

Part A:

1	Agile Software Development is _____ a) Incremental Development b) Iterative Development c) Linear Development d) Both Incremental and Iterative Development
2	A software development methodology that is formulated to the responsiveness to changing customer requirements and also to improve software quality _____. a) Estimate programming b) Extreme programming c) Functional programming d) Embedded programming
3	A _____ is a collection of activities, actions, and tasks that are performed when some work product is to be created. a) Stack b) Process c) Task d) Schedule
4	What is Software Engineering? a) Designing a software b) Testing a software c) Application of engineering principles to the design a software d) Planning to build a software
5	What is the major drawback of the Spiral Model? a) Higher amount of risk analysis b) Does not work well for smaller projects c) Additional functionalities are added later on d) Strong approval and documentation control
6	Which one is not a size measure for software product? a) LOC b) Halstead's program length c) Function Count d) Cyclomatic Complexity
7	As per an IBM report, "31% of the project get cancelled before they are completed, 53% overrun their cost estimates by an average of 189% and for every 100 projects, there are 94 restarts". What is the reason for these statistics? a) Lack of adequate training in software engineering b) Lack of software ethics and understanding c) Management issues in the company d) All of the mentioned

8	<p>_____ is the process of interacting with stakeholders of the system to discover their requirements.</p> <p>a) Requirement discovery b) Requirement engineering c) Change engineering d) Software engineering</p>
9	<p>Given, Optimistic LOC, $S_{opt} = 2600$ Pessimistic LOC, $S_{pess} = 4000$ Most Likely LOC, $S_m = 3800$ Calculate the expected value for the estimation variable</p> <p>a) 3000 b) 3200 c) 3633 d) 3666</p>
10	<p>→ The estimated LOC count is 56,100</p> <ul style="list-style-type: none"> ◆ Average productivity for this category of systems = 693 LOC/pm, Burdened labor rate = \$8000/month ◆ Calculate cost per LOC <p>a) \$11.7 b) \$12.5 c) \$13.5 d) \$11.5</p>

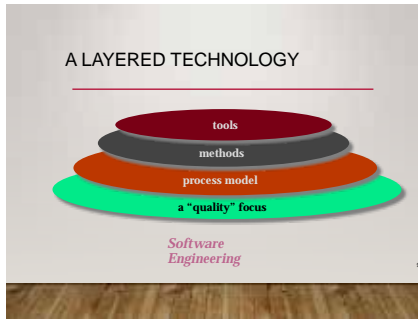
Part B:

1. **Software engineering is a layered technology – Justify this statement and briefly describe with an illustration**

Answer: **ANSWER-**

Software engineering is a layered technology.

- The foundation for software engineering is the process layer.
- Process defines a framework that must be established for effective delivery of software.
- Software engineering methods provide the technical how-to's for building software.
- Methods encompass a broad array of tasks that include communication, requirements analysis, design modeling, program construction, testing, and support.
- Software engineering tools provide automated or semi automated support for the process and the methods.
-



2. Outline the umbrella activities of a software process?

Answer:

- Software project tracking and control.
- Risk Management.
- Software Quality Assurance.
- Formal Technical Reviews.
- Software Configuration Management.
- Work product preparation and production.
- Reusability management, Measurement.

3. Compare and contrast the advantages of Spiral vs. Prototype. Justify with a case study suitable for Prototype but not suitable for Spiral

ANSWER -

Prototyping – Advantages

1. Provides working model.
2. Customer is highly satisfied with such a modeling at initial stages
3. Developer gains business insight, reducing ambiguity
4. Great involvement of users
5. Reduce risks

Spiral – Advantages

1. Applies throughout lifecycle
 - Concept Development
 - New Product Development
 - Product Enhancement
2. Risk is considered at each pass
3. Uses prototyping as risk reduction mechanism
4. Customer and developer understand and better react to risks

Case study: Website Development can be done with prototyping as at every stage the customer can be showcased using Wireframes using tools like Figma. In short the cases where and When your customer has a legitimate need, but is clueless about the details, develop a prototype as a first step to make him visualize what he thinks through your skills.

4. List out the different approaches to collaborative requirements gathering.

Answer:

Interviewing, Ethnography, Stories and Scenarios

5. Estimated LOC count is 56,100. Assuming that your organization produces 450 LOC/pm with a burdened labor rate of \$7000 per person-month, Find the total estimated project cost and estimated effort in person months.

ANSWER –

TOTAL LOC = 56100 LOC

TOTAL EFFORT= $56100/450 = 124.67 = 125$ person/month.

TOTAL COST = $125 * 7000 = \$ 875000$

Part C:

1. Assume that you are a software developer of a company. A client has approached you to for a better solution of the problem faced by their side. The client stated that risks/uncertainties will lead to loss if not properly planned and solve. Justify with a neat pictorial representation, which model will you opt for software development and mention its merits and demerits in detail.

ANSWER -

Selection of Model: Spiral Model / Agile Model can be chosen

Justification to be given

Model Pictorial Representation

Advantage and disadvantage

(OR)

2. Discuss about agile scrum process and their stages with neat diagram. Also identify 4 application which use the Agile model.

ANSWER –

Scrum is an agile software development method that was conceived by Jeff Sutherland and his development team in the early 1990s.

- In recent years, further development on the scrum methods has been performed by Schwaber and Beedle.
- Scrum principles are consistent with the agile manifesto and are used to guide development activities within a process that incorporates the following framework activities: Requirements, Analysis, Design, Evolution and delivery.
- Within each framework activity, work tasks occur within a process pattern called a sprint.
- The work conducted within a sprint is adapted to the problem at hand and is defined and often modified in real time by the Scrum team.
- The overall flow of the Scrum process is illustrated in figure below.
- Scrum emphasizes the use of a set of software process patterns that have proven effective for projects with tight timelines, changing requirements and business criticality.
- Each of these process patterns defines a set of development actions:

1. **Backlog** – a prioritized list of project managements or

features that provide business value for the customer. Items can be added to the backlog at any time. The product manager assess the backlog and updates priorities as required.

Each of these process patterns defines a set of development actions:

2. **Sprints** – consist of work units that are required to achieve a requirement defined in the backlog that must be fit into a predefined time-box (typically 30 days). Changes (e.g., backlog work items) are not introduced during the sprint. Hence, the sprint allows team members to work in a short term, but stable environment.

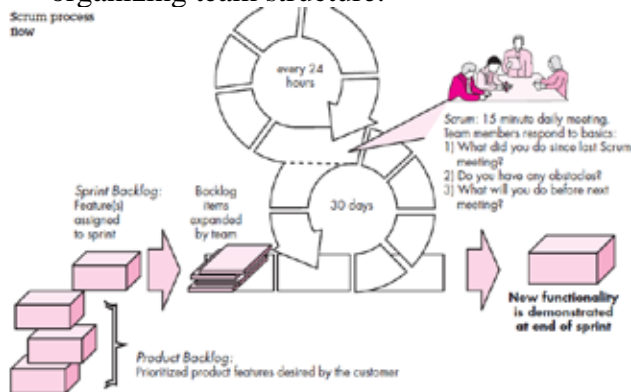
3. **Scrum meetings** – are short (typically 15 minutes) meetings held daily by the Scrum team. Three key questions are asked and answered by all team members:

- a. What did you do since the last team meeting?
- b. What obstacles are you encountering?
- c. What do you plan to accomplish by the next team meeting?

A team leader, called a Scrum master, leads the meeting and assesses the responses from each person.

The Scrum meeting helps the team to uncover potential problems as early as possible.

Also, these daily meetings lead to “knowledge socialization” and thereby promote a self-organizing team structure.



3. Using your knowledge of how an ATM is used, write a set of Functional and Non-Functional Requirements for “ATM System”.

Functional Requirements

1. An ATM machine accepts a card from a user.
2. The user inputs a Personal Identification Number (PIN) to authenticate the user's identity.
3. The system validates the card and the PIN, then Either continues processing or rejects the card.
4. The ATM prompts the validated user for the type of transaction; valid transaction types are as follows:
 - Check account balance
 - Process a deposit
 - Process a withdrawal for an ATM customer or
 - Process a cash advance for a credit card holder
 - Transfer funds
 - Pay bills
5. The ATM communicates the request to the

appropriate financial system

6. The appropriate financial system responds with permission or denial of the request.

7. The ATM asks the user if they want a printed receipt.

8. The ATM acts on the request according to the response received from the financial system. Possible actions for granted requests include the following:

- reject the request
- accept a deposit
- dispense cash
- display or print an account balance
- pay a bill
- perform an Electronic Funds Transfer (EFT)

9. The system updates the bank's financial system for ATM transactions or sends an EFT to the appropriate financial institution for credit card transactions.

10. The ATM prints a receipt if one is requested.

11. The system prompts the user for another transaction and repeats steps 4 – 10 if yes.

12. The ATM closes the session and waits for another user when done.

(OR)

4. (a) Use the COCOMO II model to estimate the effort required to build software for a simple ATM that produces 12 screens, 10 reports, and will require approximately 80 software components, Percentage of reuse is 40%, Value of Prod=9. Use the application composition model with object points. (8 marks)

Object Points Count = 12 + 10 + 80 = 102

New object count(NOP) = **Object Points** * [(100 - % reuse)/100] = 102 * 60/100 = 61.2

Estimated Effort = NOP/PROD = 61.2/7 (LOW) = 8.74 pm

(b) The range of LOC estimates for the 3D geometric analysis function is optimistic, 2600 LOC; most likely, 6900 LOC; and pessimistic, 8600 LOC. Calculate the expected value for the estimation variable. (2 marks)

$S = (S_{opt} + 4 * S_m + S_{pess}) / 6$

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu

College of Engineering and Technology
School of Computing
Department of Data Science and Business Systems

Academic Year: 2023-24 (Even)

 Set: B
 Section:

Test	: CLA-T1	Date	: 19-02-2024
Course Code & Title	: 21CSC303J & Software Engineering and Project Management	Duration	: 1hr 30 Minutes (2 Periods)
Year & Sem	: III Year / VI Sem	Max Marks	: 50

Course Articulation Matrix:

Sl. No.	Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	CO1	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
2	CO2	-	3	-	-	-	-	-	-	2	-	2	-	3	-	-
3	CO3	-	-	-	2	-	-	-	-	2	-	2	-	3	-	-
4	CO4	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
5	CO5	-	-	-	-	-	-	-	-	2	-	3	-	3	-	-

Part – A (10 x 1 = 10 Marks)
Instructions: Answer ALL questions.

Q. No	Question	Marks	BL	CO	PO	PI Code
1	_____ is a controlled process of initiating, planning, executing and closing down a project. A. System design B. System analysis C. Project scheduling D. Project Management	1	1	1	12	12.3.1
2	The prototyping model in software development is A. A reasonable approach when requirements are well defined. B. A useful approach when a customer cannot define requirements clearly. C. The best approach to use for projects with large development teams. D. A risky model that rarely produces a meaningful product.	1	1	1	1	1.3.1
3	You are a member of a Scrum Team, and you are instructed by the General Manager (GM) of your company to work on an urgent task that is not part of the current Sprint. What do you do? A. Take responsibility for the task, and tell the Product owner to postpone the deadline for the current Sprint. B. Inform the Scrum Master of the situation, and let him or her discuss the situation with the GM. C. Talk to the Product Owner, and tell him or her to re-assign your tasks to someone else. D. Talk with the other Scrum Team members and re-assign your tasks to someone else	1	1	1	12	12.3.2

4	Identify the cost estimation technique that is independent of programming language and can be measured before software design and code done. A. LOC B. FP C. Process-based D. Tool based	1	2	1	2	2.2.3
5	If you want to develop a solution to support during floods which model, would you adopt preferably? A. V-Model B. Waterfall C. Spiral D. RAD	1	1	1	2	2.2.3
6	Select the four framework activities that are found in the Extreme Programming (XP). A. Analysis, design, coding, testing B. Planning, analysis, design, coding C. Planning, design, coding, testing D. Planning, analysis, coding, testing	1	2	2	2	2.2.4
7	The elements that are implied by scenarios are A. Functional B. Class based C. Behavioral D. Flow oriented	1	2	2	2	2.2.4
8	COCOMO stands for _____. A. Consumed cost model B. Constructive cost model C. Common control model D. Composition cost model	1	2	2	2	2.2.4
9	Estimation of software development effort for organic software is COCOMO is A. $E = 2.4(KLOC)^{1.05}$ PM B. $E = 3.4(KLOC)^{1.06}$ PM C. $E = 2.0(KLOC)^{1.05}$ PM D. $E = 2.4(KLOC)^{1.07}$ PM	1	2	2	2	2.2.4
10	Which one of the following is not a step of requirement engineering? A. Elicitation B. Design C. Analysis D. Documentation	1	2	2	2	2.2.4

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu

College of Engineering and Technology
School of Computing
Department of Data Science and Business Systems

Academic Year: 2023-24 (Even)

Set: B
Section:

Test	: CLA-T1	Date	: 19-02-2024
Course Code & Title	: 21CSC303J & Software Engineering and Project Management	Duration	: 1hr 30 Minutes (2 Periods)
Year & Sem	: III Year / IV Sem	Max Marks	: 50

Part – B (4 x 5 = 20 Marks)
Instructions: Answer ANY FOUR questions

Q. No	Question	Marks	BL	CO	PO	PI Code
11	List out the merits and demerits of Bohem's model	5	2	1	2	2.3.2
12	With a neat sketch explain the types of process flow	5	2	1	1	12.3.2
13	Mention any five agile development process principles?	5	3	1	1	12.3.2
14	With the diagram brief about requirement elicitation and analysis process	5	1	2	2	2.2.4
15	How can ethnographic method be effectively utilized in software requirement elicitation processes to gain deeper insights into user behaviors, needs, and contexts, and brief the process in ethnographic with a block diagram	5	3	2	2	2.2.4

Part – C (2 x 10 = 20 Marks)
Instructions: Answer ALL questions

Q. No	Question	Marks	BL	CO	PO	PI Code
16	Given Covid-19 at its early 2019, you have been asked to support as a IT person to develop an APP as quick as possible before situation gets worsen and disastrous, which can serve as nodal point to match need and supply for beds, ventilators, drugs and other crucial essentials between patients, hospitals and pharmacy. Propose which traditional model would you adopt, why did you adopt this, what are the advantages, do identify the stakeholders. Also list if there are any challenges exist in this model as well	10	3	1	2	2.3.1
17	(OR) Assume you are a software developer of the company. A client has approached for a better solution for the problem they have faced by their side. The client has stated that the risk and uncertainty will lead to loss if not properly planned. Justify with pictorial representation, which model will be chosen for software development. List its merits and demerits in detail.					

18	What are the fundamental activities involved in the software requirements engineering process, and how do these activities contribute to the successful development and delivery of high-quality software products across various domains and industries. Sketch the spiral view of the software engineering process. (OR)	10	3	2	2	2.4.1
19	(a) The software equation is a dynamic multivariable model that assumes a specific distribution of effort over the life of a software development project. To simplify the estimation process and use a more common form for their estimation model. A set of equations derived from the software equation. Use the software equation to estimate the lawn mowing robot software. Assume that equation is applicable and that $P = 8000$, $LOC = 33,580$, $B = 0.28$ (5 marks) (b) For a given project was estimated with a size of 400 KLOC. Calculate the Effort, Scheduled time for development. Also, calculate the Average staff size and Productivity of the software for Organic, Semidetached, Embedded project type (5 marks)					

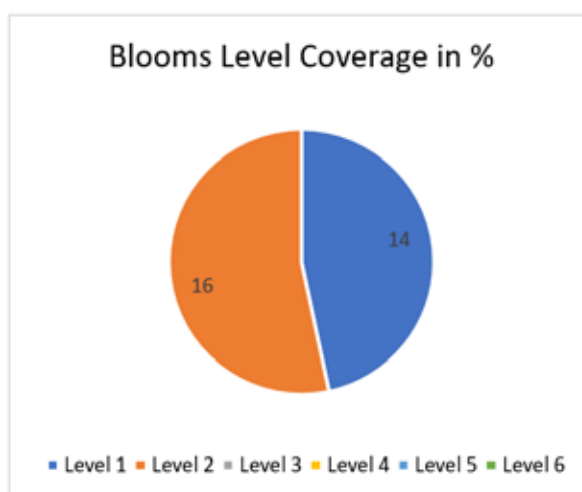
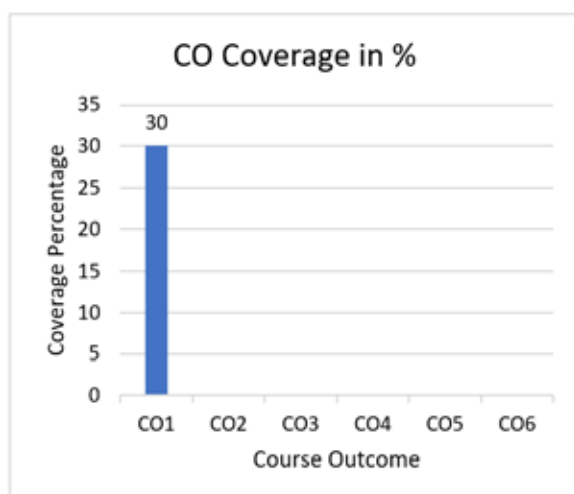
Equations

NAME	Equation	Unit
Efforts	$=a(KLOC)^b$	Persons month
Development Time	$c(effort)^d$	Months
Effort Staff Size	$effort/dev\ time$	persons
Productivity	$KLOC/Effort$	KLOC/PM

VALUES

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Course Outcome (CO) and Bloom's level (BL) Coverage



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu

College of Engineering and Technology
School of Computing
Department of Data Science and Business Systems

Academic Year: 2023-24 (Even)

 Set: A
 Section:

Test	: CLA-T2	Date	: 03-04-2024
Course Code & Title	: 21CSC303J & Software Engineering and Project Management	Duration	: 1hr 30 Minutes (2 Periods)
Year & Sem	: III Year / VI Sem	Max Marks	: 50

Course Articulation Matrix:

Sl. No.	Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	CO1	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
2	CO2	-	3	-	-	-	-	-	-	2	-	2	-	3	-	-
3	CO3	-	-	-	2	-	-	-	-	2	-	2	-	3	-	-
4	CO4	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
5	CO5	-	-	-	-	-	-	-	-	2	-	3	-	3	-	-

Part – A (10 x 1 = 10 Marks)
Instructions: Answer ALL questions.

Q. No	Question	Marks	BL	CO	PO	PI Code
1	<p>----- is to define a set of interface objects and actions that enable a user to perform all defined tasks.</p> <p>A. Interface Analysis</p> <p>B. Interface Design</p> <p>C. Interface Construction</p> <p>D. Interface Validation</p> <p>ANSWER: B</p>	1	1	2	1	1.1.2
2	<p>Which design model elements are used to depict a model of information represented from the user's view?</p> <p>A.component level elements</p> <p>B.user interface elements</p> <p>C.architectural elements</p> <p>D.data design elements</p> <p>ANSWER: D data design elements</p>	1	1	2	3	2.6.2

3	<p>Identify the best model which can be used to pictorially represent archetypes of an architectural design</p> <p>A. CSS B. HTML C. UML D. ADL</p> <p>ANSWER: C</p>	1	1	2	2	2.1.1
4	<p>----- is an indication of the relative interdependence among modules.</p> <p>1. Cohesion 2. Coupling 3. Elaboration 4. Refactoring</p> <p>ANSWER: B</p>	1	1	2	2	1.3.1
5	<p>A good software design will have</p> <p>A. High Coupling B. Low Coupling C. Average Coupling D. Very high Coupling</p> <p>ANSWER: B</p>	1	1	2	1	1.3.1
6	<p>Converting the specifications into software code is totally dependent on the _____ team.</p> <p>A. Constructing B. Sales C. Debugging D. Testing</p> <p>ANSWER: A</p>	1	1	3	3	3.5.1

7	<p>_____ is a technique where one developer writes the code and other developer sits behind him then guides him through the requirements.</p> <p>A. Automatic code generation B. Software code reuse C. Pair programming D. Test driven development</p> <p>ANSWER: C</p>	1	1	3	1	1.2.2 1.3.1
8	<p>What are the activities must be ensured during defect tracking phase?</p> <p>a.Logging defect b.Execute test c.Fixing defect d.Log and fix defect</p> <p>a) Answer:d</p>	1	2	4	11	11.2.1
9	<p>What type of test must be done for applications that have many versions?</p> <p>a.Stress testing b.Recovery testing c.Regression testing d.Mutation testing</p> <p>Answer: c</p>	1	2	4	2	2.4.2
10	<p>_____ method leads to a selection of test cases that exercise boundary values.</p> <p>a.Boundary Value Analysis b.Cause-effect graphing c.Equivalence Partitioning d.Basis Path Testing</p> <p>Answer: a</p>	1	2	4	1	1.3.1

Register Number																	
-----------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu

College of Engineering and Technology

School of Computing

Department of Data Science and Business Systems

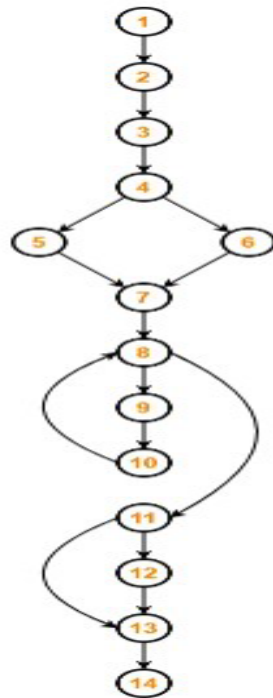
Academic Year: 2023-24 (Even)

Set: A
Section:

Test	: CLA-T1	Date	: 19-02-2024
Course Code & Title	: 21CSC303J & Software Engineering and Project Management	Duration	: 1hr 30 Minutes (2 Periods)
Year & Sem	: III Year / IV Sem	Max Marks	: 50

Part – B (4 x 5 = 20 Marks)						
Instructions: Answer ANY FOUR questions						
Q. No	Question	Marks	BL	CO	PO	PI Code
11	<p>Categorize the various architecture partitioning techniques. List out the advantages of partitioning the architecture</p> <ul style="list-style-type: none"> • Results in software that are easier to test • Leads to software that are easier to maintain • Results in propagation of fewer side effects • Results in software that are easier to extend <p>Horizontal Partitioning</p> <ul style="list-style-type: none"> • Define separate branches of the module hierarchy for each major function • Use control modules to coordinate communication between functions <p>Vertical Partitioning – Factoring</p> <ul style="list-style-type: none"> • Design so that decision making and work are stratified • Decision making modules should reside at the top of the architecture 	5	2	2	3	3.6.1
12	Discuss UI Interface design evaluation cycle with a neat sketch	5	3	3	3	3.5.1

	<div><p>Design Evaluation Cycle</p><pre>graph TD; A[preliminary design] --> B[build prototype #1 interface]; B --> C[user evaluate's interface]; C --> D[evaluation is studied by designer]; D --> E[design modifications are made]; E --> F[build prototype # n interface]; F --> A; F --> G[Interface design is complete];</pre></div>							
13	<div><p>Differentiate verification and Validation</p><table><tr><td><p>1. Verification is a static practice of verifying documents, design, code and program.</p><p>2. It does not involve executing the code.</p><p>3. It is human based checking of documents and files.</p><p>4. Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.</p></td><td><p>1. Validation is a dynamic mechanism of validating and testing the actual product.</p><p>2. It always involves executing the code.</p><p>3. It is computer based execution of program.</p><p>4. Validation uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.</p></td></tr></table></div>	<p>1. Verification is a static practice of verifying documents, design, code and program.</p> <p>2. It does not involve executing the code.</p> <p>3. It is human based checking of documents and files.</p> <p>4. Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.</p>	<p>1. Validation is a dynamic mechanism of validating and testing the actual product.</p> <p>2. It always involves executing the code.</p> <p>3. It is computer based execution of program.</p> <p>4. Validation uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.</p>	5	3	1	1	12.3.2
<p>1. Verification is a static practice of verifying documents, design, code and program.</p> <p>2. It does not involve executing the code.</p> <p>3. It is human based checking of documents and files.</p> <p>4. Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.</p>	<p>1. Validation is a dynamic mechanism of validating and testing the actual product.</p> <p>2. It always involves executing the code.</p> <p>3. It is computer based execution of program.</p> <p>4. Validation uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.</p>							
14	<div><p>Brief about top-down integration testing with its categories. Draw the diagrams for the categories Answer: Depth first and breadth-first</p><pre>graph TD; M1[M1] --- M2[M2]; M1 --- M3[M3]; M1 --- M4[M4]; M2 --- M5[M5]; M2 --- M6[M6]; M5 --- M8[M8]; M3 --- M7[M7]; style M4 stroke-dasharray: 5 5; style M7 stroke-dasharray: 5 5;</pre></div>	5	1	2	2	2.2.4		
15	<div><p>Calculate the cyclomatic complexity for the given flow graph by 3 methods</p></div>	5	3	2	2	2.2.4		



Method-01:

Cyclomatic Complexity

= Total number of closed regions in the control flow graph + 1

$$= 3 + 1$$

$$= 4$$

Method-02:

Cyclomatic Complexity

$$= E - N + 2$$

$$= 16 - 14 + 2$$

$$= 4$$

Method-03:

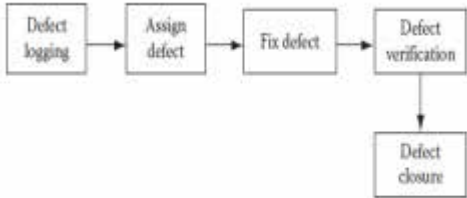
Cyclomatic Complexity


$$= P + 1$$

$$= 3 + 1$$

$$= 4$$

Part – C (2 x 10 = 20 Marks)
Instructions: Answer ALL questions

Q. No	Question	Marks	BL	CO	PO	PI Code
16	<p>IrisGold is a gold mining company that operates on three continents, with more than 21,000 employees. The company's mines are mostly located in remote places like the Amazonas in Brazil, the Andes mountain range, the Ural mountains in Russia, and eastern South Africa. The company is selecting an Enterprise Resource Planning (ERP) system package. Suggest Which architecture is suitable to build the same with a neat diagram and a suitable description</p> <p>Answer:</p> <p>Data Centered Architecture/Client Server Architecture</p> <p>Diagram: 4 Marks</p> <p>Explanation: 54Marks</p> <p>Justification with advantages: 2marks</p> <p>(OR)</p>	10	3	2	3	3.8.2
17	<p>How the performance of the testing team is measured? Relate different ways of tracking the testing efficiency. Answer: Defect tracking (3 Marks) Explanation about defect tracking (3 Marks) In-house project testing Outsourced testing (2 Marks) Deploying defect tracking applications of measuring the defects (2 Marks) Diagram (5 Marks)</p>  <pre> graph LR A[Defect logging] --> B[Assign defect] B --> C[Fix defect] C --> D[Defect verification] D --> E[Defect closure] </pre>					

18	<p>How significant is quality control for software construction ? Explain them in detail.</p> <ul style="list-style-type: none"> It is estimated that almost 70% of software defects arise from faulty software code. To compound this problem, software construction is the most labor intensive phase in software development. Any construction rework means wasting a lot of effort already put in. Moreover, it is also a fact that it is cheaper to fix any defects found during construction at the phase level itself. If those defects are allowed to go in software testing (which is the next phase), then fixing those defects will become costlier. That is why review of the software code and fixing defects is very important. There are some techniques available like desk-checks, walkthroughs, code reviews, inspections, etc. that ensure quality of the written code (Figure below-Source code review methods and their operation sequence).  <pre> graph LR A[Source code] --> B[Deskchecks] B --> C[Walkthroughs] C --> D[Code reviews] D --> E[Final inspection] </pre> <ul style="list-style-type: none"> These different kinds of reviews are done at different stages in software code writing. They also serve different purposes. While inspections provide the final go/no go decision for approval of a piece of code, other methods are less formal and are meant for removing defects instead of deciding whether a piece of code is good enough or not. <p>ATLEAST 2 points per technique</p> <p>Reviews – Desk-checks (Peer Reviews)</p> <ul style="list-style-type: none"> Desk-checks are employed when a complete review of the source code is not important. Here, the developer sends his piece of code to the designated team members. These team members review the code and send feedback and comments to the developer as suggestions for improvement in the code. The developer reads those feedbacks and may decide to incorporate or to discard those suggestions. So this form of review is totally voluntary. Still, it is a powerful tool to eliminate defects or improve software code. <p>Reviews – Walkthroughs</p>	1 0	3	2	2	2.4.1
----	--	--------	---	---	---	-------

- Walkthroughs are formal code reviews initiated by the developer. The developer sends an invitation for walkthrough to team members.
- At the meeting, the developer presents his method of coding and walks through his piece of code.
- The team members then make suggestions for improvement, if any.
- The developer then can decide to incorporate those suggestions or discard them.

Reviews – Code Reviews

- Code reviews are one of the most formal methods of reviews. The project manager calls for a meeting for code review of a developer.
- At the meeting, team members review the code and point out any code errors, defects, or improper code logic for likely defects. An error log is also generated and is reviewed by the entire team.

Reviews – Inspections

- Code inspections are final reviews of software code in which it is decided whether to pass a piece of code for inclusion into the main software build.

(OR)

Why black box testing is necessary? Explain Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.

19

It checks for the input values near the boundary that have a higher chance of error. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition. value analysis with a neat sketch

Example: Consider a system that accepts ages from 18 to 56.

Boundary Value Analysis(Age accepts 18 to 56)

Invalid (min-1)	Valid (min, min + 1, nominal, max – 1, max)	Invalid (max + 1)
17	18, 19, 37, 55, 56	57

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu

College of Engineering and Technology

School of Computing

Department of Data Science and Business Systems

Academic Year: 2023-24 (Even)

Set: B Section:

Test	: CLA-T2	Date	: 03-04-2024
Course Code & Title	: 21CSC303J & Software Engineering and Project Management	Duration	: 1hr 30 Minutes (2 Periods)
Year & Sem	: III Year / VI Sem	Max Marks	: 50

Course Articulation Matrix:

Sl. No.	Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	CO1	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
2	CO2	-	3	-	-	-	-	-	-	2	-	2	-	3	-	-
3	CO3	-	-	-	2	-	-	-	-	2	-	2	-	3	-	-
4	CO4	-	-	-	-	-	-	-	-	2	-	2	-	3	-	-
5	CO5	-	-	-	-	-	-	-	-	2	-	3	-	3	-	-

Part – A (10 x 1 = 10 Marks)

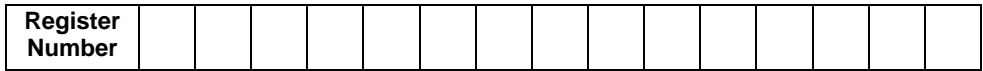
Instructions: Answer ALL questions.

Q. No	Question	Marks	BL	CO	PO	PI Code
1	<p>_____ is basically the time duration necessary to complete the project tasks</p> <p>a) Schedule</p> <p>b) Effort</p> <p>c) Outcome</p> <p>d) Attribute</p> <p>Ans: A</p>	1	1	2	1	1.1.2
2	<p>In order to estimate cost, resource and schedule for a software engineering effort, which one of these is not necessary?</p> <p>a) Experience</p> <p>b) Access to proper historical details</p> <p>c) Courage to commit quantitative predictions</p>	1	1	2	3	2.6.2

	<p>d) Funds</p> <p>Ans: D</p>					
3	<p>.Test strategies should not include things like _____</p> <p>a.test prioritization,</p> <p>b.automation strategy,</p> <p>c.risk analysis</p> <p>d.Maintenance</p> <p>Ans: D</p>	1	1	2	2	2.1.1
4	<p>For fixing defects in the source code reviews are conducted whereas technique is used for the developers when a complete review of the source code is not important.</p> <p>a. Inspections b. Walkthroughs c. desk checks d. final inspection</p> <p>Ans: C</p>	1	1	2	2	1.3.1
5	<p>Which of the following objective comes under formal technical reviews?</p> <p>a. Uncover errors in Software work products</p> <p>b. Allow senior personals to correct errors</p> <p>c. Assess programmer productivity</p> <p>d. Determining who introduced an error into a program</p> <p>Ans:A</p>	1	1	2	1	1.3.1

6	<p>Which of the following is not one of the elements of a design pattern?</p> <p>a) context</p> <p>b) Environment</p> <p>c) problem</p> <p>d) solution</p> <p>Ans: B</p>	1	1	3	3	3.5.1
7	<p>Which of the following is not part of the Test document?</p> <p>a. Test Case</p> <p>b. Requirements Traceability Matrix [RTM]</p> <p>c. Test strategy</p> <p>d. Project Initiation Note [PIN]</p> <p>Ans: D</p>	1	1	3	1	1.2.2
8	<p>Which of the following testing is also known as white-box testing?</p> <p>a. Structural testing</p> <p>b. Error guessing technique</p> <p>c. Design based testing</p> <p>d.Code testing</p> <p>Ans: A</p>	1	2	4	11	11.2.1
9	<p>Validation testing is also known as -----</p> <p>a.Dynamic Testing</p> <p>b. Verification Testing</p> <p>c. System Testing</p> <p>d. Static Testing</p> <p>Ans: A</p>	1	2	4	2	2.4.2
10	<p>Exhaustive testing is</p>	1	2	4	1	1.3.1

	a. always possible					
	b. practically possible					
	c. impractical but possible					
	d. impractical and impossible					
	Ans: C					



Section:

Part – B (4 x 5 = 20 Marks)						
Instructions: Answer ANY FOUR questions						
Q. No	Question	Marks	BL	CO	PO	PI Code
11	<p>Discuss the purpose of black box testing in software testing</p> <p>Black box testing is a software testing technique where the internal structure, design, or implementation of the system under test is not known to the tester. Instead, the tester interacts with the software's user interface or external inputs and analyzes the outputs or behavior of the system. The primary purpose of black box testing is to assess the software's functionality, its compliance with requirements, and its behavior under different conditions without considering its internal workings.</p>	5	2	2	3	3.6.1
12	<p>Draw the sequence diagram of bank atm</p>	5	3	3	3	3.5.1
13	<p>Differentiate alpha testing and beta testing</p> <ul style="list-style-type: none"> n When custom software is built for one customer, a series of acceptance tests are conducted to enable the customer to validate all requirements. n Conducted by the end-user rather than software engineers, an acceptance test can range from an informal "test drive" to a planned and systematically executed series of tests. n Most software product builders use a process called alpha and beta testing to uncover errors that only the end-user seems able to find. n The alpha test is conducted at the developer's site by a customer. n The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problems. n Alpha tests are conducted in a controlled environment. n The beta test is conducted at one or more customer sites by the end-user of the software. n beta test is a "live" application of the software in an environment that cannot be controlled by the developer. n The customer records all problems (real or imagined) that are encountered during beta testing and reports these to the developer at regular intervals. n As a result of problems reported during beta tests, software engineers make modifications and then prepare for release of the software product to the entire customer base. 	5	3	1	1	12.3.2

14	Brief about bottom up integration testing with its categories.	5	1	2	2	2.2.4
15	Calculate cyclomatic complexity for the given code- 1. IF A = 354 2. THEN IF B > C 3. THEN A = B 4. ELSE A = C 5. END IF 6. END IF 7. PRINT A	5	3	2	2	2.2.4

Part – C (2 x 10 = 20 Marks) Instructions: Answer ALL questions						
Q. No	Question	Marks	BL	CO	PO	PI Code
16	<p>Explain design types prototyping and entity relationship models</p> <p>(OR)</p> <p>Describe the process of creating test cases in black box testing. Provide examples of equivalence classes for a hypothetical login system and how you would design test cases based on them.</p>	10	3	2	3	3.8.2
17						
18	Discuss on the various architecture styles used in the software design (OR)	10	3	2	2	2.4.1
19.	Explain the concept of black box testing comprehensively. Provide detailed insights into its principles, techniques, advantages, and limitations. Additionally, elucidate with examples how black box testing differs from white box testing, and under what circumstances each approach is preferable. Support your answer with relevant illustrations or real-world scenarios.					

18. Architectural Design: Focuses on the high-level structure of the software system, defining components, their interactions, and the overall organization.

Object-Oriented Design (OOD): Based on the principles of encapsulation, inheritance, and polymorphism, OOD models software as a collection of interacting objects.

Functional Design: Emphasizes the decomposition of a system into smaller, self-contained functions or modules, often used in functional programming paradigms.

Database Design: Concerned with the structure of the database that will store and manage the application's data, including tables, relationships, and constraints.

User Interface (UI) Design: Focuses on creating an intuitive and visually appealing interface for users to interact with the software, considering usability and user experience.

Interaction Design: Deals with defining how users interact with the software, including navigation flows, input methods, and feedback mechanisms.

Component-Based Design: Involves building software by integrating pre-built, reusable components or modules, facilitating modular development and maintenance.

Service-Oriented Architecture (SOA): Designs software as a collection of loosely coupled services that communicate with each other over a network, promoting flexibility and scalability.

Model-Driven Design (MDD): Uses models to represent different aspects of the software system, allowing for automated code generation and analysis.

Data-Driven Design: Focuses on designing software based on data analysis and insights, often used in decision support systems and analytics applications.

Domain-Driven Design (DDD): Aligns software design with the domain it operates in, emphasizing a deep understanding of the problem domain and the use of ubiquitous language.

Incremental and Iterative Design: Involves designing and implementing software in small, manageable increments, refining and improving the design through multiple iterations.

Responsive Design: Ensures that software adapts and functions well across different devices and screen sizes, commonly applied in web and mobile applications.

Security Design: Integrates security measures into the software design process to identify and mitigate potential vulnerabilities and threats.

Agile Design: Emphasizes collaboration, adaptability, and customer feedback throughout the design process, often used in iterative and incremental development methodologies like Scrum and Kanban.

In software design, the Entity-Relationship (ER) model serves as a fundamental tool for conceptualizing and designing the data structure of an application. Here's how the ER model is applied in software design:

Conceptual Modeling: The ER model helps software designers understand the data requirements and relationships within the application domain. It enables them to identify key entities, attributes, and relationships between entities based on the business logic and user requirements.

Data Modeling: Once the conceptual model is established, software designers use the ER model to create a detailed data model that defines the entities, attributes, and relationships in a structured manner. This includes specifying primary keys, foreign keys, and cardinality constraints.

Database Design: The ER model serves as a blueprint for designing the database schema in a relational database management system (RDBMS). Designers translate the ER diagram into tables, columns, and relationships, ensuring that the database reflects the conceptual model accurately.

Application Development: During the software development process, developers reference the ER model to inform the design and implementation of data access and manipulation functionality. The model guides the development of database queries, data validation rules, and business logic related to data processing.

Integration with Object-Oriented Design: In object-oriented software design, the ER model can be integrated with class diagrams to bridge the gap between the conceptual model and the software

architecture. Object-relational mapping (ORM) frameworks facilitate the mapping of ER model entities to object-oriented classes and vice versa.

Maintenance and Evolution: The ER model serves as documentation for the application's data structure, making it easier to understand and maintain the software over time. When changes or updates are required, designers can refer to the ER model to assess the impact on the data model and ensure consistency.

Overall, the Entity-Relationship model plays a crucial role in software design by providing a structured approach to defining data requirements, facilitating database design, guiding application development, and supporting long-term maintenance and evolution of software systems.

19. Describe the process of creating test cases in black box testing. Provide examples of equivalence classes for a hypothetical login system and how you would design test cases based on them.

In black box testing, the internal workings of the software under test are not known to the tester. Instead, test cases are derived based on the system's requirements and functionality. Here's the general process for creating test cases in black box testing, followed by examples of equivalence classes for a login system and how to design test cases based on them:

Process of Creating Test Cases in Black Box Testing:

Understand Requirements: Begin by thoroughly understanding the requirements and specifications of the software being tested.

Identify Inputs and Outputs: Determine the inputs that the software accepts and the outputs it produces based on user interactions.

Partition Inputs into Equivalence Classes: Group inputs into equivalence classes, where each class represents a set of inputs that are treated similarly by the system.

Derive Test Cases: Generate test cases based on the equivalence classes, aiming to cover both valid and invalid scenarios, boundary conditions, and error-handling cases.

Design Test Data and Scenarios: Define the test data and scenarios needed to execute the test cases, including input values, expected outcomes, and any preconditions or dependencies.

Execute Tests: Execute the test cases against the software, observing the actual outcomes and comparing them with the expected results.

Analyze Results: Analyze the test results to identify defects, inconsistencies, or deviations from expected behavior.

Iterate and Refine: Iterate on the test cases and refine them based on feedback, additional requirements, or changes to the software.

Example: Equivalence Classes for a Login System:

Let's consider a hypothetical login system with the following input fields: Username and Password.

Equivalence Classes:

Valid Equivalence Classes:

Valid username and password combination.

Invalid Equivalence Classes:

Invalid username and valid password.

Valid username and invalid password.

Invalid username and invalid password.

Empty username and password fields.

Username and password exceeding maximum length.

Designing Test Cases:

Test Case 1: Valid Login

Input: Valid username and password combination

Expected Outcome: Successful login

Test Case 2: Invalid Username

Input: Invalid username and valid password

Expected Outcome: Login failure with appropriate error message

Test Case 3: Invalid Password

Input: Valid username and invalid password

Expected Outcome: Login failure with appropriate error message

Test Case 4: Invalid Username and Password

Input: Invalid username and invalid password

Expected Outcome: Login failure with appropriate error message

Test Case 5: Empty Fields

Input: Empty username and password fields

Expected Outcome: Login failure with appropriate error message

Test Case 6: Maximum Length Exceeded

Input: Username and password exceeding maximum length

Expected Outcome: Login failure with appropriate error message

These test cases cover various scenarios and inputs, ensuring comprehensive coverage of the login system's functionality and error-handling capabilities.