Quality assurance questions and answers

What is Software Quality Assurance?
Software QA involves the entire software development PROCESS - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to “prevention”. In other words it is:

- The planned systematic activities necessary to ensure that a component, module, or system conforms to established technical requirements.
- All actions that are taken to ensure that a development organization delivers products that meet performance requirements and adhere to standards and procedures.
- The policy, procedures, and systematic actions established in an enterprise for the purpose of providing and maintaining some degree of confidence in data integrity and accuracy throughout the life cycle of the data, which includes input, update, manipulation, and output.
- The actions, planned and performed, to provide confidence that all systems and components that influence the quality of the product are working as expected individually and collectively.

What is Test schedule?
The test schedule is a schedule that identifies all tasks required for a successful testing effort, a schedule of all test activities and resource requirements.

Explain Black-box Testing?
1. Functional testing based on requirements with no knowledge of the internal program structure or data.
2. Black box testing indicates whether or not a program meets required specifications by spotting faults of omission - places where the specification is not fulfilled.
3. Black-box testing relies on the specification of the system or the component that is being tested to derive test cases. The system is a black-box whose behavior can only be determined by studying its inputs and the related outputs
4. Black box testing is functional testing, not based on any knowledge of internal software design or code. Black box testing based on requirements and functionality.

What Is Grey box testing?
Grey box testing is a software testing technique that uses a combination of black box testing and white box testing. Gray box testing is not black box testing, because the tester does know some of the internal workings of the software under test. In grey box testing, the tester applies a limited number of test cases to the internal workings of the software under test. In the remaining part of the grey box testing, one takes a black box approach in applying inputs to the software under test and observing the outputs.

What is Manual Testing?
That part of software testing that requires operator input, analysis, or evaluation. Or a manual test is a test for which there is no automation. Instead, test steps are outlined in a document for the tester to complete. The tester can then report test results and submit bugs as appropriate.

What is Alpha Testing?
1. Acceptance testing performed by the customer in a controlled environment at the developer's site. The software is used by the customer in a setting approximating the target environment with the developer observing and recording errors and usage problems.
Testing of a software product or system conducted at the developer's site by the end user.

2. Alpha testing is testing of an application when development is nearing completion. Minor design changes can still be made as a result of alpha testing. Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers.
3. Alpha testing is final testing before the software is released to the general public. First, (and this is called the first phase of alpha testing), the software is tested by in-house developers. They use either debugger software, or hardware-assisted debuggers. The goal is to catch bugs quickly. Then, (and this is called second stage of alpha testing), the software is handed over to us, the software QA staff, for additional testing in an environment that is similar to the intended use.

What is Incremental testing?
Incremental testing is partial testing of an incomplete product. The goal of incremental testing is to provide an early feedback to software developers.

What is Interface Testing?
Testing conducted to evaluate whether systems or components pass data and control correctly to one another. Contrast with testing, unit; testing, system.

Explain Beta Testing?
Acceptance testing performed by the customer in a live application of the software in an environment which is not controlled by the developer.
Testing conducted at one or more end user sites by the end user of a delivered software product or system.
Beta testing is testing an application when development and testing are essentially completed and final bugs and problems need to be found before the final release. Beta testing is typically performed by end-users or others, not programmers, software engineers, or test engineers.
Following alpha testing, "beta versions" of the software are released to a group of people, and limited public tests are performed, so that further testing can ensure the product has few bugs. Other times, beta versions are made available to the general public, in order to receive as much feedback as possible. The goal is to benefit the maximum number of future users.

What is The Mission of Testing?
In well-run projects, the mission of the test team is not merely to perform testing, but to help minimize the risk of product failure. Testers look for manifest problems in the product, potential problems, and the absence of problems. They explore, assess, track, and report product quality, so that others in the project can make informed decisions about product development. It's important to recognize that testers are not out to "break the code".

Explain Software Testing Strategies?
A strategy for software testing integrates software test case design techniques into a well - planned series of steps that result in the successful construction of software.
Common Characteristics of Software Testing Strategies
-Testing begins at module level and works outward towards the integration of the entire system.
-Different testing techniques are appropriate at different points in time.
-Testing is conducted by the developer of the software and for large projects by an independent test group.
-Testing and debugging are different activities, but debugging must be accommodated in any testing strategy

Explain Unit testing?
Unit testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable / acceptable.

Explain Compatibility testing?
Compatibility testing is testing how well software performs in a particular hardware, software, operating system, or network environment.

Explain Integration Testing?
1. An orderly progression of testing in which software elements, hardware elements, or both are combined and tested, to evaluate their interactions, until the entire system has been integrated.
2. Integration testing is black box testing. The purpose of integration testing is to ensure distinct components of the application still work in accordance to customer requirements. Test cases are developed with the express purpose of exercising the interfaces between the components. This activity is carried out by the test team. Integration testing is considered complete, when actual results and expected results are either in line or differences are explainable/acceptable based on client input.

Explain Incremental integration testing?
Incremental integration testing is continuous testing of an application as new functionality is recommended. This may require that various aspects of an application's functionality are independent enough to work separately, before all parts of the program are completed, or that test drivers are developed as needed. Incremental testing may be performed by programmers, software engineers, or test engineers.

Explain Performance Testing?
1. Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements.
Although performance testing is described as a part of system testing, it can be regarded as a distinct level of testing. Performance testing verifies loads, volumes and response times, as defined by requirements.
2. Performance testing verifies loads, volumes, and response times, as defined by requirements.
Although performance testing is a part of system testing, it can be regarded as a distinct level of testing.

Explain Stress Testing?
1. Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements.
2. Stress testing is testing that investigates the behavior of software (and hardware) under extraordinary operating conditions. For example, when a web server is stress tested, testing aims to find out how many users can be on-line, at the same time, without crashing the server. Stress testing tests the stability of a given system or entity. It tests something beyond its normal operational capacity, in order to observe any negative results. For example, a web server is stress tested, using scripts, bots, and various denials of service tools.

Explain Regression Testing?
1. Rerunning test cases which a program has previously executed correctly in order to detect errors spawned by changes or corrections made during software development and maintenance.
2. The objective of regression testing is to ensure the software remains intact. A baseline set of data and scripts is maintained and executed to verify changes introduced during the release have not "undone" any previous code. Expected results from the baseline are compared to results of the software under test. All discrepancies are highlighted and accounted for, before testing proceeds to the next level.

Explain System Testing?
Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system. All the software components, all the hardware components and any interfaces.

Explain Path Testing?
Testing to satisfy coverage criteria that each logical path through the program be tested. Often paths through the program are grouped into a finite set of classes. One path from each class is then tested.

Explain Installation testing?
Installation testing is testing full, partial, upgrade, or install/uninstall processes. The installation test for a release is conducted with the objective of demonstrating production readiness.
This test includes the inventory of configuration items, performed by the application's System Administration, the evaluation of data readiness, and dynamic tests focused on basic system functionality. When necessary, a sanity test is performed, following installation testing.

Explain Smoke testing?
Smoke testing is a relatively simple check to see whether the product "smokes" when it runs. Smoke testing is also known as ad hoc testing, i.e. testing without a formal test plan.
With many projects, smoke testing is carried out in addition to formal testing. If smoke testing is carried out by a skilled tester, it can often find problems that are not caught during regular testing. Sometimes, if testing occurs very early or very late in the software development life cycle, this can be the only kind of testing that can be performed.
Smoke testing minimizes integration risk, reduces the risk of low quality, supports easier defect diagnosis, and improves morale. Smoke testing does not have to be exhaustive, but should expose any major problems. Smoke testing should be thorough enough that, if it passes, the tester can assume the product is stable enough to be tested more thoroughly. Without smoke testing, the daily build is just a time wasting exercise. Smoke testing is the sentry that guards against any errors in development and future problems during integration. At first, smoke testing might be the testing of something that is easy to test. Then, as the system grows, smoke testing should expand and grow, from a few seconds to 30 minutes or more.

Explain Client?
The end user that pays for the product received, and receives the benefit from the use of the product.

What is Integration?
The process of combining software components or hardware components, or both, into an overall system.

Explain Debugging?
The act of attempting to determine the cause of the symptoms of malfunctions detected by testing or by frenzied user complaints.

What is Defect Density?
Ratio of the number of defects to program length (a relative number).

Explain Error?
A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

What is Failure?
The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered.

What is Fault?
A manifestation of an error in software. A fault, if encountered, may cause a failure.

Explain Defect?
From the producer's viewpoint: a product requirement that has not been met or a product attribute possessed by a product or a function performed by a product that is not in the statement of requirements that define the product.

What is Software Bug?

A fault in a program which causes the program to perform in an unintended or unanticipated manner.

Explain Inputs?
Products, services, or information needed from suppliers to make a process work.

What is Outputs?
Products, services, or information supplied to meet end user needs.

Explain Usability Testing?
Usability testing is testing for 'user-friendliness'. Clearly this is subjective and depends on the targeted end-user or customer. User interviews, surveys, video recording of user sessions and other techniques can be used. Programmers and developers are usually not appropriate as usability testers.

Explain Sanity testing?
Sanity testing is performed whenever cursory testing is sufficient to prove the application is functioning according to specifications. This level of testing is a subset of regression testing.
It normally includes a set of core tests of basic GUI functionality to demonstrate connectivity to the database, application servers, printers, etc.

What is Security penetration testing?
Security/penetration testing is testing how well the system is protected against unauthorized internal or external access, or willful damage.
This type of testing usually requires sophisticated testing techniques.

Explain Automation testing?
Automated testing is a formally specified and controlled method of formal testing approach.
Or that part of software testing that is assisted with software tool(s) that does not require operator input, analysis, or evaluation.

What is Defect Analysis?
Using defects as data for continuous quality improvement. Defect analysis generally seeks to classify defects into categories and identify possible causes in order to direct process improvement efforts.

What is Policy?
Managerial desires and intents concerning either process (intended objectives) or products (desired attributes).

What is Life Cycle?
The period that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically includes a requirements phase, design phase, implementation (code) phase, test phase, installation and checkout phase, operation and maintenance phase, and a retirement phase.

What is Test design?
Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

Explain Test driver?
A software module used to invoke a module under test and, often, provide test inputs, control and monitoi execution, and report test results.

What is Test item?
A software item which is the object of testing

What is Test phase?
The period of time in the software life cycle in which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied.

Explain Test Procedure?
The formal or informal procedure that will be followed to execute a test. This is usually a written document that allows others to execute the test with a minimum of training.

What is Test Development?
The development of anything required to conduct testing. This may include test requirements (objectives), strategies, processes, plans, software, procedures, cases, documentation, etc.

Test engineer?
We, test engineers are engineers who specialize in testing. We, test engineers, create test cases, procedures, scripts and generate data. We execute test procedures and scripts, analyze standards of measurements, and evaluate results of system/integration/regression testing. We also...
* Speed up the work of your development staff;
* Reduce your organization's risk of legal liability;
* Give you the evidence that your software is correct and operates properly;
* Improve your problem tracking and reporting;
* Maximize the value of your software;
* Maximize the value of the devices that use it;
* Assure the successful launch of your product by discovering bugs and design flaws, before users get discouraged, before shareholders lose their cool and before employees get bogged down;
* Help the work of your development staff, so the development team can devote its time to build up your product;
* Promote continual improvement;
* Provide documentation required by FDA, FAA, other regulatory agencies and your customers;
* Save money by discovering defects 'early' in the design process, before failures occur in production, or in the field;
* Save the reputation of your company by discovering bugs and design flaws; before bugs and design flaws damage the reputation of your company.

What is Test/QA Team Lead?
The Test/QA Team Lead coordinates the testing activity, communicates testing status to management and manages the test team.

What is A System Administrator?
Test Build Managers, System Administrators, Database Administrators deliver current software versions to the test environment, install the application's software and apply software patches, to both the application and the operating system, set-up, maintain and back up test environment hardware. Depending on the project, one person may wear more than one hat. For instance, a Test Engineer may also wear the hat of a System Administrator.

What is A Technical Analyst?
Technical Analysts perform test assessments and validate system/functional test requirements. Depending on the project, one person may wear more than one hat. For instance, Test Engineers may also wear the hat of a Technical Analyst.

Explain the Requirements?

Requirement specifications are important and one of the most reliable methods of insuring problems in a complex software project is to have poorly documented requirement specifications. Requirements are the details describing an application's externally perceived functionality and properties. Requirements should be clear, complete, reasonably detailed, cohesive, attainable and testable. A non-testable requirement would be, for example, "user-friendly", which is too subjective. A testable requirement would be something such as, "the product shall allow the user to enter their previously-assigned password to access the application". Care should be taken to involve all of a project's significant customers in the requirements process. Customers could be in-house or external and could include end-users, customer acceptance test engineers, testers, customer contract officers, customer management, future software maintenance engineers, salespeople and anyone who could later derail the project. If his/her expectations aren't met, they should be included as a customer, if possible. In some organizations, requirements may end up in high-level project plans, functional specification documents, design documents, or other documents at various levels of detail. No matter what they are called, some type of documentation with detailed requirements will be needed by test engineers in order to properly plan and execute tests. Without such documentation there will be no clear-cut way to determine if a software application is performing correctly.

What is a user manual?
A "user manual" is a document that presents information necessary to employ software or a system to obtain the desired results. Typically, what is described are system and component capabilities, limitations, options, permitted inputs, expected outputs, error messages, and special instructions.

Explain Interface?
A shared boundary. An interface might be a hardware component to link two devices, or it might be a portion of storage or registers accessed by two or more computer programs.

What is a user interface?
"User interface" is the interface between a human user and a computer system. It enables the passage of information between a human user and hardware or software components of a computer system.

What is a software version?
A software version is an initial release (or re-release) of a software associated with a complete compilation (or recompilation) of the software.

Explain virtual memory?
Virtual memory relates to virtual storage. In virtual storage, portions of a user's program and data are placed in auxiliary storage, and the operating system automatically swaps them in and out of main storage as needed.

Explain bug life cycle?
Bug life cycles are similar to software development life cycles. At any time during the software development life cycle errors can be made during the gathering of requirements, requirements analysis, functional design, internal design, documentation planning, document preparation, coding, unit testing, test planning, integration, testing, maintenance, updates, re-testing and phase-out.
Bug life cycle begins when a programmer, software developer, or architect makes a mistake, creates an unintentional software defect, i.e. bug, and ends when the bug is fixed, and the bug is no longer in existence.
What should be done after a bug is found? When a bug is found, it needs to be communicated and assigned to developers that can fix it. After the problem is resolved, fixes should be re-tested. Additionally, determinations should be made regarding requirements, software, hardware, safety impact, etc., for regression testing to check the fixes didn't create other problems elsewhere.
If a problem-tracking system is in place, it should encapsulate these determinations. A variety of commercial, problem-tracking, management software tools are available. These tools, with the detailed input of software test engineers, will give the team complete information so developers can understand the bug, get an idea of its severity, reproduce it and fix it.

What is software life cycle?
Software life cycle begins when a software product is first conceived and ends when it is no longer in use. It includes aspects such as initial concept, requirements analysis, functional design, internal design, documentation planning, test planning, coding, document preparation, integration, testing, maintenance, updates, retesting, phase-out, and other aspects.

Explain Test case?
1. Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.
A test case is a document that describes an input, action, or event and an expected response, to determine if a feature of an application is working correctly. A test case should contain particulars such as test case identifier, test case name, objective, test conditions/setup, input data requirements, steps, and expected results.
Note that the process of developing test cases can help find problems in the requirements or design of an application, since it requires completely thinking through the operation of the application. For this reason, it's useful to prepare test cases early in the development cycle if possible.
Or the definition of test case differs from company to company, engineer to engineer, and even project to project. A test case usually includes an identified set of information about observable states, conditions, events, and data, including inputs and expected outputs.
2. A test case is a document that describes an input, action, or event and its expected result, in order to determine if a feature of an application is working correctly. A test case should contain particulars such as a...
*Test case identifier;
*Test case name;
*Objective;
*Test conditions/setup;
* Input data requirements/steps;
* Expected results.
Please note, the process of developing test cases can help find problems in the requirements or design of an application, since it requires you to completely think through the operation of the application. For this reason, it is useful to prepare test cases early in the development cycle, if possible.

What is Test scenario?
The terms "test scenario" and "test case" are often used synonymously. Test scenarios are test cases or test scripts, and the sequence in which they are to be executed. Test scenarios are test cases that ensure that all business process flows are tested from end to end. Test scenarios are independent tests, or a series of tests that follow each other, where each of them dependent upon the output of the previous one. Test scenarios are prepared by reviewing functional requirements, and preparing logical groups of functions that can be further broken into test procedures. Test scenarios are designed to represent both typical and unusual situations that may occur in the application. Test engineers define unit test requirements and unit test scenarios. Test engineers also execute unit test scenarios. It is the test team that, with assistance of developers and clients, develops test scenarios for integration and system testing. Test scenarios are executed through the use of test procedures or scripts. Test procedures or scripts define a series of steps necessary to perform one or more test scenarios. Test procedures or scripts may cover multiple test scenarios.

What is Test documentation?
Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report.

What is Test incident report?
A document reporting on any event that occurs during testing that requires further investigation.

What is Test log?
A chronological record of all relevant details about the execution of a test.

Explain Test plan?
A software project test plan is a document that describes the objectives, scope, approach and focus of a software testing effort. The process of preparing a test plan is a useful way to think through the efforts needed to validate the acceptability of a software product. The completed document will help people outside the test group understand the why and how of product validation. It should be thorough enough to be useful, but not so thorough that none outside the test group will be able to read it

What is Test Objective?
An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation.

Test report?
A document describing the conduct and results of the testing carried out for a system or system component.

What is Process and procedures?
Detailed and well-written processes and procedures ensure the correct steps are being executed to facilitate a successful completion of a task. They also ensure a process is repeatable.

What is a user guide?
The "user guide" is the same as the user manual. The user guide is a document that presents information necessary to employ a system or component to obtain the desired results. Typically, what is described are system and component capabilities, limitations, options, permitted inputs, expected outputs, error messages, and special instructions.

What is a utility?
"Utility" is a software tool designed to perform some frequently used support function. For example, one utility is a program to print files.

What is localization?
The aspect of development and testing relating to the translation of the software and it's representation to the end user. This includes translating the program, choosing appropriate icons and graphics, and other cultural considerations. It also may include translating the program's help files and the documentation. You could think of localization as pertaining to the presentation of your program; the things the user sees.

What is Internationalization?
The aspect of development and testing relating to handling foreign text and data within a program. This would include sorting, importing and exporting text and data, correct handling of currency and date and time formats, string parsing, upper and lower case handling, and so forth. It also includes the task of separating strings (or user interface text) from the source code, and making sure that the foreign language strings have enough space in your user interface to be displayed correctly. You could think of internationalization as pertaining of the underlying functionality and workings of your program.

What we need consider in localization testing?

- String size change breaking layout and alignment. When words or sentences are translated into other languages, most of the time the resulting string will be either longer or shorter than the native language version of the string. Two solutions to this problem:
    - 1. Account the space needed for string expansion, adjusting the layout of your dialog accordingly

    - 2. Separate your dialog resources into separate dynamic libraries.
- Data format localization: European style: DD/MM/YY North American style: MM/DD/YY. Currency, time and number format, address.
- Character sets: ASCII or Non ASCII, Single byte character 16bit (US) 256 characters, Double byte character 32bit (Chinese) 65535 code points
- Encoding: Unicode: Unicode supports many different written languages in the world all in a single character encoding. Note: For double character set, it is better to convert from Unicode to UTF8 for Chinese, because UTF8 is a variable length encoding of Unicode that can be easily sent through the network via single byte streams.
- Builds and installer: Creating an environment that supports a single version of your code, and multiple versions of the language files.
- Program's installation, uninstall in the foreign machines.
- Testing with foreign characters.
- Foreign Keyboards or On-Screen keyboard
- Text filters: Program that is used to collect and manipulate data usually provides the user with a mechanism for searching and filtering that data. As a global software tester, you need to make sure that the filtering and searching capabilities of your program work correctly with foreign text. Problem: ignore the accent marks used in foreign text.
- Loading, saving, importing, and exporting high and low ASCII
- Asian text in program: how double character set work
- Two environments to test for the program in Chinese
- In Chinese window system, (in China)
- In English Window system with Chinese language support (in USA) Microsoft language codes:
    - CHS - Chinese Simplified
    - CHT - Chinese Traditional(Taiwan)
    - ENU - English (United States)
    - FRA- French (France)
- Java Languages codes
    - zh_CN - Chinese Simplified
    - zh_TW - Chinese Traditional(Taiwan)
    - Fr or fr_FR - French (France)
    - en or en_US - English (United States)
- More need consider in localization testing:
    - o Hot key.
    - o Garbled in translation
    - o Error message identifiers
    - o Hyphenation rules
    - o Spelling rules
    - o Sorting rules
    - o Uppercase and lowercase conversion

What is Ad Hoc Testing?

A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality. Can include negative testing as well.

What is Acceptance Testing?
Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

What is the maximum length of the test case we can write?
We can't say exactly test case length, it depending on functionality.

What is internationalization testing?
Software Internationalization is process of developing software products independent from cultural norms, language or other specific attributes of a market

What is Bug life cycle?
New: when tester reports a defect
Open: when developer accepts that it is a bug or if the developer rejects the defect, then the status is turned into "Rejected" Fixed: when developer make changes to the code to rectify the bug...
Closed/Reopen: when tester tests it again. If the expected result shown up, it is turned into "Closed" and if the problem persists again, it's "Reopen"

What is a Use case?
A simple flow between the end user and the system. It contains pre conditions, post conditions, normal flows and exceptions. It is done by Team Lead/Test Lead/Tester.

What are the differences between these three words Error, Defect and Bug?
Error: The deviation from the required logic, syntax or standards/ethics is called as error.
There are three types of error. They are:
Syntax error (This is due to deviation from the syntax of the language what supposed to follow).
Logical error (This is due to deviation from the logic of the program what supposed to follow)
Execution error (This is generally happens when you are executing the same program, that time you get it.)
Defect: When an error found by the test engineer (testing department) then it is called defect
Bug: if the defect is agreed by the developer then it converts into bug, which has to fix by the developer or post pond to next version.