Fast**National University of Computer & Emerging Sciences, Karachi  
Quiz – I (Spring-2023)**

|  |  |  |
| --- | --- | --- |
| Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Section: \_\_\_\_\_\_\_\_\_ |

**Q1:**

* Incremental model
* Agile approach
* Check for students’ reasons and marks them correct accordingly if their valid reasons validate their statements.

**Q2.**

* Scrum is an Agile project management methodology that is designed to help teams work together more effectively to deliver high-quality products. It is based on iterative and incremental development, where work is divided into small chunks called sprints. Here is an overview of how the Scrum technique works:

1. Product backlog: The Scrum process begins with the product backlog, which is a list of all the features, enhancements, and bug fixes that the team plans to work on during the project. The product backlog is owned by the product owner, who is responsible for prioritizing the backlog items based on business value.
2. Sprint planning: The team then meets to plan the next sprint, which is a time-boxed period typically lasting 1-4 weeks during which the team will work on a set of backlog items. During the sprint planning meeting, the team selects a set of items from the product backlog that they believe they can complete during the sprint.
3. Sprint backlog: The selected items are then moved from the product backlog to the sprint backlog, which is a list of all the items that the team plans to work on during the sprint. The team then creates a plan for how they will complete the items, breaking them down into smaller tasks.
4. Daily stand-up: During the sprint, the team holds daily stand-up meetings, also known as daily scrums, where each team member answers three questions: What did I do yesterday? What will I do today? Are there any impediments in my way?
5. Sprint review: At the end of the sprint, the team holds a sprint review meeting, where they demonstrate the work they have completed to the product owner and any other stakeholders. The team also reviews the product backlog and makes any necessary adjustments.
6. Sprint retrospective: The team then holds a sprint retrospective meeting, where they review the sprint and identify ways to improve the process for the next sprint.

The Scrum technique is designed to be flexible and adaptable, allowing teams to adjust their process based on feedback and changing requirements. The focus on iterative development and continuous feedback helps ensure that the team is delivering high-quality products that meet the needs of the customer.

* based on the information provided, it is likely that ABC Co. is operating at Level 3: Defined of the Capability Maturity Model, given their structured working environment and their focus on producing high-quality, innovative products.Top of Form

**Q3.**

* 1. Pair programming is a software development technique in which two programmers work together on the same task at the same workstation. While this approach has some potential advantages, it is not always a practical solution and is not universally adopted by organizations. Here are some reasons why:

1. Cost: Pair programming requires two developers to work on a single task, effectively doubling the cost of development for that task. This can be an issue for organizations with limited budgets or tight deadlines.
2. Team dynamics: Pair programming requires two people to work closely together for extended periods, which can be challenging if they do not have a good working relationship or if their communication styles do not match well.
3. Skills and experience: Pair programming requires two developers who have compatible skill levels and experience. If one developer is significantly more experienced or skilled than the other, the pairing may not be effective.
4. Productivity: While some studies have shown that pair programming can lead to improved code quality and reduced defects, it is not always clear that it leads to higher overall productivity. Some developers may find it difficult to stay focused or may feel that they could complete the task more quickly on their own.
5. Individual work preferences: Not all developers enjoy pair programming or find it to be an effective way to work. Some prefer to work independently, while others may prefer to collaborate in different ways, such as through code reviews or team discussions.

In summary, while pair programming can be a useful technique in certain situations, it is not always a practical solution. The cost, team dynamics, skills and experience, productivity, and individual work preferences all factor into whether or not it is a good fit for a particular organization or project. Ultimately, the decision to use pair programming should be based on an assessment of the benefits and drawbacks, as well as the needs of the project and the team's working style.

B) The user stories and task cards approach is a popular agile software development methodology for managing project tasks and requirements. Here are some advantages and disadvantages of this approach:

* Advantages:
* Focus on the user: User stories are centered around user needs, which helps ensure that the development team is focused on delivering value to the end-user. This approach emphasizes understanding the user’s perspective, which can lead to higher satisfaction with the final product.
* Collaborative: The user stories and task cards approach is a collaborative effort involving developers, product owners, and other stakeholders. This helps to ensure everyone is on the same page, working towards the same goals, and avoids confusion or misunderstanding.
* Flexibility: The approach allows for flexibility and adaptability, enabling teams to respond quickly to changes in requirements or priorities. User stories are typically small, manageable pieces of work, which means that they can be reprioritized or changed without disrupting the entire development process.
* Clear Priorities: The approach encourages prioritization and visibility of tasks, making it easier to manage workload, identify bottlenecks, and ensure that the most critical work is addressed first.
* Disadvantages:
* Time-consuming: The user stories and task cards approach can be time-consuming and may require a lot of documentation. Creating and maintaining user stories can be time-consuming, and if not managed well, can lead to the process becoming a bottleneck.
* Limited scope: The approach may not be suitable for complex projects with multiple interdependent systems or components. In these cases, user stories may not adequately capture all the dependencies, leading to unforeseen issues later in the development cycle.
* Overemphasis on user needs: Focusing on user needs can sometimes lead to a lack of attention on technical details, which can result in overlooked issues or quality concerns.
* Potentially confusing: The user stories and task cards approach can be confusing for newcomers to the methodology, as it requires a good understanding of the principles behind agile development and how user stories are written and managed.
* In summary, the user stories and task cards approach has several advantages, including a focus on the user, collaboration, flexibility, and clear priorities. However, it can also be time-consuming, may have limited scope, overemphasize user needs, and be confusing for newcomers. Ultimately, the decision to use this approach will depend on the specific requirements of the project and the team's ability to manage the process effectively.