1. Difference between
   1. Smoke and Sanity

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke testing is done to assure that the major functionalities of program is working fine. | Sanity testing is done to check the bugs have been fixed after the build. |
| Smoke testing is also called subset of acceptance testing. | Sanity testing is also called subset of regression testing. |
| Smoke testing is documented. | Sanity testing isn’t documented. |
| Smoke testing is performed by either developers or testers. | Sanity testing is normally performed by testers. |
| Smoke testing is used to test all over function of the system/product. | Sanity testing is used in the case of only modified or defect functions of system/products. |
| Smoke testing can be performed either manually or by using automation tools. | Sanity testing is commonly executed manually, not by using any automation approach. |
| Smoke testing is performed when new product is built. | Sanity testing is conducted after the completion of regression testing. |
| It includes all the system’s essential basic functionality. | It includes only those modules where change in code is made. |
| Smoke Testing firstly performs on the initial build. Smoke testing is done first. | Sanity Testing is done on stable builds or for the introduced new features in the software. |
| Smoke testing can be carried out either way-manually or automatically. | Without using test cases or scripts sanity testing can be carried out. |
| There is end-to-end system verification done in smoke testing. | A specific component gets verified in sanity testing. |
| In the smoke testing process, the software build could be stable or unstable. | During sanity testing, the software build is comparatively stable. |
| For every new build release smoke testing is carried out. | Sanity testing is carried out when in-depth testing is not possible because of short time. |

* 1. Verification and Validation

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| We check whether we are developing the right product or not. | We check whether the developed product is right. |
| Verification is also known as static testing. | Validation is also known as dynamic testing. |
| Verification includes different methods like Inspections, Reviews, and Walkthroughs. | Validation includes testing like functional testing, system testing, integration, and User acceptance testing. |
| It is a process of checking the work-products (not the final product) of a development cycle to decide whether the product meets the specified requirements. | It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specified business requirements. |
| Quality assurance comes under verification testing. | Quality control comes under validation testing. |
| The execution of code does not happen in the verification testing. | In validation testing, the execution of code happens. |
| In verification testing, we can find the bugs early in the development phase of the product. | In the validation testing, we can find those bugs, which are not caught in the verification process. |
| Verification testing is executed by the Quality assurance team to make sure that the product is developed according to customers' requirements. | Validation testing is executed by the testing team to test the application. |
| Verification is done before the validation testing. | After verification testing, validation testing takes place. |
| In this type of testing, we can verify that the inputs follow the outputs or not. | In this type of testing, we can validate that the user accepts the product or not. |

1. What is Agile Methodology?

Agile is a project management and software development approach that aims to be more effective.

1. It focuses on delivering smaller pieces of work regularly instead of one big launch.
2. This allows teams to adapt to changes quickly and provide customer value faster.

Agile methodologies are iterative and incremental, which means it’s known for breaking a project into smaller parts and adjusting to changing requirements.

1. They prioritize flexibility, collaboration, and customer satisfaction.
2. Major companies like Facebook, Google, and Amazon use Agile because of its adaptability and customer-focused approach.

3. Epic and User stories

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Epic** | **User Story** |
| Definition | It is a piece of work that’s broken down into specific tasks based on the needs of customers or end-users. | User story is a piece of functionality from the perspective of an end-user, which developers use to build the software. |
| Scope | Its scope is along multiple sprints | Its scope is within a given iteration/sprint |
| Duration | It is long term | It is short term. |
| Relation | It provides foundation to user stories | It is a subset of epics. |
| Target | It represents a part of work for achieving the task | It represents a specific and manageable task. |
| Granularity | It is less detailed | It is more detailed |
| Span | Epics span multiple functionalities or modules within a software system. | User Story is small and focused for a particular feature. |