**Q. What is a software development life cycle?**

The software development life cycle, also known as the SDLC, describes the process through which software is developed. There are slight differences in what different organizations call these specific steps, but they all start with development and end with application maintenance. The most popularly accepted steps are: requirement phase, the design phase, the coding phase, the testing phase, release, and maintenance.

**Q. What is positive testing? Negative testing? Compare the two.**

When you test an application with the correct data, ensuring that everything works as it should with expected user behavior, you are performing a positive test. For example, if a password only allows numbers, you would only put numbers into this field with a positive test. If everything works as it should, your program has passed this particular positive test.

When you test an application with incorrect data, ensuring that errors don’t occur with unexpected user behavior, you are performing a negative test. Using the previous example, you would put both numbers and letters into the password field with a negative test. If an error message occurs, you know that your program has passed this particular negative test; it didn’t let you into the system when there was unexpected behavior.

Negative testing ensures that everything works when there is unexpected user behavior, while positive testing ensures that everything works when the end user does what the program expects. In other words, negative testing makes sure that things that shouldn’t work, don’t, while positive testing makes sure that things that should work, do.

**Q. What is a primary key? What is a unique key? How are they different?**

A primary key is a column in a database where each row has a unique value. Each table has only one primary key. No NULL values are allowed. A unique key is a column or group of columns that together hold unique values.  A table can have more than one unique key.  For example, in a list of American Citizens, the column with social security numbers would be a primary key whereas the first and last name columns combined with phone number would be a unique key.

**Q. What is the difference between requirements and specifications?**

Requirements are the features, functions, and goals of the proposed software system as defined by the client. For example, a company may want their software to “Store shopping cart data for at least 30 days.” This would be a requirement.

Specifications, on the other hand, explain how these features, functions, and goals are to be met. For example, a specification would be “A user’s session information will be persisted into the operational datastore upon logout or session timeout and maintained for 30 days.”

Q. **When do we perform functional testing?**

Functional testing tests the code from end-to-end, making sure that all parts of the application are working –- even the parts that occur during failures. Functional testing should start early. Planning can be initiated during the analysis phase, and functional testing should start well before development is complete. It is common to start running functional tests within the QA department once a Minimum Viable Product is released.

**Q. What is the sequence to write a test case?**

A test case should have these four things:

1.    Test case name

2.    Description of the test case

3.    Steps, description of the actions that need to be performed

4.    Expected Results, describe expected outcomes so as to compare

results

**Q. What is application programming interface?**

An application programming interface, also known as an API, is a programming language that allows different applications to communicate with one another. Companies often release APIs so that other organizations can build products that are based off of this original software. An example of an API would be Java’s JDBC (Java Data Base Connectivity).

**Q. What if there isn’t enough time for thorough testing? What would you do?**

In order to understand where to focus testing efforts during a time crunch, the project team must perform risk analysis and speak with stakeholders. Primary considerations include, but are not limited to, understanding important functionality, especially those apparent to the end user, and making sure that the software is capable of meeting it’s initial specifications.

**Q. What is the difference between performance testing and load testing?**

When you want to see what happens to your system with a certain user load, you are load testing. With this form of testing, you have a better understanding of what is happening from the users’ perspective, whereas performance testing allows you to see what is happening from an architecture perspective. Performance testing is concerned with an application’s efficiency not user load.

|  |
| --- |
| **Q. What is difference between QA, QC and Software Testing?**  Quality Assurance (QA): QA refers to the planned and systematic way of monitoring the quality of process which is followed to produce a quality product. QA tracks the outcomes and adjusts the process to meet the expectation.  Quality Control (QC): Concern with the quality of the product. QC finds the defects and suggests improvements. The process set by QA is implemented by QC. The QC is the responsibility of the tester.  Software Testing: is the process of ensuring that product which is developed by the developer meets the user requirement. The motive to perform testing is to find the bugs and make sure that they get fixed. |
| **Q. When to start QA in a project?**  A good time to start the QA is from the beginning of the project startup. This will lead to plan the process which will make sure that product coming out meets the customer quality expectation. QA also plays a major role in the communication between teams. It gives time to step up the testing environment. The testing phase starts after the test plans are written, reviewed and approved. |
| **Q. What are verification and validation and difference between these two?**   * Verification: process of evaluating steps which is followed up to development phase to determine whether they meet the specified requirements for that stage.  Validation: process of evaluating product during or at the end of the development process to determine whether product meets specified requirements.  Difference between Verification and Validation: * - Verification is Static Testing where as Validations is Dynamic Testing. * - Verification takes place before validation. * - Verification evaluates plans, documents, requirements and specifications, where as Validation evaluates product. * - Verification inputs are checklist, issues list, walkthroughs and inspection, where as in Validation testing of actual product. * - Verification output is set of documents, plans, specifications and requirement documents where as in Validation actual product is output. |
| **Q. What is difference between Smoke testing and Sanity Testing?**  The difference between smoke and sanity testing is described below:  - Sanity testing is performed when new build is released after fixing bugs where as smoke testing is performed to check the major functionalities of the application. - Sanity is performed by the tester or the developer but smoke testing can be performed by the tester or developer. - Smoke testing is performed earlier where as sanity is performed after the smoke testing. - Sanity testing is narrow and deep approach of testing and smoke testing is focused testing based on major functionalities. |