

## CS611 Curreant paper's solved By Kainat Rajpoot

**Q. What are the three main activities in software testing?**

A. The three main activities in software testing are:

- **Test Planning:** This involves defining objectives, scope, and resources for testing, as well as creating test plans and strategies.
- **Test Execution:** This is where actual testing takes place according to the test plans, including running test cases, identifying defects, and collecting test results.
- **Test Reporting:** After executing tests, the results are documented, defects are reported, and overall test summaries are provided to stakeholders, aiding decision-making processes.

**Q. Do you agree that code is main artifact in testing? [Yes/No]. Also describe how testing is performed.**

A. Yes, the code is indeed the main artifact in testing because testing aims to verify that the code behaves as expected and meets the requirements. Testing is performed by executing the code through various test cases, which are designed to cover different scenarios and functionalities. Test results are then analyzed to identify any discrepancies between expected and actual behavior, helping to ensure the quality and reliability of the software.

**Q. Write any two impacts of Unit Testing.**

A. Two impacts of unit testing are:

- **Early Bug Detection:** Unit testing helps catch bugs or defects in the code at an early stage of development, reducing the cost and effort required to fix them later in the development cycle.
- **Improved Code Quality:** By testing individual units of code in isolation, unit testing encourages developers to write modular, well-structured code, leading to higher code quality and easier maintenance.

**Q. Do you agree that "Requirement Engineering" is a purely Technical Activity? [Yes/No]. Justify your answer with solid reason.**

A. No, "Requirement Engineering" is not purely a technical activity. It involves understanding and eliciting stakeholders' needs, which often requires communication and negotiation skills. Additionally, it involves translating these needs into technical specifications, bridging the gap between technical and non-technical stakeholders. Therefore, it encompasses both technical and non-technical aspects, making it a multidisciplinary activity.

**Q. Do you agree that Software Requirements are similar to requirements in other domains? [Yes/No].**

- A. Yes, I agree. Software requirements share similarities with requirements in other domains because they both aim to define what needs to be achieved or delivered. However, software requirements often have unique characteristics due to the intangible nature of software and its dynamic environment, which may require specific techniques and considerations during the requirements elicitation and management process.

**Q. Describe one difference or one similarity between Software Requirement and requirements of other domains.**

A. One difference between software requirements and requirements in other domains is that software requirements often involve technical specifications and dependencies on software architecture, whereas requirements in other domains may focus more on functional outcomes or physical constraints. However, a similarity is that both types of requirements aim to capture stakeholders' needs and define the desired outcomes or behaviors to be achieved.

**Q. Do you agree that correctness centric activities are basics of quality assurance? [Yes/No]. Enlist any two objectives of Quality Assurance.**

A. Yes, I agree. Correctness-centric activities, such as ensuring the software function as intended and meets requirements, form the foundation of quality assurance. Two objectives of Quality Assurance are:

- **Ensuring Customer Satisfaction:** Quality assurance aims to deliver products or services that meet or exceed customer expectations, ensuring satisfaction and loyalty.
- **Continuous Improvement:** Quality assurance seeks to identify areas for improvement in processes, products, and services, fostering a culture of continuous enhancement and innovation.

**Q. What is verified and ensured by Regression Testing? Explain with an example.**

A. Regression testing verifies and ensures that recent changes or enhancements to a software application haven't adversely affected existing functionalities. It ensures that previously working features still function correctly after modifications.

For example, let's consider a banking application that allows users to transfer funds between accounts. After software update aimed at improving the user interface, the development team conducts regression testing to ensure that the fund transfer functionality still works as expected. They execute test cases covering various scenarios, such as transferring different amounts of money between different types of accounts, verifying that the transfer process remains accurate and reliable despite the UI changes.

**Q. Define Earned Value Management (EVM). Enlist three types of value used in EVM.**

A. Earned Value Management (EVM) is a project management technique that helps track the performance and progress of a project by integrating measurements of scope, schedule, and

cost. It compares the actual work accomplished (earned value) against the planned work (planned value) and the actual costs incurred (actual cost).

Three types of value used in EVM are:

- **Planned Value (PV):** The planned value represents the authorized budget allocated for the planned work to be accomplished by a specific point in time.
- **Earned Value (EV):** Earned value represents the value of the work actually completed and accepted at a specific point in time, measured in terms of the budget allocated to that work.
- **Actual Cost (AC):** The actual cost represents the total costs incurred in completing the work up to a specific point in time.

**Q. To fulfill the new requirements in projects, how project team work performs?**

**A.** To fulfill new requirements in projects, the project team collaborates closely to understand the new needs, assess their impact on the project, and adjust plans accordingly. They may conduct discussions, brainstorming sessions, and feasibility studies to determine the best approach for incorporating the new requirements into the project scope and timeline, ensuring that all stakeholders' expectations are met.

**Q. Why post-release defects are costly to fix?**

**A.** Post-release defects are costly to fix because they require additional resources, time, and effort to identify, diagnose, and rectify after the software has been deployed to users. Moreover, they can result in negative impacts on user experience, reputation, and customer satisfaction, leading to potential loss of revenue and increased support costs.

**Q. Attributes of software testing requirements..**

**A.** Attributes of software testing requirements include:

- **Clarity:** Requirements should be clear and unambiguous, ensuring that testers understand what needs to be tested and how.
- **Completeness:** All aspects of the software functionality, including both functional and non-functional requirements, should be specified to ensure comprehensive testing coverage.
- **Testability:** Requirements should be designed in a way that allows for effective testing, with clear criteria for determining whether the software meets the specified requirements.
- **Traceability:** Requirements should be traceable to higher-level system requirements and design specifications, facilitating impact analysis and change management.
- **Prioritization:** Requirements should be prioritized based on their importance and impact on the software, allowing testers to focus on critical areas during testing.

- **Consistency:** Requirements should be consistent with each other and with the overall project objectives, avoiding contradictions or conflicts that could lead to confusion during testing.
- **Verifiability:** Requirements should be verifiable, meaning that their correctness and completeness can be objectively determined through testing or inspection.

**Q. Anlist any 5 level of testing?**

Here are five levels of software testing:

1. **Unit Testing:** Testing individual components of the software.
2. **Integration Testing:** Testing the interaction between different components.
3. **System Testing:** Testing the entire system as a whole.
4. **Acceptance Testing:** Verifying the system meets the user's requirements.
5. **Regression Testing:** Checking that changes haven't broken existing functionality.

**Q. white box testing**

**A.** White box testing, also known as structural testing or glass box testing, is a software testing technique where the internal structure, code, and logic of the software application are examined. Testers have access to the source code and use their knowledge of how the software is implemented to design test cases that evaluate the correctness of the code paths, data flows, and control flows within the application.

**Q. blackbox testing.**

**A.** Black box testing is a software testing technique where the internal structure, code, and logic of the software application are not known to the tester. Testers focus on testing the functionality and behavior of the software by providing inputs and observing outputs, without knowledge of the underlying implementation. It is akin to testing a "black box" where only the inputs and outputs are visible.

1. **Use case template..**
2. **impact of unit testing...**
3. **correctness**
4. **code artifact..**
5. **one question is conceptual so I left it ....**
6. **Three main types of software testing**
7. **When the client demand to add new requirements in the current project then how project team work**
8. **Q. SPI formula?**
9. **Value Type of Earned value management**  
Most mcqs from pre assessment or post assessment...