

**Birla Institute of Technology & Science, Pilani**  
**Work Integrated Learning Programmes Division**  
**Second Semester 2024-2025**

**Mid-Semester Test**  
**(EC-2 Regular)**

Course No. : SE ZG501  
Course Title : Software Quality Assurance and Testing  
Nature of Exam : Closed Book  
Weightage : 30%  
Duration : 2 Hours  
Date of Exam : 25/03/2025(AN)

No. of Pages	= 2
No. of Questions	= 6

**Note to Students:**

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

**Answer Key**

**Q1. A small online shopping website is developed to allow customers to add items to their cart and proceed to checkout. However, some problems arise during testing and after deployment:**

1. A developer accidentally writes incorrect code for adding items to the cart. Instead of increasing the quantity, the system removes the item.
2. During testing, a tester tries to add two items to the cart, but instead of showing 2 items, the cart appears empty.
3. After launch, many customers complain that their cart is empty after adding products, making them unable to place orders.

**Based on the scenario provided.**

**i) Define and differentiate between defect, error, and failure using the given case.**

**Marks-2**

**ii) What methods can be used to identify each of these issues, and what strategies should be employed to fix them?**

**Marks-3**

**Answer:**

**(i) Define and Differentiate Between Defect, Error, and Failure (Marks: 2)**

An **error** is a mistake made by the developer in the code. In this case, the developer accidentally writes incorrect code for adding items to the cart, where instead of increasing the quantity, the system removes the item. A **defect (bug)** is when this error is detected during testing. Here, during testing, a tester tries to add two items to the cart, but instead of showing two items, the cart appears empty. A **failure** occurs when the defect is not fixed and affects real users. After launch, many customers complain that their cart is empty after adding products, making them unable to place orders.

**(ii) Methods to Identify and Fix Each Issue (Marks: 3)**

The **error** in the code, where adding an item removes it instead, can be identified through **code reviews and unit testing**, ensuring the logic in the add-to-cart function is correct. It can be fixed by updating the code to correctly increase the item count instead of decreasing it.

The **defect**, where the cart appears empty after adding items, can be detected through **functional testing and debugging**. Testers should verify that items are stored correctly in the database and displayed in the cart. Fixing this requires correcting database queries and ensuring session management properly updates cart contents.

The **failure**, where customers find their cart empty after adding products, can be identified through **user complaints and analysing system logs**. It can be fixed by implementing **data persistence techniques** to ensure cart data is not lost after adding products. Additionally, **load testing** should be performed to check if multiple users can add items without issues.

**Q2. a) Analyze the differences between Quality Assurance (QA) and Quality Control (QC).**

Marks-1

**b) A software development team is building a library management system for a university. During development, the team discovers that some key requirements were not clearly defined or were misunderstood, such as the rules for book reservations, borrowing limits, and overdue fine calculations. As a result, they need to make significant modifications, causing delays and extra development effort to align the system with the library's actual policies.**

**i) Explain how unclear and incomplete requirements can affect software quality in this scenario?**

Marks- 2

**ii) What best practices could the team have followed to ensure that all requirements were well-defined and correctly understood from the start?**

Marks- 2

**Answer:**

**Q2 (a) Differences Between Quality Assurance (QA) and Quality Control (QC) (Marks: 1)**

**Quality Control (QC)** involves checking a finished product to ensure it meets quality standards.

**Quality Assurance (QA)** focuses on preventing defects by improving processes throughout development and manufacturing. The goal is to reduce costs while ensuring quality.

**2b. (i) Impact of Unclear and Incomplete Requirements on Software Quality**

(Marks: 2)

Unclear and incomplete requirements cause **misinterpretations, missing features, and delays**, affecting software quality. In this case, misunderstandings about **book reservations, borrowing limits, and overdue fines** led to major modifications, increasing development time and cost. Without clear requirements, the system may have **incorrect functionality, usability issues, and inconsistencies**, making it unreliable for university

staff and students. Properly defined requirements help prevent defects and ensure the system functions as intended.

**(ii) Best Practices to Ensure Well-Defined and Understood Requirements (Marks: 2)**

To avoid these issues, the team should have followed **best practices for requirement gathering**:

1. **Stakeholder Discussions** – Engaging with librarians and students to clarify rules for book reservations and fines.
2. **Detailed Documentation** – Clearly defining all system requirements before starting development.
3. **Prototyping & Validation** – Creating a prototype to gather early feedback and confirm requirements.
4. **User Stories & Use Cases** – Defining real-world scenarios to ensure proper implementation.
5. **Regular Review Meetings** – Continuously validating requirements to prevent misunderstandings.

By following these steps, the team could have **minimized errors, reduced rework, and ensured a high-quality software product.**

**Q3. a) A company is developing a new e-commerce platform to handle high traffic, secure transactions, and seamless user experience. Management is debating whether to invest in Software Quality Assurance (SQA) or focus on marketing and feature expansion.**

**Some argue that early SQA investment ensures stability and security, while others believe prioritizing faster deployment will drive quicker growth. However, neglecting quality early could lead to security breaches, performance issues, and costly failures.**

**Illustrate why it is necessary and profitable for the company to invest in Software Quality Assurance (SQA). Include considerations of cost, risk, and long-term benefits in your arguments.**

**Marks - 3**

**b) How does reducing prevention costs affect the overall cost of quality? Explain the relationship between prevention, appraisal, and failure costs in software development.**

**Marks -2**

**Answer**

**Q3. (a) Importance and Profitability of Investing in Software Quality Assurance (SQA)**

**(3 Marks)**

Investing in Software Quality Assurance (SQA) ensures stability, security, and long-term profitability for an e-commerce platform. While prioritizing marketing and features may boost short-term growth, neglecting quality early can lead to security breaches, performance issues, and costly failures.

**Key Benefits of SQA Investment:**

1. **Cost Efficiency:** Early defect detection reduces expensive post-release fixes.
2. **Risk Mitigation:** Prevents security vulnerabilities and system failures that could damage reputation.

3. Long-Term Benefits: Ensures customer trust, reduces maintenance costs, and enhances scalability.

Investing in SQA early leads to higher customer satisfaction, lower failure costs, and long-term business success.

**Q3. (b) Impact of Reducing Prevention Costs on Overall Cost of Quality (2 Marks)**

Reducing prevention costs (such as training and defect prevention) increases overall quality costs due to higher failure rates.

1. Prevention Costs help avoid defects early, reducing rework and testing costs.
2. Lower Prevention = Higher Failure Costs, leading to expensive post-release fixes, security risks, and customer dissatisfaction.

Ignoring prevention may save money short term, but it increases failure costs and damages business reputation in the long run.

**Q4. a) What are the three perspectives of the McCall Model?**

**Marks-1**

The **three key perspectives** of the **McCall Model** in software quality assessment are:

1. **Product Revision** – Focuses on maintainability, flexibility, and testability.
2. **Product Transition** – Ensures portability, reusability, and interoperability.
3. **Product Operation** – Evaluates correctness, reliability, efficiency, and usability.

**b) What is the primary focus of the McCall Model?**

**Marks-1**

The McCall quality model was primarily aimed at software product quality (i.e., the internal perspective) and did not easily tie in with the perspective of the user who is not concerned with technical details.

**c) How does quality measurement contribute to software program acquisition?**

**Marks-1**

Quality measurement in software acquisition ensures that the software meets agreed-upon quality objectives for users. It helps in setting contractual commitments, verifying compliance with requirements, and measuring quality before adapting and releasing the software.

**d) How does the PDCA (Plan-Do-Check-Act) cycle in ISO 9001 help improve quality? (Explain in two sentences).**

**Marks-1**

The PDCA (Plan-Do-Check-Act) cycle in ISO 9001 helps improve quality by ensuring processes are well-planned, properly executed, and regularly evaluated. It promotes continuous improvement by identifying risks, implementing corrective actions, and seizing opportunities for enhancement.

**e) List the five stages of the service lifecycle in the ITIL framework.**

**Marks-1**

The five stages of the service lifecycle in the ITIL framework are:

1. Service Strategy
2. Service Design
3. Service Transition
4. Service Operation
5. Continual Service Improvement

**Q5. a) Differentiate between static testing and dynamic testing.**

**Marks- 2**

**Static Testing:** Conducted during the requirement analysis and design stages, it ensures the SRS aligns with user requirements. Techniques include code reviews, inspections, walkthroughs, and software technical reviews (STRs).

**Dynamic Testing:** Begins when code or modules are ready and involves executing the software to validate functionality. Techniques include black-box testing (focusing on inputs/outputs), gray-box testing (combining internal and external testing), and white-box testing (examining internal structures and logic).

**b) A school management system is being tested to ensure that it correctly accepts and validates student grades. The system only allows grades between 0 and 100 (inclusive). Using Boundary Value Analysis, what test cases would you create to ensure that the application correctly handles the valid and invalid age inputs? List the test cases with their expected outcomes.**

**Marks – 3**

**Boundary Value Analysis (BVA) Test Cases for Student Grades**

The **valid range** for student grades is **0 to 100 (inclusive)**. Using **Boundary Value Analysis (BVA)**, the test cases are:

1. **-1** (Below the minimum boundary) – Expected outcome: **✗ Invalid (Out of Range)**
2. **0** (Minimum boundary) – Expected outcome: **✓ Valid**
3. **1** (Just above the minimum) – Expected outcome: **✓ Valid**
4. **99** (Just below the maximum) – Expected outcome: **✓ Valid**
5. **100** (Maximum boundary) – Expected outcome: **✓ Valid**
6. **101** (Above the maximum boundary) – Expected outcome: **✗ Invalid (Out of Range)**

**Explanation:**

- The test cases check values **just outside and inside** the valid range to ensure the system correctly handles **edge cases**.
- This method helps identify potential **boundary errors** in grade validation.

**Q6. a) What is functional testing, and how does it differ from nonfunctional testing?**

**Marks – 2**

Functional testing verifies that the system performs as expected based on requirements, focusing on what the system should do (e.g., login, transactions).

Non-functional testing evaluates system performance, security, usability, and reliability, ensuring how well the system operates under different conditions.

<b>Functional testing</b>	<b>Non functional testing</b>
<b>1.</b> It involves the product's functionality.	<b>1.</b> It involves the product's quality factors.
<b>2.</b> Failures, here, occur due to code.	<b>2.</b> Failures occur due to either architecture, design, or due to code.
<b>3.</b> It is done during unit, component, integration, and system testing phase.	<b>3.</b> It is done in our system testing phase.
<b>4.</b> To do this type of testing only domain of the product is required.	<b>4.</b> To do this type of testing, we need domain, design, architecture, and product's knowledge.
<b>5.</b> Configuration remains same for a test suite.	<b>5.</b> Test configuration is different for each test suite.

**b) Explain the types of Automated Testing. List the advantages of automated testing.**

**Marks – 3**

**Types of Automated Tests**

- 1. Code Auditing:** Automatically checks code compliance with standards, reporting deviations and statistical summaries.
- 2. Coverage Monitoring:** Analyzes test coverage, reporting the percentage of covered and uncovered lines.
- 3. Functional Tests:** Automates black-box correctness tests by executing pre-recorded test cases and generating error reports.
- 4. Load Tests:** Ensures system stability under full load, preventing failures that could cause significant operational damage.

**Advantages of automated tests**

- (1) Accuracy and completeness of performance.
- (2) Accuracy of results log and summary reports.
- (3) Comprehensiveness of information.
- (4) Few manpower resources required to perform tests.
- (5) Shorter duration of testing.
- (6) Performance of complete regression tests.
- (7) Performance of test classes beyond the scope of manual testing.

**[2 marks for types of automated tests and 1 mark for the advantages]**