

**Cs611 current final term 2024 short and long question with answer by
Izhar Ullah mng01 swat campus**

Short question;

Q1. write any three software testing activities.

Ans: There are three main activities in software testing

1. Test Planning & Preparation
2. Test Execution
3. Analysis & Follow-up (page 66)

Q.2 Do you think code is main artifact of the testing. How testing is performed.

Ans: The primary software artifact to be tested is the software program or code written in different programming language. Testing is usually performed by writing and executing test cases which are derived from use cases. Test case is document that describes the testing activities, pre and post condition, expected system behaviour and result for a specific test. (Page 66).

Q.3 Is software requirements is different from requirement in other domains. Give reason.

Ans: Software Requirements vs. Requirements Software requirements are not similar to the requirement in other domains. Software requirements comes with a lot of variation and changes (in all stages of SDLC) as compared to other domains where requirements are completely understood and defined in the beginning of the project and they remain same throughout. (Page 45)

Q.4. How functional requirement can extract from customer statement.

Ans: To extract functional requirement from customer statements in software quality engineering, identify key functionalities or features needed to fulfill customer needs and document them clearly using techniques like cases or user stories. (ChatGPT)

Q.5. Write three impacts of Unit testing.

Following are three impacts of Unit testing.

1. Early Defect Detection
2. Efficient Debugging
3. Cost-Effective Defect Fixing. (Page 78)

Long Question 5 marks:

Q.1. Write five major software testing activities.

Ans:

1. Test planning
2. Test implementation
3. Test output analysis and follow-up activates
4. Test Execution
5. Test monitoring and control (page 66)

Q.2. What is software release management. what is important in software release management.

Ans: Release Management Release management includes the planning, scheduling and execution of the agreed features of the release. The deadlines and timelines are shared with the clients as well. The most important part is the approvals that must be seek on schedule and features to be added in the release. Release management is another name of controlling a software build throughout different stages. (Page 52)

Q.3. What should be verified and ensured during software regression testing? Explain with Example.

Ans: . Regression Testing This test is performed to verify that with the addition of new feature the previous developed and tested system still perform correctly. The complete software development project includes maintenance, defect fixing, optimizations, error correction and that result in continuous update in the product. The goal is to make sure that with the addition or change in one part does not impact other parts and the newly added features are fully integrated with the already developed system. The regression testing verifies that the goal is achieved or not. (Page 80).

AA. Example 1: Regression Testing

Let's consider development of a product and the following features were developed in the release 1.0 and 1.1.

Feature No.	Release 1.0	Release 1.1
1	Login	Login
2	Product Listing	Product Listing
3	Shopping Cart	Shopping Cart
4	Payment via Credit Card Only	Payment with Discount Card using PayPal also
5	---	Discount Card

In release 1.1 two features were added or updated (discount card and payment option with discount card or PayPal) and that should not impact the already developed features.

Q.4. Define following from “Use Case” template.

1. Use Case ID

2. Actor

Ans: **Use Case ID:** A unique identifier assigned to each specific use case within a system or project. It helps in organizing and referencing use cases during the development use cases during the development process. Typically, it consists an alphanumeric code or number. (ChatGPT).

Actor: An actor represents a role played by an external entity (such as a user or another system) that interacts with the system being described.

Actors initiate use cases and receive outputs from system. They are not part of the system itself but interact with it to achieve certain goals. (ChatGPT).

Second paper current 2024 final term

Q.1. Attributes Of software requirement

Ans; Attributes of Software Requirements There are multiple attributes of software requirements and they are explained with examples in the following headings.

1. Correct
2. Coherent
3. Complete
4. Feasible
5. Necessary
6. Verifiable
7. Traceable

Q.2. Do you agree Requirements engineering us Technical?

Ans: Yes, requirement engineering is technical because it involves gathering, analyzing, documenting, and managing the requirements for a software system, which often requires technical knowledge and skill.

Q.3. Activity Of Test Case?

1. Test Planning & Preparation
2. Test Execution
3. Analysis & Follow-up

Q.4. Define Reviews in testing? Also, its goals.

Ans: Reviews in testing involve systematically examining software artifacts to find defects and improve quality, aiming to enhance communication, meet requirement, identify risks, and share knowledge among team members.

Q.5. SPI formula?

Ans

- If the SPI is greater than 1, it means the project is ahead of schedule
- If the SPI is less than 1, it means the project is behind schedule.
- If the SPI is equal to one, it means the project is on time.

Q.6. Defect lifecycle

Ans: Defect Lifecycle A defect lifecycle consists of all the necessary steps to be carried out from defect identification or reporting to defect resolution. All the reported defects are to be fixed, re-verified and formally closed by the QA team. All the defects are tracked by assigning different statuses to them. Like as mentioned above, sometime it is not possible to remove defects, so status like Assigned, In-Progress, Fixed, NotAFix, Pending, Not Reproducible and Closed are usually used to track defect.

Q.7 Defect Tracking

Defect tracking is a methodology to track the discovered and reported defects and it ensures that the defects are being monitored and controlled throughout different stages of defect resolution. This is done by using templates or tools to ensure that tracking is being done properly. Today's modern software product are complex and may contain huge numbers of defects with high level of severity and criticality and this makes defect logging and tracking very important part of software development.

.Q.8 Black Box Testing

Functional testing and black box testing are two terms used interchangeably. Functional testing is usually achieved by performing black box testing. In the most simplest for, black box testing is an ad-hoc running and execution of the software system to observe any difference between expected and actual performance and in case of gap or variance from the desired results this process is repeated to eliminate the possibility of hardware failure causing the difference. Page | 68 Specification checklist is a good way to perform this testing. Specification checklist describes the functionality and features of the product with sample input and expected output data. Smoke Tests are most abstract level of testing that aims at ensuring that the most important functions work. H. Example 1: Black Box Testing Consider the following feature of a software product to perform arithmetic operation on numbers: This software takes some input and gives some output and what's going on inside is unknown. Some of the test ideas would be something like: ● To explore supported values, perform test for floating point number, integers, and boundaries of integers. ● In case of division, Divide by 0 would be a classic error. ● Use mouse, keyboard, type and paste to explore the possibility of entering values.

.Q.9 What is Configuration Management?

In software engineering, configuration management is the collection of the best practices of tracking and controlling changes in the software development. Multiple teams and resources work on different parts of the software at same time and their work need constant changes. So configuration management team helps them in keeping record of different versions of their work and determining what was changes and who changed it and make them able to revisit the changes if needed. Configuration management is closely associated with QA as QA team receives shipment of software code from configuration management team and vice versa.

Q.10: Feature Level Testing

- Registration of User
- Searching of Rooms in Hotel
- Selecting a particular Room in a Hotel
- Filling – in payment details
- Verification of Payment Details
- Generation of Unique Booking Id
- Sending Booking ID to User via Email

Each feature can be developed as one or multiple units and in feature level testing all features of the product is to be tested. Feature level testing require integration testing to test feature if there are multiple units.

Q.11.Post Production Issues

Post Production issues are reported be end-users once Staging Environment is Productionize, productionalizing an environment mean to deploy staging server codebase to Production Server which is provided be client or client has some sort of arrangement with Vendor for deployment. Usually, good QA will ensure that there will no critically bug on Production side but even good QA will not ensure 100% bug free code that's just possible, production issues are those issues which are reported be end-users can be of low priority from development but from end-user viewpoint they might take high-priority which are needed to be addressed.

Q.12.Patch

Patch as opposite to Hot Fix are those issues which are critical but client can wait for short period of time to get them fixed. Difference between Patch and Release is that for timeframe for Patch to deliver is usually less than Release but more than Patch and dev team do proper estimation which include QA cycles also and then share deadline. Usually, leftover bugs or WishLists or both are delivered as patch after Major Release.

Q.13.Stakeholders in Defect Lifecycle

The most important factor in defect resolution is that the decision regarding any defect must be accepted by all stakeholders. For example, if it's decided that certain defect is not to be fixed in current release, then all stakeholders must agree to that. Usually minor defects with very less impact are deferred to the next release with mutual consent and even in some cases the reported one also but can be reclassified as 'not a bug' if the situation demands so. The key point is that all the decision regarding defects must be made with mutual consent of all stakeholders.

Q.14.Software Maintenance and Defect Handling

As mentioned above, project and product based defect handling is different. Same is the case with support to be provided after deployment of the product. Software maintenance defines the scope of the support to be provided with all relevant terms and conditions. There are following levels of software support:

Q.15.Regression Testing

This test is performed to verify that with the addition of new feature the previous developed and tested system still perform correctly. The complete software development project includes maintenance, defect fixing, optimizations, error correction and that result in continuous update in the product. The goal is to make sure that with the addition or change in one part does not impact other parts and the newly added features are fully integrated with the already developed system. The regression testing verifies that the goal is achieved or not.