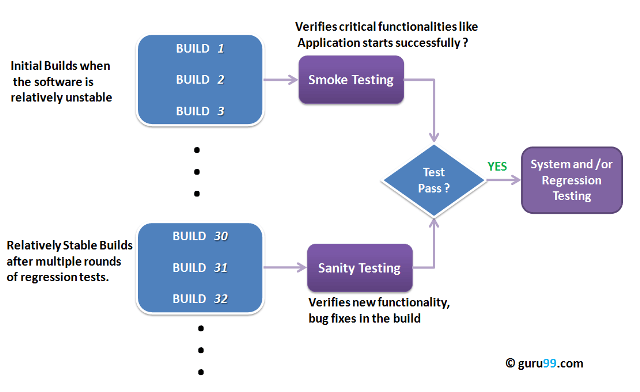
**Software testing interview questions**

1. Difference between Smoke and Sanity testing?

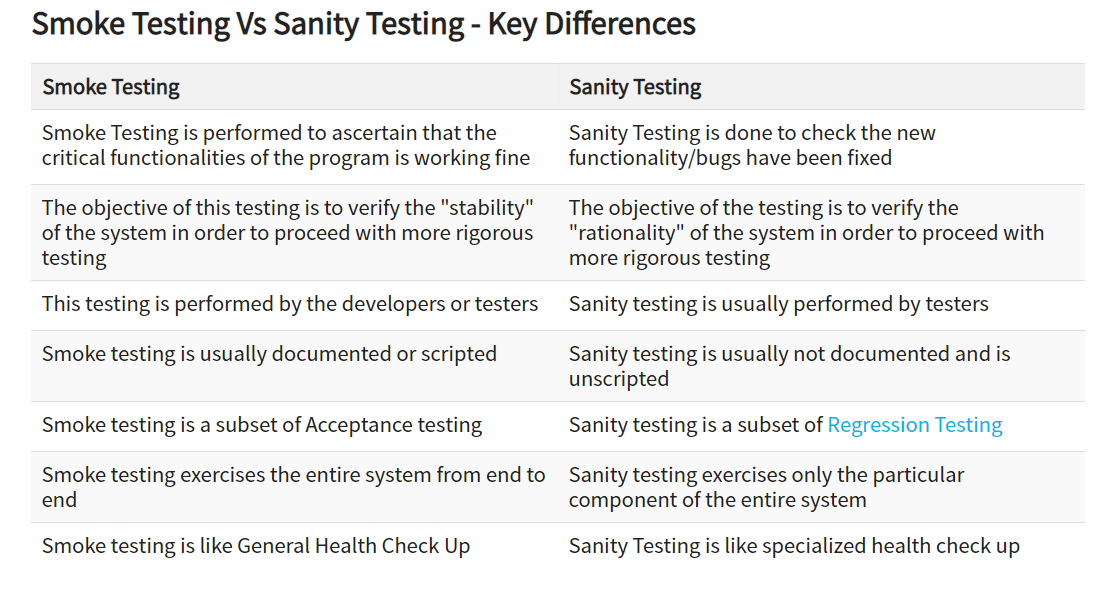
**Ans-** 

**What is Smoke Testing?** [Smoke Testing](https://www.guru99.com/smoke-testing.html) is a kind of Software Testing performed after software build to ascertain that the critical functionalities of the program are working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application so that the QA team does not waste time installing and testing the software application.

In Smoke Testing, the test cases chose to cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine.  
For Example, a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.

**What is Sanity Testing?** Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

The objective is "not" to verify thoroughly the new functionality but to determine that the developer has applied some rationality (sanity) while producing the software. For instance, if your scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.



## Points to note.

* Both sanity tests and smoke tests are ways to avoid wasting time and effort by quickly determining whether an application is too flawed to merit any rigorous testing.
* Sanity Testing is also called tester acceptance testing.
* Smoke testing performed on a particular build is also known as a build verification test.
* One of the best industry practice is to conduct a Daily build and smoke test in software projects.
* Both smoke and sanity tests can be executed manually or using an automation tool.  When automated tools are used, the tests are often initiated by the same process that generates the build itself.
* As per the needs of testing, you may have to execute both Sanity and Smoke Tests in the software build. In such cases, you will first execute Smoke tests and then go ahead with Sanity Testing. In industry, test cases for Sanity Testing are commonly combined with that for smoke tests, to speed up test execution. Hence, it's a common that the terms are often confused and used interchangeably

2**) What is Regression Testing?**

Ans- Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the new code changes are done.

## Need of Regression Testing

Regression Testing is required when there is a

* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix

## How to do Regression Testing

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary. Regression Testing can be carried out using the following techniques:

[](https://www.guru99.com/images/regressiontestingtypes.png)

### Retest All

* This is one of the methods for Regression Testing in which all the tests in the existing test bucket or suite should be re-executed. This is very expensive as it requires huge time and resources.

### Regression Test Selection

* Instead of re-executing the entire test suite, it is better to select part of the test suite to be run
* Test cases selected can be categorized as 1) Reusable Test Cases 2) Obsolete Test Cases.
* Re-usable Test cases can be used in succeeding regression cycles.
* Obsolete Test Cases can't be used in succeeding cycles.

### Prioritization of Test Cases

* Prioritize the test cases depending on business impact, critical & frequently used functionalities. Selection of test cases based on priority will greatly reduce the regression test suite.

## Selecting test cases for regression testing

It was found from industry data that a good number of the defects reported by customers were due to last minute bug fixes creating side effects and hence selecting the[Test Case](https://www.guru99.com/test-case.html)for regression testing is an art and not that easy.  Effective Regression Tests can be done by selecting the following test cases -

* Test cases which have frequent defects
* Functionalities which are more visible to the users
* Test cases which verify core features of the product
* Test cases of Functionalities which has undergone more and recent changes
* All Integration Test Cases
* All Complex Test Cases
* Boundary value test cases
* A sample of Successful test cases
* A sample of Failure test cases

## Regression Testing Tools

If your software undergoes frequent changes, regression testing costs will escalate.

In such cases, Manual execution of test cases increases test execution time as well as costs.

Automation of regression test cases is the smart choice in such cases.

The extent of automation depends on the number of test cases that remain re-usable for successive regression cycles.

**3) What is System testing?**

Ans- System Testing is the testing of a complete and fully integrated software product

Two Category of Software Testing

* Black Box Testing
* White Box Testing

System test falls under the **black box testing** category of software testing.

**White box testing** is the testing of the internal workings or code of a software application. In contrast, black box or System Testing is the opposite. System test involves the external workings of the software from the user's perspective.

System Testing involves testing the software code for following

* Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
* Verify thorough testing of every input in the application to check for desired outputs.
* Testing of the user's experience with the application.

**Software testing hierarchy**

* **Unit testing -** testing performed on each module or block of code during development. [Unit Testing](https://www.guru99.com/unit-testing-guide.html) is normally done by the programmer who writes the code.
* **Integration testing -** testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.
* **System testing -** testing done by a professional testing agent on the completed software product before it is introduced to the market.
* **Acceptance testing -** beta testing of the product done by the actual end users.

## Different Types of System Testing

There are more than 50 types of System Testing. For an exhaustive list of software testing types click [here](https://www.guru99.com/types-of-software-testing.html). Below we have listed types of system testing a large software development company would typically use

1. **Usability Testing -** [Usability Testing](https://www.guru99.com/usability-testing-tutorial.html) mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives
2. **Load Testing -** [Load Testing](https://www.guru99.com/load-testing-tutorial.html) is necessary to know that a software solution will perform under real-life loads.
3. **Regression Testing-** - [Regression Testing](https://www.guru99.com/regression-testing.html) involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.
4. **Recovery Testing -** Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.
5. **Migration Testing -** Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.
6. **Functional Testing -** Also known as functional completeness testing,[Functional Testing](https://www.guru99.com/functional-testing.html) involves trying to think of any possible missing functions. Testers might make a list of additional functionalities that a product could have to improve it during functional testing.
7. **Hardware/Software Testing -** IBM refers to Hardware/Software testing as "HW/SW Testing". This is when the tester focuses his/her attention on the interactions between the hardware and software during system testing.

**4) What is Agile methodology?**

## Ans- AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.

The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

|  |  |
| --- | --- |
| **Agile Model** | **Waterfall Model** |
| Agile method proposes incremental and iterative approach to software design | Development of the software flows sequentially from start point to end point. |
| The **agile process** is broken into individual models that designers work on | The design process is not broken into an individual models |
| The customer has early and frequent opportunities to look at the product and make decision and changes to the project | The customer can only see the product at the end of the project |
| Agile model is considered unstructured compared to the waterfall model | Waterfall model are more secure because they are so plan oriented |
| Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time. | All sorts of project can be estimated and completed. |
| Error can be fixed in the middle of the project. | Only at the end, the whole product is tested. If the requirement error is found or any changes have to be made, the project has to start from the beginning |
| Development process is iterative, and the project is executed in short (2-4) weeks iterations. Planning is very less. | The development process is phased, and the phase is much bigger than iteration. Every phase ends with the detailed description of the next phase. |
| Documentation attends less priority than software development | Documentation is a top priority and can even use for training staff and upgrade the software with another team |
| Every iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released. | Only after the development phase, the testing phase is executed because separate parts are not fully functional. |
| In agile testing when an iteration end, shippable features of the product is delivered to the customer. New features are usable right after shipment. It is useful when you have good contact with customers. | All features developed are delivered at once after the long implementation phase. |
| Testers and developers work together | Testers work separately from developers |
| At the end of every sprint, user acceptance is performed | User acceptance is **performed** at the end of the project. |
| It requires close communication with developers and together analyze requirements and planning | Developer does not involve in requirement and planning process. Usually, time delays between tests and coding |

## Scrum

SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, Scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members). It consists of three roles, and their responsibilities are explained as follows:

* Scrum Master
  + Master is responsible for setting up the team, sprint meeting and removes obstacles to progress
* Product owner
  + The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration
* Scrum Team
  + Team manages its own work and organizes the work to complete the sprint or cycle

## Process flow of Scrum Methodologies:

Process flow of scrum testing is as follows:

* Each iteration of a scrum is known as Sprint
* Product backlog is a list where all details are entered to get the end-product
* During each Sprint, top user stories of Product backlog are selected and turned into Sprint backlog
* Team works on the defined sprint backlog
* Team checks for the daily work
* At the end of the sprint, team delivers product functionality