Enter first number: ");
    scanf("%.lf", &first);
    printf("Enter second number: ");
    scanf("%.lf", &second);

    // value of first is assigned to temp
    temp = first;
    // value of second is assigned to first
    first = second;
    // value of temp is assigned to second
    second = temp;
    // %.2lf displays number up to 2 decimal points
    printf("After swapping, first number = %.2lf\n", first);
    printf("After swapping, second number = %.2lf\n", second);
    return 0;
}
```

→ Calculations

- Lines of code = 16
- Halstead Matrics:
  - → Program length (N) = 8
  - → Program Vocabulary (n) = 5
  - → Program Volume (V) = $N * \log_2(n) = 8 * \log_2(5) \approx 26.51$
  - → Program difficulty (D) = $(n_1/2) * (n_2/N_2) = (6/2) * (1/3) \approx 1.00$
  - → Program Effort (E) = $D * V = 1.00 * 26.51 \approx 26.51$

Spiral

Date .....

Q.7

Ans7 Black box Testing:

Black box testing is a software testing technique where the tester examines the functionality of an application without knowing it's internal structure, design, or implementation details. It focuses on testing the software's external behaviour based on it's specifications and requirements. There are various types of black box testing strategies:

a) Equivalence Partitioning - In this technique, the input domain of the application is divided into equivalence classes, and a representative value from each class is chosen as a test case. For example, if an input field accepts values between 1 and 100, equivalence partitioning would select value like 0, 50 and 101 for testing.

b) Boundary Value Analysis - This technique focuses on testing the boundaries of input ranges. Test cases are designed to include the boundary values, as these values are more likely to cause errors.

c) Decision Table Testing - Decision tables are used to model complex business rules or logic. Test cases are derived based on different combinations of input conditions and their corresponding actions or outcomes.

Spiral

Date .....

d) State Transition Testing - This technique is used for systems that exhibit different behaviours based on their states or conditions. Test cases are designed to verify the transitions between different states of the system.

d-8

Ans Spiral Model and Change Management:

The spiral model is an iterative software development process model that combines elements of both waterfall and iterative development methodologies. It supports both ~~support~~ change avoidance and change tolerance through it's iterative nature and risk-driven approach.

- Change Avoidance - Each iteration ~~in~~ involves risk analysis and mitigation. By identifying and ~~dd~~ addressing risks early in the development process, the likelihood of significant changes later on is reduced.

- Change Tolerance - Despite risk mitigation efforts, changes may still occur during the development process. The spiral model accomodate changes by allowing for flexibility and adaption in each iteration.

Spiral

Date .....

Q. 9

Ans 9 A high-quality Software requirement Specification (SRS) is essential for ensuring the development of a high-quality software product.

- Clear understanding - A well-written SRS provides a clear and comprehensive understanding of the software requirements to all stakeholders, including developers, testers, and clients. This clarity helps in avoiding misunderstandings and misinterpretations during the development process

- Basis for Development - The SRS serves as the foundation for the entire software development process. It outlines the functional and non-functional requirements, constraints, and expectations from the software

- Minimizing Rework - A detailed and precise SRS helps in minimizing rework during the development lifecycle. By clearly defining the scope and functionality of the software upfront, the likelihood of significant changes or revisions later in the development process is reduced, leading to cost and time savings.

- Client Satisfaction - A high quality SRS ensures that the final software product meets the expectations and needs of the clients or

Date .....

end - exercises.

Q-10

Ans 10 Life cycles models for Projects -

- Library management System: For a library management system, the most suitable life cycle model would be the Waterfall Model. This is because the requirements for such a system are generally well-defined and stable upfront, allowing for a sequential and systematic development approach. Libraries often have strict budgets and time constraints, making the linear progression of Waterfall model advantageous.

- Web Application - For developing a web application, the Agile model would be more appropriate. Agile methodologies, such as Scrum or Kanban, are well-suited for projects with evolving requirements and a need for frequent iterations and feedback.

Q-11

Ans 11 Cohesion in University

→ Registration System: The described subsystems in the University registration system exhibit low cohesion as they combine unrelated functionalities within each sub-system. To

Date .....

improve cohesion and adhere to the principle of cohesion, the subsystems should be redesigned to focus on performing a single, well-defined task.

- Subsystem A -
  Improvement - Separate the functionalities into distinct subsystems. For example, create separate subsystems for displaying course lists, registering students, checking for schedule conflicts and eligibility, and database management.

- Subsystem B:
  Improvement - Similarly, divide the functionalities into separate subsystems based on their distinct purposes. For instance, create separate modules for entering student grades, assigning courses to faculty, managing course details, and generating student bills. This separation of concerns will improve the cohesion of each subsystems and make the system easier to maintain and understand.

Spiral

Date .....

Q-12

Ans 12

→ Project Plan:

Objective - Implement an online transaction processing system for a hospital to handle the transactions of up to 200 inpatients efficiently.

Phases - Requirement Analysis, System Design, Implementation, Testing, Deployment, Maintainence

Duration - 6 Months

→ Schedule:

- Requirement Analysis - 1 month
- System Design - 2 month
- Implementation - 2 months
- Testing - 1 month
- Deployment - 1 month

→ Functional Requirements

- Patient Admission
- Billing and Payment
- Medical Records Management
- Inventory Management
- Appointment Scheduling

Date .....

Q-

	WF (High)
User input = 55	6
User Output = 35	7
User Enquiries = 50	6
User Files = 8	15
External Interfaces = 5	10

weighting factors are high
adjustment factor is ~~not~~ significant =>

$$UFP = 330 + 255 + 240 + 120 + 50$$
$$= 985$$

$$CAF = [0.65 + 0.04 \Sigma f_i)$$
$$= [0.65 + 0.01(14 \times 5))$$
$$= [0.65 + 0.56)$$
$$= 1.21$$

$$FP = UFP \times CAF$$
$$= 985 \times 1.21$$
$$= 1191.85$$

Date .....

		WF (Average)
Q-	Input = 10	4
	Output = 30	5
	Enquiry = 50	4
	Interface = 10	10
	use file = 20	7

CAF is 8

UFP = 40 + 150 + 200 + 100 + 150
= 630

FP = UFP x CAF
= 630 x 8
= 5040

Q-

user input	=	5 x 3
user output	=	5 x 5
user enquiries	=	6 x 6
files	=	5 x 7
external interfaces	=	5 x 7

CAF ⇒ significant (4)

UFP = 15 + 25 + 36 + 35 + 35
= 146

CAF = (0.65 + 0.01 ΣF_i) = 0.65 + 0.01 (14 x 4)
= 0.65 + 0.56
= 1.21

FP = UFP x CAF
= 146 x 1.21 = 176.66

Signal