SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2024-2025 (EVEN)

SET - D

ANSWER KEY

Test: FJ1 **Date: 19.02.2025**
Course Code & Title:21CSC303J - Software Engineering and Project Management **Duration: 100 minutes**
Year & Sem: III & VI **Max. Marks:** 50

Course Articulation Matrix:

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	--	--	--	--	--	--	--	--	2	--	2	--
CO2	--	3	--	--	--	--	--	--	2	--	2	--

Note: CO1 : Identify the process of project life cycle model and process.
CO2 : Analyze and translate end-user requirements into system and software requirements.

Part – A (10 × 1 = 10 Marks)
Instructions: Answer all
The duration for answering part A is **15 minutes** (this sheet will be collected after 15 minutes).

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Which of the following is NOT a part of the generic process framework in software engineering? A) Communication B) Planning **C) Testing** D) Deployment	1	1	1	2	1.6.1*
2	You are working on a project where the requirements are well-defined and unlikely to change throughout the project's lifecycle. What is the most appropriate process model to use for this project? **A) Waterfall Model** B) Agile Model C) Spiral Model D) Incremental Model	1	1	1	2	2.6.3
3	You are in the initial phase of a software development project following the V-Model, and you're working on gathering the requirements. What is the key testing activity that should be prepared during this phase? A) Preparing for system testing. B) Writing unit test cases. **C) Creating acceptance criteria for the system.** D) Preparing for integration testing.	1	1	1	2	1.6.1
4	What is the time-boxed period in which Scrum teams deliver a potentially shippable product increment? A) Milestone B) Phase **C) Sprint** D) Release	1	1	1	9	1.6.1
5	Mid-sprint, the product owner asks for additional features that are not part of the original sprint goal. How should the Scrum Master handle this?	1	1	1	9	1.6.1

	A) Immediately agree to add the features without adjusting the sprint plan. B) Politely reject the request and remind the product owner of the sprint goal. C) Work with the product owner to evaluate the new request and adjust the sprint goal if needed. **D) Defer the new features to the next sprint and focus only on current work**.					
6	During the review of a Software Requirement Specification (SRS), you notice that some requirements are vague and unclear, making it difficult for the development team to implement them. What should you do? A) Skip the vague requirements and continue with the clear ones. **B) Report the vague requirements to the stakeholders and ask for clarification.** C) Assume the intended meaning and proceed without clarification. D) Remove the ambiguous requirements entirely from the SRS.	1	1	2	2	1.6.1
7	Which section of the SRS defines performance, security, and usability constraints? A) Functional Requirements **B) Non-functional Requirements** C) Technical Design D) Test Cases	1	1	2	2	2.6.3
8	The effort estimation formula in the Basic COCOMO model is: **A) Effort = a × (KLOC)^b** B) Effort = (KLOC) + b × a C) Effort = a + b × KLOC D) Effort = (KLOC) × a × b	1	1	2	2	1.6.1
9	You are collecting requirements from multiple stakeholders with varying interests and priorities. What approach should you take to ensure all needs are captured accurately? A) Gather requirements from only the most influential stakeholders to avoid confusion. B) Hold individual meetings with each stakeholder and then combine the requirements in a single document. **C) Organize a workshop or group discussion where all stakeholders can share and prioritize their requirements.** D) Assume the most vocal stakeholders represent the needs of all other stakeholders	1	1	2	2	2.6.3
10	In requirement elicitation, what is meant by "requirements conflict"? A) A situation where two developers disagree on coding standards B) A mismatch between software features and hardware requirements **C) When different stakeholders have contradictory requirements** D) When software testing fails to detect bugs	1	1	2	2	2.6.3

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

SET - D

Academic Year: 2024-2025 (EVEN)

Test: FJ1 | **Date: 19.02.2025**
Course Code & Title:21CSC303J - Software Engineering and Project Management | **Duration: 100 minutes**
Year & Sem: III & VI | **Max. Marks:** 50

Course Articulation Matrix:

Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO 12
CO1	--	--	--	--	--	--	--	--	2	--	2	--
CO2	--	3	--	--	--	--	--	--	2	--	2	--

Note: CO1 : Identify the process of project life cycle model and process.
CO2 : Analyze and translate end-user requirements into system and software requirements.

Part – B (4 × 5 = 20 Marks)
Instructions: Answer any Four questions.

11	What are the risks associated with unclear requirements in the Waterfall Model, and how can they be mitigated? **Risks Associated with Unclear Requirements in the Waterfall Model:** **Scope Creep: Unclear or ambiguous requirements can lead to scope creep as stakeholders introduce new requirements or modify existing ones during the project, causing delays and cost overruns.** **Miscommunication: If requirements are not clearly understood or documented, there may be a mismatch between the stakeholders' expectations and the development team's implementation, resulting in system features that do not meet user needs.** **Inflexibility to Changes: The Waterfall model is sequential, making it difficult to accommodate changes once development begins. Unclear requirements that are not properly defined early on will lead to costly and complex changes later in the process.** **Customer Dissatisfaction: When the final product does not align with stakeholder expectations due to unclear requirements, it can lead to dissatisfaction and require extensive rework, affecting timelines and budgets.**	5	1	1	3	1.6.1

12	A Scrum team is halfway through a Sprint when the Product Owner receives feedback from stakeholders requesting significant changes to a key feature. How should the Scrum team handle these change requests? **Handling Change Requests Mid-Sprint in Scrum:** **1. Defer Changes to the Next Sprint: The Scrum team should focus on the current Sprint goal and not introduce significant changes mid-Sprint. The changes can be added to the Product Backlog and prioritized for the next Sprint Planning session.** **2. Discuss with the Product Owner: The Product Owner should evaluate the request's priority and impact with the stakeholders. If the change is urgent and highly valuable, the Product Owner may choose to adjust the Product Backlog and consider it for future work, not interrupting the current Sprint.** **3. Transparency and Collaboration: The Scrum Master facilitates a discussion between the team and the Product Owner, ensuring transparency and collaboration. It is essential that all stakeholders understand that changes may affect the project's timeline if introduced mid-Sprint.** 4. **Maintain Sprint Focus: The Scrum team should remain focused on delivering the committed work for the current Sprint, as altering the scope could impact their ability to meet the Sprint goal. Managing the scope is essential for maintaining Sprint integrit**y.	5	1	1	3	1.6.1
13	An Agile team struggles because the customer is rarely available for feedback during Sprint Reviews or Backlog Refinement sessions. Why is this a problem in Agile, and how can the team address it? **Problem:** **1. Lack of Customer Feedback: Agile relies on customer collaboration to ensure that the team is building features that meet the customer's needs. Without regular feedback, the team risks building features that may not align with the customer's evolving priorities.** **2. Missed Opportunities for Refinement: Backlog Refinement sessions depend on customer input to clarify and prioritize requirements. If the customer is unavailable, the team may make assumptions, leading to misunderstandings or incorrect priorities.** **How to Address It:** **1. Use a Proxy or Representative: If the customer is unavailable, the team can work with a proxy, such as a business analyst or product manager, who can provide relevant feedback and ensure**	5	2	1	3	2.6.3

	the team stays on track. 2. **Improve Communication Channels: Set up asynchronous feedback channels like email or collaboration tools to gather input from the customer when they are unavailable for live sessions.** 3. **Schedule Regular Check-ins: Establish a regular time for customer check-ins to ensure their feedback is captured and prioritized, even if they can't attend every Sprint Review or Refinement session.** 4. **Product Owner Involvement: The Product Owner should act as the main point of contact between the customer and the team, ensuring that priorities and feedback are captured and conveyed accurately.**					
14	A company needs to develop a digital signal processing software for one of its newest investors. The software is expected to have 40000 lines of code. The company needs to determine the effort in person months needed to develop this software using COCOMO model. The multiplication factor is given as 1.20. Calculate the estimated effort in person months. **Effort Applied (E) = ab(KLOC)bb [person-months]** **= 2.8 x(40)1.20** **= 2.8 x 83.65** **= 234.25**	5	2	2	11	1.6.1
15	**During the development phase of a project, the client requests changes to the requirements listed in the SRS. The changes include additional features and alterations to previously agreed-upon functionalities. How should the SRS be updated, and how would you ensure these changes are managed effectively?** **Updating the SRS:** **Document Changes Clearly:** **All new features and modifications should be clearly documented in the SRS, including detailed descriptions and specifications. This ensures the scope of changes is understood by all stakeholders.** **Version Control:** **The updated SRS should be version-controlled to track changes over time. Both the original and modified versions should be accessible to ensure transparency and traceability.** **Impact Analysis:** **An impact analysis should be conducted to assess how these changes will affect the overall system, timeline,**	5	3	2	9	1.6.1

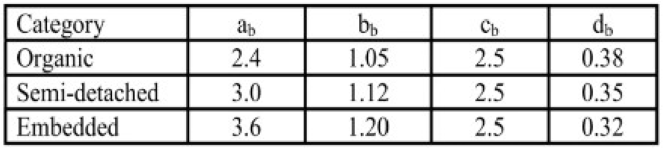
	budget, and other requirements. The affected areas of the SRS should be updated accordingly. **Review and Approval:** **The updated SRS should undergo a review and approval process with all relevant stakeholders (client, development team, and project manager) to ensure alignment on the new requirements and their impact.**					
	Part – C (2 × 10 = 20 Marks) **Instructions: Answer all**					
16	**Your team conducts Sprint Reviews regularly, but stakeholders rarely attend, and when they do, they seem disengaged. This leads to minimal feedback, affecting the product's alignment with business goals. How would you improve the effectiveness of Sprint Reviews?** **To improve the effectiveness of Sprint Reviews and ensure better engagement from stakeholders, the following actions can be taken:** **Ensure Relevance of the Sprint Review:** **Make sure that the Sprint Review focuses on features and deliverables that align with business goals and stakeholder interests. Tailor the presentation to highlight the most valuable aspects of the work that have been completed and how they contribute to the overall business objectives.** **Involve Stakeholders Early and Often:** **Encourage stakeholders to participate in Backlog Refinement sessions and other Agile ceremonies. By involving them earlier in the process, they will have a better understanding of the product's development, making them more likely to engage during Sprint Reviews.** **Set Clear Expectations and Outcomes:** **Communicate the purpose of the Sprint Review upfront and make it clear that the goal is to get feedback and alignment from stakeholders. Define what specific input is needed from them (e.g., prioritization, feature validation, etc.) to guide discussions.** **Make the Review Interactive:** **Instead of just presenting the work, encourage discussions and active participation. Use tools like demonstrations, prototypes, or mockups to showcase features. Create an environment where stakeholders feel comfortable asking questions and giving feedback.** **Incentivize Attendance:** **If stakeholders continue to be disengaged, find ways to motivate their attendance. This could be by making the**	10	3	1	9	1.6.1

	meeting time more convenient, emphasizing the value of their input, or even offering incentives like highlighting how their involvement will directly impact the product's success. **Follow-Up on Feedback:** **Ensure that the feedback from the Sprint Review is captured, analyzed, and addressed in future Sprints. Demonstrating that their input leads to concrete changes increases stakeholders' engagement in future reviews.** **Provide a Clear Agenda and Context:** **Share the agenda ahead of time, so stakeholders know exactly what to expect and can prepare any relevant feedback. Additionally, give context during the review to explain how the completed work fits into the overall project goals and roadmap.**					
	(OR)					
17	Company plans to launch an innovative e-commerce platform with unique features not commonly found in the market. However, there's uncertainty about how users will react to these features. How would the Prototyping Model help reduce risks in this project? **The Prototyping Model can significantly reduce risks in the development of an innovative e-commerce platform with unique features by providing early feedback and allowing for rapid iteration. Here's how the model addresses uncertainty and risk:** **1. Early User Feedback:** **A prototype of the e-commerce platform can be developed early in the project, even if it's not feature-complete. By presenting this working model to potential users or stakeholders, the team can gather feedback on how users interact with the features and identify any usability issues or misunderstandings.** **Risk Reduction: This feedback helps the team validate assumptions about how users will react to the new features, enabling the company to adjust before investing heavily in full-scale development.** **2. Iterative Refinement:** **The Prototyping Model involves building prototypes, getting feedback, and refining the product iteratively. Each iteration provides a clearer understanding of what works and what doesn't.** **Risk Reduction: The platform's features can be continuously improved based on real-world testing and user responses, which minimizes the risk of developing features that users find confusing or unnecessary.** **3. Mitigating Scope Creep:** **Prototypes help clarify the specific features needed by stakeholders and users. Through user testing, stakeholders can prioritize features and functionalities based on actual feedback.**	10	4	1	9	1.6.1

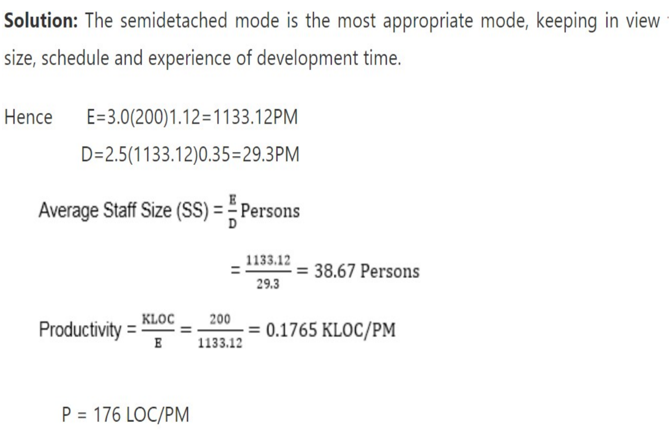
	Risk Reduction: By involving users early and iterating based on their input, scope creep is reduced, as the project team will be better aligned with user needs and expectations. **4. Clarification of Requirements:** **Prototyping allows the team to explore uncertain or complex features and get a clearer understanding of user needs. Stakeholders can see and experience the product early, helping to clarify ambiguous or unclear requirements.** **Risk Reduction: With better-defined requirements, the team is less likely to develop features that miss the mark or fail to meet user needs.** **5. Improved Communication Among Stakeholders:** **Prototypes act as a visual and tangible representation of the product, making it easier for stakeholders to understand the product vision. This improves communication between developers, designers, and stakeholders.** **Risk Reduction: Clearer communication ensures everyone is on the same page regarding project goals, expectations, and constraints, reducing the likelihood of misaligned goals and unexpected challenges.**					
18	You are assigned to a project where the stakeholders have vague ideas about the software they need. They often change their minds and provide conflicting information. As the Business Analyst, how would you approach the requirements elicitation process to ensure clarity and completeness? As a Business Analyst, when faced with stakeholders who have vague ideas, frequently change their minds, or provide conflicting information, it's crucial to adopt a structured and iterative approach to requirements elicitation. Here's how I would approach the process to ensure clarity and completeness: **1. Facilitate Clear Communication:** **Establish Clear Objectives: Begin by ensuring stakeholders clearly understand the project's purpose, scope, and objectives. This can be done through initial workshops or interviews, where I would set clear expectations regarding what is to be achieved and what is out of scope.** **Ask Open-Ended Questions: Use open-ended questions to uncover the underlying needs of stakeholders rather than just taking their statements at face value. This encourages them to think more deeply about their actual requirements.** **Active Listening: Ensure I'm actively listening to all stakeholders to understand their pain points, goals, and expectations, which will help in reconciling conflicting information later.** **2. Iterative Elicitation with Prototypes:** **Build Prototypes: Given the vagueness, I would consider building prototypes or mockups of the software early in**	10	4	2	9	2.6.3

	the process. This allows stakeholders to interact with the product, providing more concrete feedback and better understanding of their needs. Elicit Feedback Iteratively: Through iteration and feedback loops, gather input from stakeholders after each prototype or concept is reviewed. This enables gradual clarification of unclear requirements and helps identify any discrepancies or inconsistencies between stakeholders' expectations. 3. Use of Use Cases and User Stories: Define Use Cases: Use use cases or user stories to define real-world scenarios for how the software will be used. These documents help clarify the functional requirements in a way that is easy for stakeholders to understand. User-Centric Focus: Involve end-users, if possible, to ensure that the requirements are focused on what users need rather than abstract ideas. This keeps the scope grounded and helps manage changes effectively. 4. Conflict Resolution and Prioritization: Identify Conflicts Early: When conflicting information arises, I would facilitate discussions to understand the reasoning behind each perspective. This helps in reconciling differences by analyzing the trade-offs and agreeing on compromises. Prioritize Requirements: Work with the stakeholders to prioritize requirements based on their business value, feasibility, and dependencies. This ensures the most important features are clearly defined and agreed upon early in the project. Conflict Resolution: If conflicts persist, leverage tools like MoSCoW (Must have, Should have, Could have, Won't have) to facilitate decision-making and establish a clear path forward. 5. Document and Validate Requirements: Clear and Structured Documentation: I would document the requirements in clear, concise, and structured formats, such as SRS (Software Requirements Specifications) or user stories with acceptance criteria. Each requirement should be testable and traceable to avoid ambiguity. Frequent Validation: Regularly validate the requirements with stakeholders to ensure they reflect the actual needs and prevent scope drift. This can include walkthroughs or review meetings to confirm that requirements align with business goals. 6. Stakeholder Engagement: Regular Meetings and Updates: Ensure continuous engagement with stakeholders through regular meetings (e.g., workshops, sprint reviews). This helps in managing changes by ensuring that all parties are on the same page and can make timely decisions. Be a Mediator: As the BA, I would act as a mediator between stakeholders with conflicting ideas, ensuring their concerns are addressed while guiding them towards a shared solution. 7. Managing Changes and Expectations:					

<table>
<tr><td></td><td><b>Change Control Process: Establish a formal change control process to manage changes effectively, ensuring that changes are documented, evaluated, and approved based on their impact on scope, timeline, and cost.</b>
<b>Set Realistic Expectations: Communicate potential impacts of scope changes on timelines, budgets, and resources. Set expectations for the flexibility of the project scope and manage stakeholder demands realistically.</b></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td><b>(OR)</b></td><td colspan="5"></td></tr>
<tr><td>19</td><td>A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.



Category	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



Solution: The semidetached mode is the most appropriate mode, keeping in view size, schedule and experience of development time.

Hence E=3.0(200)1.12=1133.12PM
D=2.5(1133.12)0.35=29.3PM

Average Staff Size (SS) = $\frac{E}{D}$ Persons

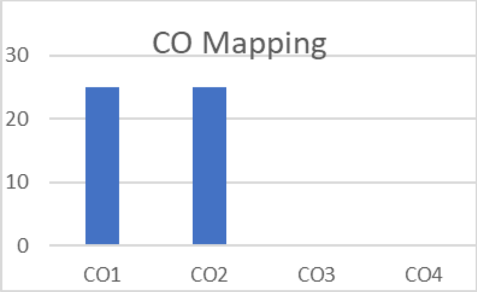
$= \frac{1133.12}{29.3} = 38.67$ Persons

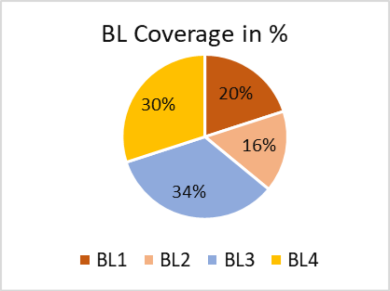
Productivity = $\frac{KLOC}{E} = \frac{200}{1133.12} = 0.1765$ KLOC/PM

P = 176 LOC/PM</td><td>10</td><td>3</td><td>2</td><td>9</td><td>1.6.1</td></tr>
</table>

***Performance Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

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





Approved by the Audit Professor/Course Coordinator