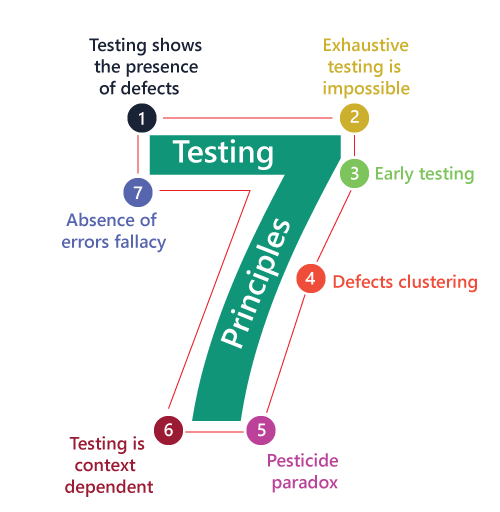
**Software Testing Principles**

Software testing is a procedure of implementing software or the application to identify the defects or bugs. For testing an application or software, we need to follow some principles to make our product defects free, and that also helps the test engineers to test the software with their effort and time. Here, in this section, we are going to learn about the seven essential principles of software testing.

Let us see the seven different testing principles, one by one:

* Testing shows the presence of defects
* Exhaustive Testing is not possible
* Early Testing
* Defect Clustering
* Pesticide Paradox
* Testing is context-dependent
* Absence of errors fallacy



1. *Testing shows presence of Defects :*

* Software testing reduces the presence of Defects.
* Software testing talks about the Presence of Defects and doesn’t talk about the Absence of Defects.
* Even Multiple testing can never ensure that Software is 100 % Bug-Free.

1. *Exhaustive Testing is not Possible :*

* Exhaustive Testing is impossible means the Software can never test at every test cases.
* It can test only some test cases and assume that Software is correct and it will produce the correct output in every test cases.

1. *Early Testing :*

* The Defect detect in early phases of SDLC will very less expensive.
* For better performance of software, start software testing at Initial phases.

1. *Defect Clustering :*

* In a project, a small number of the module can contain most of the Defects.
* ***Pareto Principle***to software testing state that 80 % of software defect comes from 20 % of Modules.

1. *Pesticide Patradox :*

* Repeating the same test cases again and again will not find new Bugs.
* It is necessary to review the test cases and add or update test cases to find new Bugs.

1. *Testing is Context Dependent :*

* Testing approach depends on context of software developed.
* Different types of software need to perform different types of testing.
* The testing of the E-Commerce site is different from the testing of Android application.

1. *Absence of error fallacy :*

* If a built software is 99 % bug-free but it does not follow the user requirement then it is Unusable.
* It is not only necessary that software is 99 % bug-free but is also mandatory to fulfil all the customer requirements.