`

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | USN |  |  |  |  |  |  |  |  |  |  |  |  |   **School of Computer Science and Engineering**  **B. Tech (Hons.)**  **Midterm Question Paper – Set 2 Answer Scheme**  **Academic Year 2024-2025** | | |
| **Course: Agile Software Engineering and DevOps** | | **Course Code: CS2004** | **Semester: IV** |
| **Time: 9:30am to 11am** | **Duration: 90 minutes** | **Date: March 14, 2025** | **Max Marks: 25** |

**Notes/ Instructions:**

1. **Answer all questions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **PART A – (MCQs) Max Marks (5)** | **Marks** | **L1-L6** | **CO** |
|  | A Scrum team has an average velocity of 30 story points per sprint. They plan a sprint with 40 story points. What is the most likely outcome?  A. The team will complete all 40 points easily  B. The team may struggle to complete the sprint goal  C. Velocity will automatically increase to 40  D. The sprint duration will be extended  **Answer: B)** The team may struggle to complete the sprint goal  Reason: Velocity is an average measure, and planning beyond it may lead to incomplete work | 1 | L3 | CO2 |
|  | Identify the key benefit of Continuous Integration (CI):  A. Less frequent code integration  B. Reducing merge conflicts by integrating frequently  C. Delayed testing process  D. Testing only at the end of development  **Answer: B)** Reducing merge conflicts by integrating frequently  **Reason:** Continuous Integration ensures frequent code merging, minimizing integration issues | 1 | L3 | CO4 |
|  | A team’s burndown chart shows a flat line mid-sprint. What does this indicate?  A. The team is progressing steadily  B. Work is not being completed as expected  C. The team finished early  D. Story points increased mid-sprint  **Answer: B)** Work is not being completed as expected  Reason: A flat burndown chart indicates stalled progress; tasks are not getting completed | 1 | L2 | CO2 |
| 4. | In SAFe, who owns the Product Backlog?  A. Scrum Master  B. Product Owner  C. Agile Coach  D. Release Train Engineer  **Answer: B)** Product Owner  **Reason:** The Product Owner is responsible for defining and prioritizing the Product Backlog | 1 | L2 | CO2 |
| 5. | Which of the following is a characteristic of non-functional testing?  A. It focuses on specific behaviors of the software  B. It verifies that the software does what it's supposed to do  C. It evaluates the performance, security, and usability of the system  D. It is only performed at the end of the development cycle  **Answer: C)** It evaluates the performance, security, and usability of the system  **Reason:** Non-functional testing focuses on aspects e.g. performance, security, and usability rather than functionality | 1 | L2 | CO3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **PART B – Max Marks (20)** | **Marks** | **L1-L6** | **CO** |
| **6.** | **a.** Your Scrum team completes six user stories in a sprint, but only four meet the Definition of Done. How should the team handle the remaining two stories when delivering the Increment? **(2 Marks)**  **Answer:**  In Scrum, only **user stories that meet the Definition of Done (DoD)** are considered part of the Increment. Since **two stories are incomplete**, the team should:   1. **Move the unfinished stories back to the Product Backlog** – They are not part of the current Increment and should be re-prioritized for a future sprint. 2. **Analyze the reasons for incomplete work** – Identify blockers or scope issues to improve future sprint planning. 3. **Refine and re-estimate the stories if needed** – Ensure they are well-defined and achievable in an upcoming sprint. 4. **Communicate with stakeholders** – Set clear expectations that only the completed stories are included in the Increment.   This approach maintains **transparency, quality, and adherence to Scrum principles**.  **b.** Apply your understanding of the 3Cs (Card, Conversation, Confirmation) for creation of user stories and its significance **(3 Marks)**  **Answer:**  **Applying the 3Cs (Card, Conversation, Confirmation) in User Stories**   1. **Card** – A user story is written on a physical or digital card as a short, simple statement of a requirement. Example: *"As a user, I want to reset my password so that I can regain account access."* 2. **Conversation** – The development team discusses the user story with the Product Owner and stakeholders to clarify details, constraints, and expectations. Example: The team may ask, *"Should the password reset link expire after a certain time?"* 3. **Confirmation** – Acceptance criteria are defined to ensure the user story is complete and meets expectations. Example: *"The user should receive a password reset link via email, and it should expire after 30 minutes."*   **Significance of the 3Cs**   * **Ensures Clear Requirements** – Reduces misunderstandings. * **Encourages Collaboration** – Aligns developers, testers, and stakeholders. * **Defines Completion Criteria** – Ensures the story is testable and meets expectations.   By applying the **3Cs**, teams create user stories that are well-defined, actionable, and aligned with business needs. | 5 | L3 | CO2 |
| **7.** | **a.** Write a Git command sequence to clone a repository, create a new branch, switch to it, and push it to a remote repository **(3 Marks)**  **Answer:**  git clone <repository\_url>  git checkout -b new-branch  git add .  git commit -m "Initial commit"  git push origin new-branch  This sequence clones the repo, creates a branch, makes changes, and pushes it to GitHub/GitLab  **b.** Explain the difference between git pull and git fetch **(2 Marks)**  **Answer:**  **git fetch**  Retrieves the latest changes from the remote repository but does not merge them into the local branch. It is used when you want to review changes before applying them.  Example:  git fetch origin  git log origin/main --oneline # View new commits without merging  **git pull**  Fetches the latest changes and merges them into the current local branch.  Used when you want to update your branch with the latest remote changes immediately.  Example:  git pull origin main # Fetches and merges changes from the main branch  Key Difference:  git fetch = Retrieve remote changes without merging (safe for reviewing updates).  git pull = Fetch + Merge (directly updates the local branch).  Using git fetch first helps avoid conflicts before merging changes | 5 | L3 | CO4 |
| **8.** | Consider a scenario where you have the following tasks:   |  |  |  | | --- | --- | --- | | Task | Cost of Delay | Duration | | A | 80 | 20 | | B | 150 | 30 | | C | 60 | 10 |  1. Explain Weighted Shortest Job First (WSJF) **(1 Marks)**   **Answer:**  WSJF = **Cost of Delay / Duration** → Prioritizes tasks delivering the highest value in the shortest time.   1. Compute the WSJF score and rank the tasks **(3 Marks)**   **Answer:**  Formula:  WSJF=Cost of Delay/Duration  Task C (WSJF = 6.0) – Highest priority  Task B (WSJF = 5.0) – Second priority  Task A (WSJF = 4.0) – Lowest priority  Conclusion: Task C should be completed first, followed by B and then A, ensuring the highest value delivery in the shortest time   1. What are the core components of Agile Release Train? **(1 Mark)**   **Answer:**  Core Components of Agile Release Train (ART)   * PI Planning, System Demo, Iteration Execution, Inspect & Adapt, Agile Teams. | 5 | L3 | CO2 |
| **9.** | 1. Compare Regression Testing and Smoke Testing, give examples **(3 Marks)**   **Answer:**  Regression Testing vs. Smoke Testing  Regression Testing ensures that recent code changes do not break existing functionality. It is conducted after modifications, bug fixes, or new feature additions. Since it covers a broad scope, it often takes more time and is usually automated for efficiency.  Smoke Testing, on the other hand, is a preliminary test performed to check if the critical functionalities of a new build are working. It is quick and focuses on basic stability before moving to detailed testing.  Example:  In Regression Testing, if an e-commerce site adds a new payment method, testers will recheck all existing payment options to ensure they still work.  In Smoke Testing, after a new software build, testers verify if login, navigation, and core features function correctly before proceeding with deeper testing.  Both tests are essential, with Smoke Testing acting as an initial checkpoint and Regression Testing ensuring stability after changes.   1. Explain Load Testing and its significance **(2 Marks)**   **Answer:**  Load Testing is a type of performance testing that evaluates how a system behaves under expected and peak user loads. It measures response time, system stability, and resource utilization when multiple users access the system simultaneously.  Significance of Load Testing  Ensures Performance Stability – Identifies bottlenecks before real users experience slowdowns.  Prevents System Failures – Helps avoid crashes during high-traffic events.  Optimizes Scalability – Determines if the system can handle growth in user demand.  Improves User Experience – Ensures fast and smooth interactions, even under heavy load.  Example  An e-commerce website expects high traffic during a holiday sale. Load Testing simulates thousands of users browsing, adding items to the cart, and making payments. If the system slows down or crashes, developers can optimize it before the actual event.  This ensures the system remains reliable under heavy usage and prevents downtime. | 5 | L3 | CO3 |

Course Outcomes

|  |  |
| --- | --- |
| CO 1 | Evaluate the advantages and disadvantages of Agile development compared to traditional models |
| CO 2 | Assess various Agile methodologies such as Scrum, XP, Lean, and Kanban, and determine their appropriate applications |
| CO 3 | Create software requirements, design specifications, test plan and Analyze test coverage, requirements traceability for a software project |
| CO 4 | Utilize and implement various DevOps tools (e.g., Git, GitHub, Docker) in a software project |
| CO 5 | Develop a mini software project using Agile Scrum methodology, simulating its roles, meetings, processes, and artifacts |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Marks Distribution** | | | | | | | | | |
| **L1** | **L2** | **L3** | **L4** | **L5** | **L6** | **CO1** | **CO2** | **CO3** | **CO4** |
| 0 | 3 | 22 | 0 | 0 | 0 | 0 | 13 | 6 | 6 |