**\* Some terminologies:**

**Error:**

* An error is a mistake made by a human during the software development process. It is a human action that produces an incorrect or unexpected result.
* An **error** is a human mistake.

**Defect:**

* A defect is a flaw or imperfection in a software product. It is a variance between the expected and actual behavior of the system.
* A **defect** is a flaw in the software.

**Fault:**

* A fault is a static defect in the code that may lead to errors. It is a mistake or an error in the source code that, when executed, causes the program to behave unexpectedly.
* A **fault** is a static defect in the code.

**Bug:**

* The terms "bug" and "defect" are often used interchangeably. A bug is a general term for any error, flaw, mistake, fault, or defect in a computer program that prevents it from behaving as intended.
* A **bug** is a general term for any kind of software defect.
* Defect accepted by development team is called bug.

**Failure:**

* A failure occurs when a software application or system does not perform its intended function. It is the deviation of the software from its expected result.
* A **failure** is the deviation of the software from its expected result.
* Lots of defect leads to failure of the software.

**Missing/Wrong:**

* "Missing" refers to the absence of a functionality that should be present in the software but is not implemented. "Wrong" indicates that the implemented functionality does not conform to the specified requirements.
* **Missing** is the absence of expected functionality, and **wrong** is the incorrect implementation of functionality.

A person can make an error (mistake), which can lead to the introduction of a defect (fault or bug) in the software code or in some other related work product. If a defect in the code is executed, this may cause a failure.

For example, a requirements elicitation error can lead to a requirements defect, which then results in a programming error that leads to a defect in the code.

Errors may occur for many reasons, such as:

* Time pressure
* Inexperienced or insufficiently skilled project participants
* Miscommunication between project participants, including miscommunication about requirements and design
* Complexity of the code, design, architecture, the underlying problem to be solved, and/or the technologies use

**Root Causes:** The root causes of defects are the earliest actions or conditions that contributed to creating the defects.

**Defects:** Defects can be analyzed to identify their root causes, so as to reduce the occurrence of similar defects in the future.

**Effects:** Effect is the result of the defects.

**Project**: If software application is developed for specific customer based on the requirement then it is called project.

**Product:** if software application is developed for multiple customer based on market requirement then it is called product.

**Testing**

* Executing tests can show failures that are caused by defects in the software.
* Testers are responsible for the initial test and the final confirmation test.

**Debugging**

* Debugging is the development activity that finds, analyzes, and fixes such defects.
* Developers do the debugging, associated component and component integration testing (continues integration)

**\* Verification and Validation**

**What is Verification?**

**Definition :** The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

* Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection, design analysis and specification analysis. It is a relatively objective process.
* Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful.Verification is concerned with whether the system is well-engineered and error-free.

Methods of Verification : Static Testing

* Walkthrough
* Inspection
* Review

**What is Validation?**

**Definition:** The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

Methods of Validation : Dynamic Testing

* Testing
* End Users

A clickable button with name Submet

* Verification would check the design doc and correcting the spelling mistake.
* Otherwise, the development team will create a button like



Example of Verification

So new specification is “A clickable button with name **Submit**”

Once the code is ready, Validation is done. A Validation test found –



Example of Validation

* Owing to Validation testing, the development team will make the submit button clickable

|  |  |
| --- | --- |
| 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 can not 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 |
| Are we implementing the system right | Are we implementing the right system |
| Verifies that outputs are according to inputs or not | Validates that the user accept the software or not |