**TASK 4**

**Difference between smoke testing and sanity testing**

Smoke testing is done to assure that the acute 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** | **Sanity Testing** |
| Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | Sanity Testing is done to check the new functionality/bugs have been fixed |
| The objective of this testing is to verify the “stability” of the system in order to proceed with more rigorous testing | The objective of the testing is to verify the “rationality” of the system in order to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing in software testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of [Regression Testing](https://www.guru99.com/regression-testing.html) |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

**Difference between validation and verification**

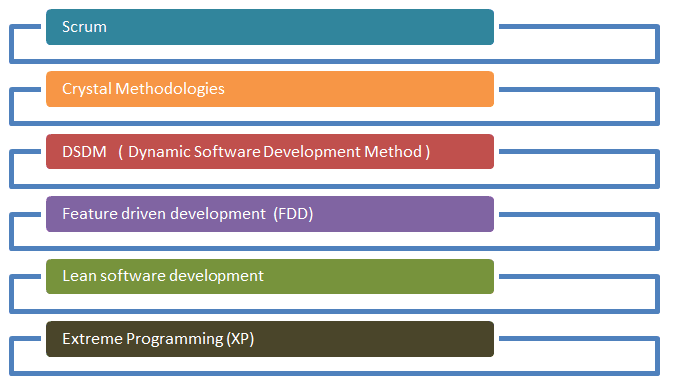
**Verification:** An internal process that occurs early in the development process. It involves checking the final status of a software product, including documents, designs, and other programming aspects. Verification is a quality control process that determines if a system meets its system-level requirements.

**Validation:** An external procedure that occurs after verification. It involves checking whether the specification captures the customer's requirements. Validation is the process of evaluating a system, component, or product during or at the end of the development cycle to determine its fitness for its intended purpose.

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| The verifying process includes checking documents,  design, code, and program | It is a dynamic mechanism of testing and validating the actual product |
| It does ***not*** involve executing the code | It always involves executing the code |
| Verification uses methods like reviews, walkthroughs, inspections, and desk- checking etc. | It uses methods like Black Box Testing, [White Box Testing](https://www.guru99.com/white-box-testing.html), and non-functional testing |
| Whether the software conforms to specification  is checked | It checks whether the software meets the requirements and expectations of a customer |
| It finds bugs early in the development cycle | It can find bugs that the verification process cannot catch |
| Target is application and software architecture, specification, complete design, high level,  and database design etc. | Target is an actual product |
| QA team does verification and make sure that  the software is as per the requirement in the  SRS document. | With the involvement of testing team validation is executed on software code. |
| It comes before validation | It comes after verification |
|  |  |

**Agile Process**

Below [Agile methodology](https://www.guru99.com/agile-testing-course.html) process to deliver successful systems quickly.

[](https://www.guru99.com/images/11-2014/agile_Processesv1_2.png)

**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)

**Scrum Master:** Scrum Master is responsible for setting up the team, sprint meeting and removes obstacles to progress

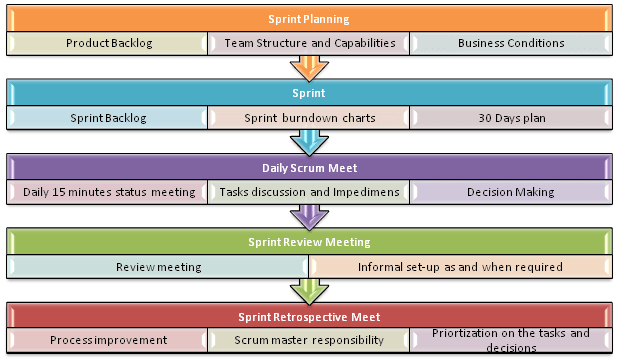
**Product owner:** The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration

**Scrum Team:** Team manages its own work and organizes the work to complete the sprint or cycle

**Product Backlog:** This is a repository where requirements are tracked with details on the no of requirements(user stories) to be completed for each release. It should be maintained and prioritized by Product Owner, and it should be distributed to the scrum team. Team can also request for a new requirement addition or modification or deletion.

## Scrum Practices

Practices are described in detailed as below.

[](https://www.guru99.com/images/11-2014/agile_Processesv1_4.png)

## Process flow of Scrum Methodologies:

* 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

Extreme Programming (XP)

Extreme Programming technique is very helpful when there is constantly changing demands or requirements from the customers or when they are not sure about the functionality of the system. It advocates frequent “releases” of the product in short development cycles, which inherently improves the productivity of the system and also introduces a checkpoint where any customer requirements can be easily implemented. The XP develops software keeping customer in the target.

**Extreme Programming**

Business requirements are gathered in terms of stories. All those stories are stored in a place called the parking lot.

In this type of methodology, releases are based on the shorter cycles called Iterations with span of 14 days’ time period. Each iteration includes phases like coding, unit testing and system testing where at each phase some minor or major functionality will be built in the application.

**Phases of eXtreme programming:** There are 6 phases available in Agile XP method, and those are explained as follows:

**Planning**

* Identification of stakeholders and sponsors
* Infrastructure Requirements
* Security related information and gathering
* Service Level Agreements and its conditions

**Analysis**

* Capturing of Stories in Parking lot
* Prioritize stories in Parking lot
* Scrubbing of stories for estimation
* Define Iteration SPAN(Time)
* Resource planning for both Development and QA teams

**Design**

* Break down of tasks
* Test Scenario preparation for each task
* Regression Automation Framework

**Execution**

* Coding
* Unit Testing
* Execution of Manual test scenarios
* Defect Report generation
* Conversion of Manual to Automation regression test cases
* Mid Iteration review
* End of Iteration review

**Wrapping**

* Small Releases
* Regression Testing
* Demos and reviews
* Develop new stories based on the need
* Process Improvements based on end of iteration review comments

**Closure**

* Pilot Launch
* Training
* Production Launch
* SLA Guarantee assurance
* Review SOA strategy
* Production Support

There are two storyboards available to track the work on a daily basis, and those are listed below for reference.

**Story Cardboard:** This is a traditional way of collecting all the stories in a board in the form of stick notes to track daily XP activities. As this manual activity involves more effort and time, it is better to switch to an online form.

**Online Storyboard:** Online tool Storyboard can be used to store the stories. Several teams can use it for different purposes.

**Crystal Methodologies:** Crystal Methodology is based on three concepts

* **Chartering:** Various activities involved in this phase are creating a development team, performing a preliminary feasibility analysis, developing an initial plan and fine-tuning the development methodology
* **Cyclic delivery**: The main development phase consists of two or more delivery cycles, during which the

1. Team updates and refines the release plan
2. Implements a subset of the requirements through one or more program test integrate iterations
3. Integrated product is delivered to real users
4. Review of the project plan and adopted development methodology

* **Wrap Up:** The activities performed in this phase are deployment into the user environment, post- deployment reviews and reflections are performed.

**Dynamic Software Development Method (DSDM)**

DSDM is a Rapid Application Development (RAD) approach to software development and provides an agile project delivery framework. The important aspect of DSDM is that the users are required to be involved actively, and the teams are given the power to make decisions. Frequent delivery of product becomes the active focus with DSDM. The techniques used in DSDM are

1. Time Boxing
2. Moscow Rules
3. Prototyping

The DSDM project consists of 7 phases

* Pre-project
* Feasibility Study
* Business Study
* Functional Model Iteration
* Design and build Iteration
* Implementation
* Post-project

**Feature Driven Development (FDD)**

This method is focused around “designing & building” features. Unlike other Agile methods in software engineering, FDD describes very specific and short phases of work that has to be accomplished separately per feature. It includes domain walkthrough, design inspection, promote to build, code inspection and design. FDD develops product keeping following things in the target

* Domain object Modeling
* Development by feature
* Component/ Class Ownership
* Feature Teams
* Inspections
* Configuration Management
* Regular Builds
* Visibility of progress and results

**Lean Software Development**

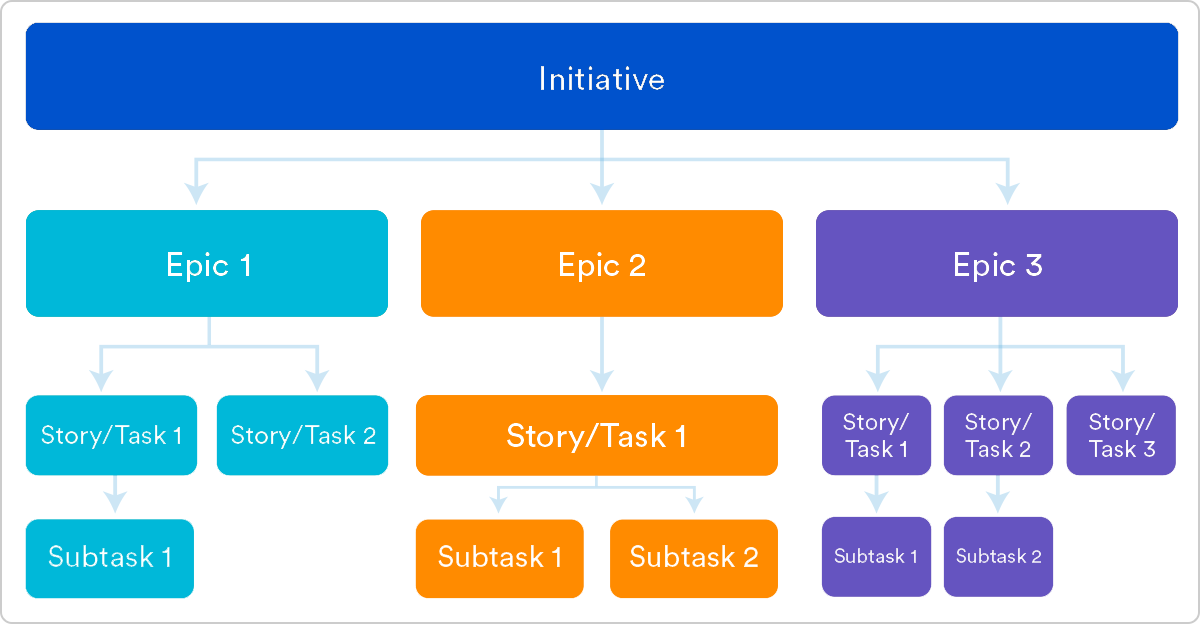
Lean software development method is based on the principle “Just in time production”. It aims at increasing speed of software development and decreasing cost. Lean development can be summarized in seven steps.

* Eliminating Waste
* Amplifying learning
* Defer commitment (deciding as late as possible)
* Early delivery
* Empowering the team
* Building Integrity
* Optimize the whole

**Kanban**

Kanban originally emerged from Japanese word that means, a card containing all the information needed to be done on the product at each stage along its path to completion. This framework or method is quite adopted in software testing method especially in Agile concepts.

**Explain about Epic and user stories**



1. Epics and user stories are two methods for building new software features in agile software development. Epics are large bodies of work that can be broken down into smaller tasks, called stories, or user stories. Epics are a way for teams to prioritize their work and tie it together. User stories are short descriptions of how software should work from the end user's perspective.
2. The main difference between the two is that user stories are small, while epics are larger. User stories are typically added directly to the product backlog, and should be phrased in simple language that outlines for who and how they will generate value. For example, "As a customer, I want to know about upcoming events so I can plan my visit".
3. Epics are user stories that are too big to fit in one sprint or iteration. For example, "As an authorized user, I can run reports so that I can see the financial state of the organization".
4. Epics are helpful ways to organize work and create a hierarchy. Epics do not have any time frames, but releases do. Any bigger part of a project that has to release within a specified time will be treated as a release.