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

**17th Feb 2023**

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

**Q1:**

Based on the given information, it can be inferred that Symantec is operating at CMM Level 4, which is the Managed level. CMM (Capability Maturity Model) is a framework that describes the maturity of an organization's software development processes. CMM Level 4 is defined as the level at which an organization has established a set of defined processes, which are followed by the development team to achieve predictable quality results. Symantec's use of prototyping approaches and defined sub-activities for each process indicate a defined and managed software development process. Additionally, their use of techniques to predict system performance suggests that they are focusing on quantitative management and are actively monitoring and controlling their software development processes. These practices are characteristic of organizations operating at Level 4 of the CMM.

**Q2.**

In the case of the Al-Shifa hospital's digital system, an Agile approach is more appropriate than a plan-driven approach. Agile is a flexible and iterative approach that allows for changes to be made based on customer feedback and evolving requirements. It involves collaboration between the development team and the customer to deliver a working product in small increments. In the case of Al-Shifa hospital, an Agile approach allows for continuous input and feedback from medical staff and patients, which can help to create a more usable and efficient system. Additionally, security is a key requirement for the system, and Agile development includes regular testing and evaluation to identify and address any vulnerabilities. This can lead to a more secure and robust system in the long run. On the other hand, a plan-driven approach is more suitable for projects with fixed requirements and a well-defined scope. In the case of Al-Shifa hospital's digital system, requirements may change over time, and a plan-driven approach may not be able to accommodate those changes effectively.

**Q3.**

* The maintenance cost for a software is often greater than the development cost due to a number of factors. First, as software is used, bugs and issues are discovered, and it is necessary to fix these issues to keep the software running properly. This requires ongoing effort from developers and support staff, which can be expensive. Additionally, as technology evolves, the software may need to be updated to keep up with new hardware or software platforms, which can be a major undertaking. Furthermore, user requirements often change over time, and the software may need to be updated to meet these changing needs. This can also be time-consuming and expensive, especially for complex software systems. Finally, software maintenance often requires specialized skills and expertise, which can be in short supply and command higher salaries than more general development skills. All of these factors contribute to the higher cost of software maintenance compared to development.
* Test-first development, also known as Test-Driven Development (TDD), is an approach to software development that involves writing automated tests before writing the actual code. This approach has several benefits, which is why it is considered a better approach: Improved Code Quality: Writing tests first forces developers to think about the design and architecture of the code before they write it. This helps ensure that the code is well-structured, modular, and easy to maintain. Faster Feedback: By writing tests first, developers can get immediate feedback on whether the code is working or not. This helps catch bugs early in the development cycle, which reduces the time and cost of fixing them later. Increased Productivity: TDD helps developers focus on what the code should do, rather than how it should be implemented. This can lead to more efficient and effective coding, resulting in increased productivity. Better Collaboration: TDD promotes collaboration between developers and testers, as both groups are involved in writing and reviewing tests. This can lead to better communication and a shared understanding of the requirements and expectations for the code. Overall, **the test-first development approach can help improve code quality, reduce costs, increase productivity, and promote collaboration, making it a better approach to software development.**