



UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2025 – 3rd Year Examination – Semester 5

***IT6206 – Software Quality Assurance
Structured Question Paper***

(TWO HOURS)

To be completed by the candidate

Important Instructions:

- The duration of the paper is **2 (Two) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **13 pages**.
- **Answer all questions.** All questions **do not** carry **equal** marks.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.
If a page is not printed, please inform the supervisor immediately.
- All kinds of electronic devices including calculators are **not** allowed.
- *All Rights Reserved.*

Questions Answered

Indicate by a cross (×), (e.g.) the numbers of the questions answered.

To be completed by the candidate by marking a cross (×).	Question numbers			
	1	2	3	4
To be completed by the examiners:				

1)

- (a) Define the concept of **Software Quality Assurance (SQA)** and explain two objectives of SQA in Maintenance. (03 Marks)

ANSWER IN THIS BOX

Quality Assurance: A set of activities which detect and prevent the causes of errors, and correct errors early in the development process.

Objectives:

- Assuring that the software maintenance activities are satisfying the functional and technical requirements of the system.
- Assuring that the software maintenance activities are satisfying the managerial scheduling and budgetary requirements.
- Initiating and managing activities to improve and increase the efficiency of software maintenance and software quality assurance activities

- (b) There are four *Software Testing Levels* listed in the answer box. Select the correct **Primary Objective** for each testing level using the list of objectives given below. Write down the correspond letter of the correct objective in the box provided. (1x4 = 04 marks)

List of Objectives

- A. Verify complete system requirements
- B. Verify individual components
- C. Verify interaction between modules
- D. Verify business/user requirements

Testing Level	Answer
1. Unit Testing	B
2. Integration Testing	C
3. System Testing	A
4. Acceptance Testing	D

- (c) A software company develops an e-commerce website that is updated every two weeks. The QA team is considering introducing automated testing for regression testing, while continuing manual testing for exploratory testing.

- (I) Explain three (03) advantages of using automated testing in this project. (03 Marks)

ANSWER IN THIS BOX

- Accuracy and completeness of performance
- Higher accuracy in documentation
- Comprehensive information
- Lesser amount of manpower resources in test executions
- Shorted testing periods

- (II) Explain three (03) disadvantages or risks associated with automated testing. (03 Marks).

ANSWER IN THIS BOX

- ☐ **High initial cost** – Tools, infrastructure, and script development require investment.
- ☐ **Maintenance effort** – Test scripts must be updated when the application changes.
- ☐ **Not suitable for all testing types** – Exploratory, usability, and ad-hoc testing still require human testers.
- ☐ High manpower resources for test preparation

- (I) Based on the given scenario in part (c), justify whether the company should invest in test automation. (02 Marks)

ANSWER IN THIS BOX

The company should invest in automation because frequent software releases require repeated regression testing. Automated tests can reduce testing time, improve consistency, and support continuous delivery. However, automation should complement rather than replace manual testing, especially for exploratory and usability testing

(II)

- (d) State whether the following statements are True (T) or False (F) based on Software Testing Techniques and their Characteristic. (1 × 10 = 10 marks)

1	Exhaustive testing is achievable for large-scale software systems.	F
2	Early testing helps reduce the cost of fixing defects.	T
3	Software defects are usually distributed evenly across all modules.	F
4	Testing can prove that a software system is completely defect-free.	F
5	Testing techniques should be selected according to the application context.	T
6	Reusing the same test cases forever guarantees the discovery of new defects.	F
7	Most defects are often found in a small number of modules.	T
8	A software product with no known defects will always satisfy users.	F
9	Testing should start only after coding is completed.	F
10	Testing demonstrates the presence of defects rather than their absence.	T

- 2)
(a) Compare *Software Static Testing* vs *Software Dynamic Testing*. (04 Marks)

ANSWER IN THIS BOX

Software Static Testing:

Software static testing is a verification technique where software work products such as requirements, design documents, and source code are reviewed and analyzed **without executing the program**. It aims to identify defects early in the development lifecycle through activities like reviews, walkthroughs, inspections, and static analysis.

Software Dynamic Testing:

Software dynamic testing is a validation technique where the software is **executed with test data** to observe its behavior and verify whether it produces the expected results. It involves running test cases and checking actual outputs against expected outputs to detect defects in the system during execution

(b) List four (04) Static Testing Techniques.

(04 Marks)

ANSWER IN THIS BOX

- Informal reviews
- Technical reviews
- Walkthrough
- Inspection
- Static Code review

(c) A software company is preparing a test environment for a newly developed online banking system before conducting system testing.

(I) What is meant by a *Test Environment* and explain its purpose in software testing.
(02 Marks)

ANSWER IN THIS BOX

- | |
|--|
| ☐ Combination of hardware, software, and network configurations used to execute tests (1) |
| ☐ Supports and enables test execution (1) |
| |
| |
| |

- (II) Describe three (03) activities that have to be performed when designing and setting up a Test Environment. (03 Marks)

ANSWER IN THIS BOX

- | |
|--|
| ☐ Design test environment |
| ☐ Verify network configuration |
| ☐ Identify server OS, databases, and components |
| ☐ Determine license requirements |
| ☐ Analyze setup requirements |
| ☐ Prepare hardware/software requirements |
| ☐ Obtain official setup confirmation |
| |

- (d) A smart home system consists of multiple sensors (temperature, motion, and smoke sensors), a mobile application, and cloud services that continuously exchange data to monitor and control household devices.

- (I) Explain the importance of *Interoperability Testing* for this smart home system. (04 Marks)

ANSWER IN THIS BOX

Interoperability Testing is performed to verify that different software and hardware components can communicate, exchange data, and work together correctly according to specified requirements.

In a smart home system, sensors, mobile applications, and cloud services are developed using different technologies and may use different communication methods.

Interoperability Testing ensures that:

Data generated by sensors is accurately transmitted to the cloud.

The cloud service correctly processes and stores the received data.

The mobile application can retrieve and display information correctly.

Commands sent from the mobile application are correctly executed by smart devices.

The entire end-to-end system functions reliably without communication failures.

Without Interoperability Testing, users may receive incorrect information, experience delayed responses, or encounter failures in controlling smart devices.

- (II) Describe any two (02) types of Interoperability Testing that should be performed on this system. For each type, provide a suitable example related to the above scenario. (08 marks)

ANSWER IN THIS BOX

1. Data Type Interoperability Testing

This testing verifies whether different systems correctly handle and interpret the data types exchanged between them.

Example:

A temperature sensor sends a reading of **25.5°C** as a floating-point number. The cloud service and mobile application must correctly process and display this value without converting it incorrectly.

2. Semantic Interoperability Testing

This testing ensures that all communicating systems interpret the meaning of exchanged data in the same way.

Example:

If a motion sensor sends a status value of "**Detected**", the cloud service and mobile application must understand it as motion being detected and not interpret it differently.

3. Physical Interoperability Testing

This testing verifies that hardware devices can physically connect and communicate with each other through the required communication interfaces.

Example:

Smart sensors, gateways, routers, and controllers should be able to connect through Wi-Fi, Bluetooth, Zigbee, or other supported communication technologies without connectivity issues.

4. Protocol Interoperability Testing

This testing ensures that systems using communication protocols can exchange messages correctly and follow the same protocol standards.

Example:

Temperature sensors may communicate with the cloud using **MQTT**, while the mobile application accesses data through **HTTPS**. Testing verifies that messages are transmitted and received correctly according to protocol specifications.

5. Data Format Interoperability Testing

This testing verifies that systems can correctly exchange and interpret data formats.

Example:

Sensor data may be transmitted in **JSON** format. The cloud service and mobile application must correctly parse the JSON structure and display the information accurately.

3)

- (a) Write down whether each of the following statements is true (T) or false (F) with regard to XML-Based Test Automation. (1x10 = 10 marks)

1	XPath can be used to locate dynamic elements in a web page.	T
2	ClassName locator identifies elements using the HTML tag name.	F
3	Link Text locator is used to find hyperlinks based on visible text.	T
4	CSS Selector cannot locate elements without ID, class, or name attributes.	F
5	XPath is only used for locating elements with static attributes.	F
6	CSS Selector is an alternative method to XPath for locating web elements.	T
7	ID locator is used to find elements based on the element's unique ID attribute.	T
8	Name locator is used to find elements using the name attribute.	T
9	XPath can traverse between different elements in the DOM structure.	T
10	Locators are used in Selenium to find web elements on a web page.	T

- (b) Sample XML structure for inventory records is given below

```
<database>
  <inventory>
    <product code="P101" category="Electronics">Samsung TV</product>
    <product code="P102" category="Furniture">Wooden Chair</product>
    <product code="P103" category="Electronics">Sony Speaker</product>
    <product code="P104" category="Clothing">Cotton Shirt</product>
    <product code="P105" category="Electronics">Apple Laptop</product>
    <product code="P106" category="Furniture">Steel Table</product>
  </inventory>
</database>
```


- (I) Write the **Absolute XPath** to get the product with code "**P103**". (03 marks)

ANSWER IN THIS BOX

/database/inventory/product[@code='P103']

- (II) Write the **Relative XPath** to get all products that belong to the "**Electronics**" category.

(03 marks)

ANSWER IN THIS BOX

//product[@category='Electronics']

- (III) Write the **Relative XPath** to get all products that have the word "**Sony**" in their name.

(03 marks)

ANSWER IN THIS BOX

//product[contains(text(),'Sony')]

- (c) (I) Explain the purpose of a *Test Report* in software testing. (02 marks)

ANSWER IN THIS BOX

A test report is a document which contains a summary of all test activities and final test results of a testing project.

A Test Report is a **formal document** prepared during the software testing process that communicates the current status of testing to management and key stakeholders

- (II) Describe four (04) key information that should be included in a Test Report. (04 marks)

ANSWER IN THIS BOX

- Summary of testing activities and results
- Completed deliverables (during period)
- New risks and issues
- Schedule, effort, and budget status
- Test status
- Key risks and issues
- Milestone status
- Activities and deliverables planned (next period)

(04)

- (a) Briefly explain *Continuous Integration (CI)* and *Continuous Delivery (CD)*. (04 marks)

ANSWER IN THIS BOX

Continuous Integration:

Continuous Integration (CI) is a method of continuously verifying the condition of a codebase through automated testing.

It is best achieved by integrating CI with version control.

Continuous Delivery:

Approach to regularly deploying artifacts that successfully pass the CI phase to ensure confident around the deployment.

[Note: Similar answer will be considered as correct]

- (b) *Version Control* is one of the most important practices in software development.

- (I) Briefly explain why version control is needed in Software Quality Assurance with a suitable example. (04 Marks)

ANSWER IN THIS BOX

Version control is needed in Software Quality Assurance (SQA) to manage and track changes made to software code, test cases, and test scripts throughout the development process. It ensures that all changes are recorded, allows multiple testers and developers to work together without conflict, and enables rollback to previous versions if a defect is introduced.

Example:

In a project using Git, a tester updates a Selenium test script to include a new login test case. If this update causes errors in other tests, the team can use version control to revert back to the previous working version of the script.

[Note: Similar answer will be considered as correct]

- (II) List down (04) Features of Version Control Systems (VCS).
(02 Marks)

ANSWER IN THIS BOX

- Back-up methodology
- Record increments - know which version is live
- Initiating change history - when features were added or amended
- Allow multiple developers (in remote locations) to work on same code base
- Merge changes across same files - handle collisions
- Answers who did what
- Point in time marking aka. Tagging
- Branching - release versions maintained & main development can continue

- (III) You are working on a Git repository called "*shopproject*" that contains the following files: "*home.html*", "*product.css*", and "*cart.js*". Write the appropriate Git commands for each of the following tasks.
(05 Marks)

	Task	Git Command
1	Initialize a new Git repository called "shopproject"	git init shopproject
2	Add all the files to the staging area	git add .

3 Commit the files with the message "initial shop setup"

```
git commit -m "initial shop setup"
```

4 Create a new branch called "payment-feature"

```
git branch payment-feature
```

5 Push the committed files to the remote repository

```
git push origin main
```

- (c) Mobile applications can be categorized into different types based on how they are built and accessed. Compare Native mobile applications and Browser-based mobile applications in terms of platform, installation, connectivity, and feature.

(04 Marks)

ANSWER IN THIS BOX

Native app is an app developed for a particular mobile device or platform (Android, iOS, Windows).

Can run in both online mode as well as in offline mode. Tied to the mobile OS it has been developed so, hence can't run on other OS. —02 marks

Browser-based apps are accessed over a mobile browser. Served from a server, not stored offline anywhere on the device. Has a common code base. Might lack in native specific features like GPS, Camera, etc. —02 marks

- (d) Examine the following TestNG program carefully and predict the output by filling in the given box. Apply your understanding of how TestNG annotations are executed in sequence.

(06 Marks)

```
import org.testng.annotations.*;

public class IT6206 {

    @Test
        public void testCase1() { System.out.println("Test1");}

    @BeforeMethod
        public void beforeMethod() {System.out.println("Before");}

    @AfterMethod
        public void afterMethod() { System.out.println("After"); }

    @BeforeTest
        public void beforeTest() {System.out.println("Start");}

    @AfterTest
        public void afterTest() { System.out.println("End"); }

}
```

Output

Start
Before
Test 1
After
End
