Software testing:-

Software testing is **the process of assessing the functionality of a software program**. The process checks for errors and gaps and whether the outcome of the application matches desired expectations before the software is installed and goes live.

**Bugs:-**

an error that may cause components or the system to fail to perform its required functions. For ex,  incorrect data description, statements, input data, design,etc

there are 2 types of bugs **Functional bugs**. **Content bugs**

Error:-

Error means mistake in code there are six types: **constant error, systematic error, random error, absolute error, relative error and percentage error**.

Mistake:-

The software is used incorrectly and so does not behave as we expected.

Fault:-

It arises when the expected result don't match with the actual results.

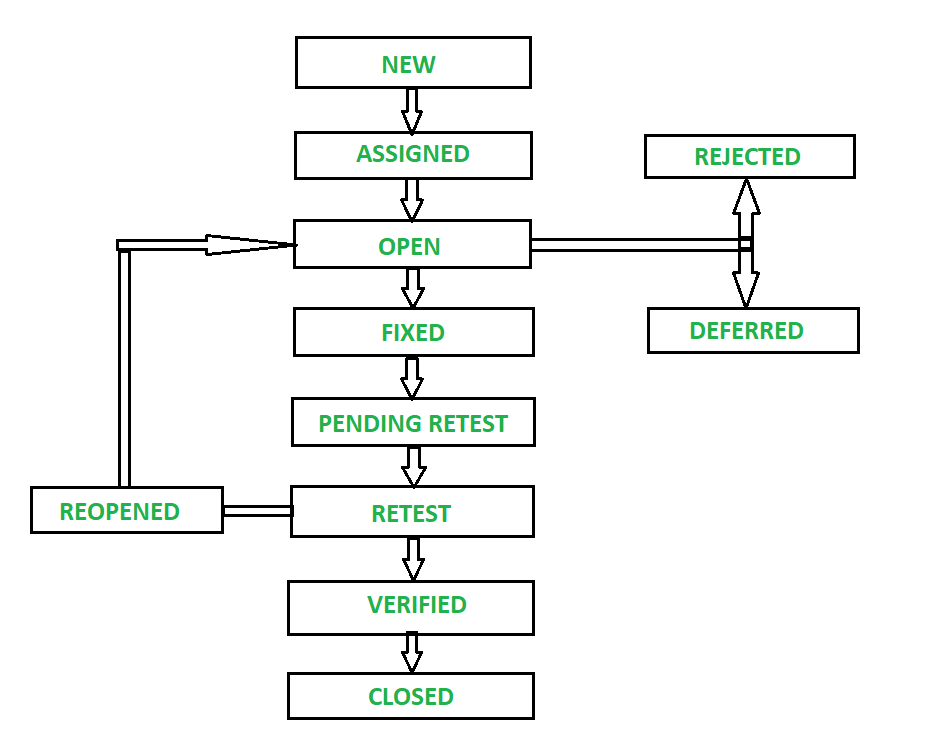
Defect:-

A defect is a system error that doesn't allow the intended action to be completed.

Failure:-

A failure is the inability of a software system or component to perform its required functions within specified performance requirements. When a defect reaches the end customer it is called a Failure. During development, Failures are usually observed by testers.

**BUG LIFE CYCLE**



**1. NEW –**

It is first state of Bug Life Cycle. The tester provides a proper Defect document to Development team so that development team can refer to Defect Document and can fix bug accordingly.

**2. ASSIGNED –**

The newly identified defect is assigned to the development team for working on defect and to resolve that. When the defect is assigned to developer team then status of bug changes to ‘Assigned’ state.

**3. OPEN –**

Based on some specific reason if developer team feels that defect is not appropriate then it is transferred to either ‘Rejected’ or ‘Deferred’ state.

**4. FIXED –**

After necessary changes of codes or after fixing identified bug developer team marks state as ‘Fixed’.

**5. PENDING RETEST –**

During the fixing of defect is completed, developer team passes new code to testing team for retest. And the code is pending for retesting at Tester side so status is assigned as ‘Pending Retest’.

**6. RETEST –**

At this stage, tester starts work of retesting defect to check whether defect is fixed by developer or not, and the status is marked as ‘Retesting’.

**7. REOPEN –**

After ‘Retesting’ if tester team found that bug continues like previously even after developer team has fixed the bug, then status of bug is again changed to ‘Reopened’. Once again bug goes to ‘Open’ state and goes through life cycle again. This means it goes for Re-fixing by the developer team.

**8. VERIFIED –**

The tester re-tests bug after it got fixed by developer team and if tester does not find any kind of defect/bug then bug is fixed and status assigned is ‘Verified’.

**9. CLOSED –**

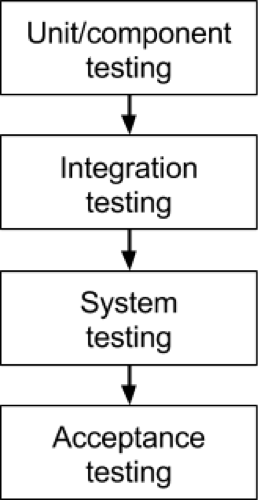
It is the final state of Defect Cycle, after fixing defect by developer team when testing found that the bug has been resolved and it does not persist then they mark defect as a ‘Closed’ state.

Different Levels of Testing

The levels of software testing involve the different methodologies, which can be used while we are performing the software testing.

In [software testing](https://www.javatpoint.com/software-testing-tutorial), we have four different levels of testing, which are as discussed below:

* **Unit Testing**
* **Integration Testing**
* **System Testing**
* **Acceptance Testing**



### Level1: Unit Testing

* **Unit testing** is the first level of software testing, which is used to test if software modules are satisfying the given requirement or not.
* The first level of testing involves **analyzing each unit or an individual component** of the software application.

.**Level2: Integration Testing**

* The second level of software testing is the **integration testing.** The integration testing process comes after **unit testing**.
* Checking the connection between the module.
* It is mainly used to test the **data flow from one module or component to other modules.**
* In integration testing, the **test engineer** tests the units or separate components or modules of the software in a group.

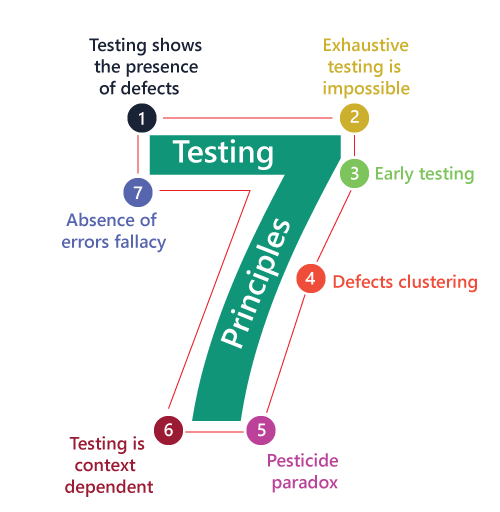
### Level3: System Testing

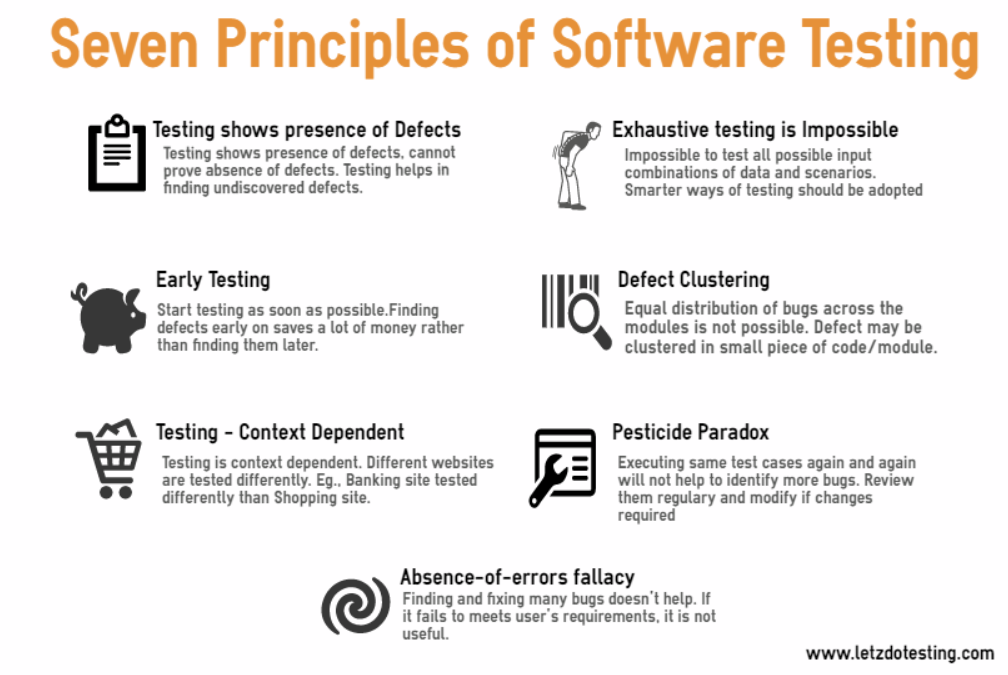
* The third level of software testing is **system testing**, which is used to test the software's functional and non-functional requirements.
* It is **end-to-end testing** where the testing environment is parallel to the production environment. In the third level of software testing, **we will test the application as a whole system.**
* To check the end-to-end flow of an application or the software as a user is known as **System testing**.

### Level4: Acceptance Testing

* The **last and fourth level** of software testing is **acceptance testing**, which is used to evaluate whether a specification or the requirements are met as per its delivery.
* The software has passed through three testing levels (**Unit Testing, Integration Testing, System Testing**). Some minor errors can still be identified when the end-user uses the system in the actual scenario.
* In simple words, we can say that Acceptance testing is the **squeezing of all the testing processes that are previously done.**
* The acceptance testing is also known as **User acceptance testing (UAT)** and is done by the customer before accepting the final product.

**Software Testing Principles**





**DECISION COVERAGE:-**

Demo(int a) {

If (a> 5)

a=a\*3

Print (a)

}

**BRANCH COVERAGE:-**

Read A    
Read B    
IF A+B > 50 THEN    
Print "Large"    
ENDIF    
If A+B<50 THEN    
Print "Small"    
ENDIF

Read A and B

A+B>50

**STATEMENT COVERAGE:-**

Prints (int x, int y, int z)

{

Int x,y,z;

If (x=3)

Print ("display\_messageX", result)

If (y=2)

Print ("display\_messageY", result)

Else

Print ("display\_messageZ", result)

}

