`

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | USN |  |  |  |  |  |  |  |  |  |  |  |  |   **School of Computer Science and Engineering**  **B. Tech (Hons.)**  **Midterm Question Paper – Set 1 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** |
|  | Apply your understanding of Epics, Features, and Stories in Agile to illustrate their relationship?  A. Stories are part of Epics, and Epics are part of Features  B. Features are part of Epics, and Epics are part of Stories  C. Epics, Features, and Stories are unrelated  D. Epics are broken down into Features, and Features are broken down into Stories  **Answer: D.** Epics are broken down into Features, and Features are broken down into Stories | 1 | L3 | CO2 |
|  | Apply your understanding of DevOps to identify which of the following is NOT a core component  A. Continuous Integration  B. Continuous Delivery  C. Manual Documentation  D. Automation  **Answer: C.** Manual Documentation | 1 | L3 | CO4 |
|  | “Burnt Down Chart” primarily tracks the following  A. The number of bugs found during testing  B. The amount of work remaining and time left to complete it  C. The team's mood throughout the sprint  D. The number of meetings held during the sprint  **Answer: B.** The amount of work remaining and time left to complete it | 1 | L2 | CO2 |
| 4. | Who is responsible for facilitating PI Planning in SAFe?  A. Scrum Master  B. Product Owner  C. Release Train Engineer (RTE)  D. Portfolio Manager  **Answer: C.** Release Train Engineer (RTE) | 1 | L2 | CO2 |
| 5. | Identify a key best practice for creating an effective test plan?  A. Ignore project requirements  B. Focus only on functional testing  C. Clearly define the scope of testing  D. Avoid documenting potential risks  **Answer: C.** Clearly define the scope of testing | 1 | L2 | CO3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **PART B – Max Marks (20)** | **Marks** | **L1-L6** | **CO** |
| **6.** | **a.** Why is the "Definition of Done" critical in Scrum? Give 1 example **(2 Marks)**  **Answer:**  The "Definition of Done" ensures shared understanding among team members about when a task is complete, improving quality and consistency. *Example:* In a Scrum project, a feature is considered "Done" only when it passes unit tests, integration tests, and is deployed to a staging environment.  **b.** Apply your understanding of the INVEST acronym to explain its meaning for user stories and its significance **(3 Marks)**  **Answer:**  *INVEST* in user stories:   * **I**ndependent (self-contained) * **N**egotiable (flexible) * **V**aluable (delivers value) * **E**stimable (can be estimated) * **S**mall (manageable size) * **T**estable (clear acceptance criteria)   *Significance:* Ensures clarity, prioritization, and efficient planning in Agile development. | 5 | L3 | CO2 |
| **7.** | Consider a scenario where you need to create a new branch named "feature-x" in Git, switch to that branch, make some changes, add the changes to the staging area, commit the changes, and finally push the branch to a remote repository named "origin".   1. Give the sequence of Git commands to accomplish this task **(2 Marks)**   **Answer:**  Git Commands:  git branch feature-x # Create a new branch  git checkout feature-x # Switch to the branch  # or git switch feature-x (for newer versions)  git add . # Stage changes  git commit -m "Added changes" # Commit changes  git push origin feature-x # Push branch to remote repository   1. You have completed your work on the "feature-x" branch and give the sequence of commands to merge it into the "main" branch. **(3 Marks)**   **Answer:**  git checkout main # Switch to the main branch  git pull origin main # Fetch and merge the latest changes from the remote main branch  git checkout feature-x # Switch back to the feature branch  git merge main # Merge the latest main branch changes into feature-x  # If conflicts occur, manually resolve them in the affected files  git add . # Stage resolved files  git commit -m "Resolved merge conflicts and updated feature-x"  git checkout main # Switch back to main branch  git merge feature-x # Merge the feature branch into main  git push origin main # Push the updated main branch to the remote repository | 5 | L3 | CO4 |
| **8.** | Consider a scenario where you have the following tasks:   |  |  |  | | --- | --- | --- | | Task | Cost of Delay | Duration | | A | 60 | 20 | | B | 120 | 30 | | C | 50 | 10 |  1. Explain Weighted Shortest Job First (WSJF) **(1 Marks)**   **Answer:**  *WSJF (Weighted Shortest Job First)* prioritizes tasks by dividing the *Cost of Delay* by the *Duration* to maximize value delivery.   1. Calculate the WSJF score for each task and rank them in order of priority   **(3 Marks)**  **Answer:**  **WSJF Scores:**   * Task A: **60/20 = 3** * Task B: **120/30 = 4** * Task C: **50/10 = 5** **Priority Order:** C > B > A   **c.**Describe the key characteristics of an Agile Release Train **(1 Mark)**  **Answer:**  *Agile Release Train (ART)*   * Aligns multiple teams working together. * Works in Program Increments (~8-12 weeks). * Uses PI Planning for coordination. * Includes cross-functional teams. | 5 | L3 | CO2 |
| **9.** | **a.** Distinguish between integration testing and unit testing in terms of their scope and objectives by giving one example **(3 Marks)**  **Answer:**   * *Unit Testing:* Tests individual components (e.g., a single function in a module). * *Integration Testing:* Tests interactions between multiple components (e.g., API calls between frontend and backend).   **Example:**   * *Unit Test:* Checking if a function correctly calculates sum. * *Integration Test:* Ensuring the payment gateway correctly interacts with the order system.   **b.** Illustrate the concept of non-functional testing by providing two relevant examples **(2 Marks)**  **Answer:**  Non-Functional Testing Examples:   * *Performance Testing:* Ensuring a website loads within 2 seconds under high traffic. * *Security Testing:* Validating that user authentication prevents unauthorized access. | 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 |