Q: 1

- **What is Software Testing?**
  - Explain what software testing is and why it is necessary in software development.
- **Why is early detection of bugs important in software testing?**
  - Discuss how early bug detection can benefit the development process.
- **Explain the key principles of software testing and their importance.**
- **What is the difference between manual testing and automated testing?**
- **What is the difference between black box, white box and grey box testing?**

What is Software Testing?

Software Testing is the process of evaluating the software application to ensure it meets the specified requirements and is free of defect.

Explain What software testing is and why it is necessary in software development?

Software testing is the process of evaluating the software application to ensure it meets the specified requirement and is free of defect. It helps improving customer satisfaction, making features stay updated. Software development ensures quality of the product which is achieved by software testing like functional testing(what the system does), non-functional testing(how the system performs under certain conditions like load, stress, security, etc under expected and unexpected conditions).

Why is early detection of bugs important in software testing?

Early detection of bugs is important because of the following reasons:

1. Reduces cost.
2. Time saving and fast.
3. Reduces complexity and build team confidence.

Discuss how early bug detection can benefit the development process.

Benefits of early bug detection costs less and saves time. Like

1. Misunderstanding between team: If the Business Analyst(BA) wants “user to login by email address only” but the team adds additional input for username. For this case need to ask Business Analyst permission to add username field for more clarity.
2. Presence of an inexperienced candidate: Who takes lots of time to complete the task. For this case candidate has to learn skills in before entering into project team.

3. Calculator Operator: After creating a calculator gadget/hardware, it is found ‘+’/addition operator works like ‘-’ or subtraction operation and vice versa. To fix it, need focused work during development.

Explain the key principles of software testing and their importance.

There are 7 Key principles of software testing:

1. Testing shows presence of defect.
2. Exhaustive testing is impossible.
3. Early detection saves time and money.
4. Defect clustering.
5. Pesticide paradox.
6. Testing is context-dependent.
7. Absence of Error Fallacy.

Testing shows presence of defect:

It’s not always the software is 100% defect free. In real world scenarios, even if login features works 1000 times perfectly, but can’t be sure it will perform in all possible cases in non-functional testing(load testing, stress testing, security testing etc).

Exhaustive testing is impossible:

In a calculator, having two numbers and single ‘+’ operator can have infinite possible combinations or big sequence(1 + 2 + 2 + 1 + …). Doing all possible ways is impossible. To fix this there are several approaches, using black box techniques like boundary value analysis, equivalence partitioning, state transition testing, decision table testing.

Early detection saves time and money:

Detecting the bug in the early stages cost less as compared to finding bugs in the Acceptance testing. Suppose the login module does not need OTP feature, if it is known in the beginning it reduces cost, it happens due to several reasons, like misunderstanding between BA and testers/developers etc.

Defect clustering:

Doing testing at areas where more bugs/defects are likely to be found using the Pareto principle(80/20 rule). Find the 20% of module where there might be 80% of bugs/defects. If a team lacks some skills then there might be more bugs/defects on those areas.

Pesticide paradox:

Avoid use of same test cases which does not improve finding the new defects/bugs. To fix this add reviewing process to check unnecessary test cases, including test cases which are also sub part of other test cases. Making user get logged in which also include checking title and url are as expected or not. So no need to create other test cases to check title and url.

Testing is context-dependent:

Software application must meet its corresponding field of interaction with the end user/client meaning the Banking application must not look like gaming application with make cause addition to unwanted doings/distractions.

Absence of Error Fallacy:

Even if the software is defect/bug free, if it does not fallow customer requirements is still a failure. Banking app can transfer money through Scanner. To fix this give a feedback, or use other app if not fixed as per customer patience.

What is the difference between manual testing and automation testing?

Manual testing:
In this testing process, tester does not uses any automation tools to fallow test cases instead testers manually uses his observation, domain knowledge, exploration over software either to test in form of functional and non-functional way to find defect/bug.

Automation testing:
In this testing process, tester uses automation tools like cypress, selenium, playwright etc to test software applications based on test cases/customer requirements, making human work effortless and reduces repeated tasks, reduces human unnecessary human observations.

Manual testing	Automation testing
Requirement of Human effort, like observation, skills, domain knowledge, etc.	Requirement of Automation tools like cypress, selenium, playwright, etc.
Possibility of human errors like misunderstanding, etc.	Error done by tools are less under high maintenance of devices, application, internet speed, etc.
Important when human instinct, exploration to find defect are necessary which can't be done by automation tools which might take long code design.	Used to find bugs/defects under low human instinct, less creativity to find bugs, during repeating task over different credentials, etc.

What is the difference between black box, white box and grey box testing?

Black box Testing	White box Testing	Grey box Testing
Testing software application without knowing internal code structure, implementation details and internal paths.	Testing the software application on internal code structure, etc.	Testing the software application with having partial knowledge on internal and external code structure, implementation details, etc. Its the combination of black box and white box testing.
It focuses on what the software does.	It primarily focuses on how the system does.	It focuses on partially both on what and how the system does.

Performed by testers.	Performed by developers.	Performed by developers/testers.
Example: Testing login feature by providing valid credentials like username, password, click on login button.	Example: Testing functions, methods and classes working properly.	Example: If tester knows additional developer skills, then tester does the developers process.