*3.3. Software Testing*

**WHY IS TESTING NECESSARY?**

Software Testing is necessary because we all make mistakes.

Software testing is very important because of the following reasons:

* Software testing is really required to point out the defects and errors that were made during the development phases.
* It’s essential since it makes sure of the Customer’s reliability and their satisfaction in the application.
* It is very important to ensure the Quality of the product.  Quality product delivered to the customers helps in gaining their confidence.
* Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
* Testing is required for an effective performance of software application or product.
* It’s important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
* It’s required to stay in the business

**Software System Context:**

Testing Is Context Dependent

**Causes of Software Defects:**

Error 🡪 Defect 🡪 Failure

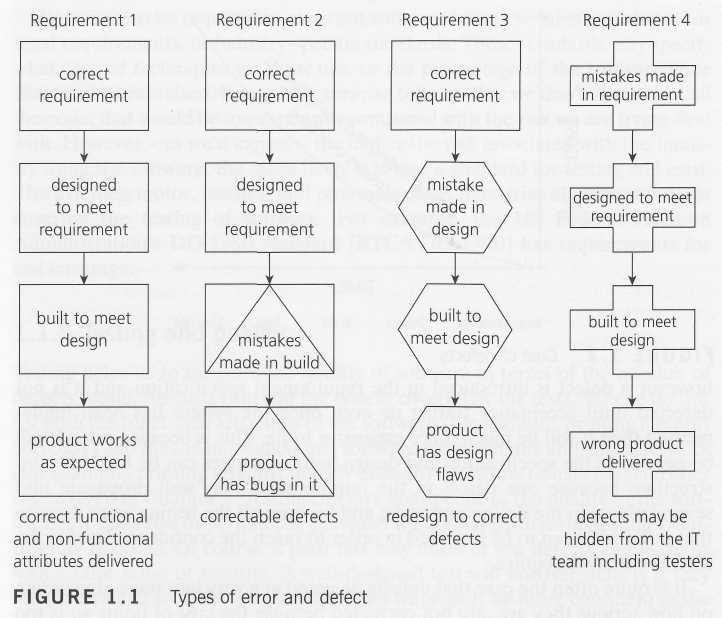
Defects and failures basically arise from:

* Errors in the specification, design and implementation of the software and system
* Errors in use of the system
* Environmental conditions
* Intentional damage
* Potential consequences of earlier errors

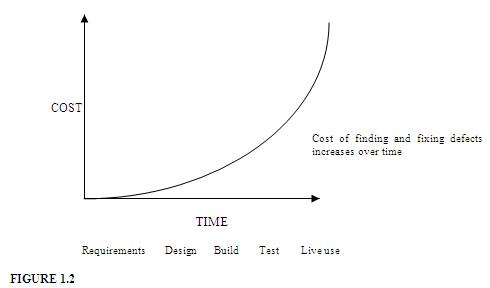
**When do defects arise?**

Because of the following reasons the software defects arise:

* The person using the software application or product may not have enough knowledge of the product.
* Maybe the software is used in the wrong way which leads to the defects or failures.
* The developers may have coded incorrectly and there can be defects present in the design.
* Incorrect setup of the testing environments.



**What is the cost of defects?**

**[](http://istqbexamcertification.com/wp-content/uploads/2011/12/cost-of-defects.jpg)**

**How much testing is enough?**

**What is Testing? \***

* Software testing is a process of executing a program or application with the intent of finding the **software bugs**.
* It can also be stated as the **process of validating and verifying**that a software program or application or product:

1. Meets the business and technical requirements that guided it’s design and development
2. Works as expected
3. Can be implemented with the same characteristic.

**Objectives of Software Testing: \***

* Finding defects which may get created by the programmer while developing the software.
* Gaining confidence in and providing information about the level of quality.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.

**Basic Terminologies in Testing:**

* Error: A Human mistake is called as an ‘error’.
* Defect: The discrepancy between the actual and expected results is known as ‘defect’.
* Bug: If testers find any mismatch in the system/application in testing phase it is termed as ‘Bug’.
* Failure: When build system does not meet users’ requirements, it results into failure.
* Test: A Test is a group of related test cases, or a group of related test cases and test procedures.
* Test Set: A group of related tests is sometimes referred to as a test set.
* Test Suite: A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a Test Suite.
* Test Case: A test case in a practical sense is attest related item which contains the following Information such as test inputs, execution conditions and expected outputs.
* Test Oracle: Test Oracle is a piece of software that allows tester to evaluate whether a test has been passed or failed.
* Test Bed: A test bed is an environment that contains all the hardware and software which is needed to test a software system.

**Principles of Testing: \***

1. Testing shows presence of defects.
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence – of – errors fallacy

**Testing shows presence of defects:**

The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn’t talk about the absence of defects. Software testing can ensure that defects are present but it cannot prove that software is defects free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not removes all defects.

**Exhaustive testing is not possible:**

It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. 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. If the software will test every test cases then it will take more cost, effort, etc. and which is impractical.

**Early Testing:**

To find the defect in the software, early test activity shall be started. The defect detected in early phases of SDLC will very less expensive. For better performance of software, software testing will start at initial phase i.e. testing will perform at the requirement analysis phase.

**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.

**Pesticide paradox:**

Repeating the same test cases again and again will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

**Testing is context dependent:**

Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example, the testing of e-commerce sites is different from the testing of an Android application.

**Absence of errors 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 it also mandatory to fulfil all the customer requirements.

**Test Oracles \***

* A test oracle is a process defined to check the correctness of the program output for the designed test cases. Test cases are provided to the test oracle and the program under the process of software testing.
* Since test are the sequence of inputs and expected outputs, an oracle is a predicate used to determine whether a given sequence is acceptable or not and will respond with an outcome in terms of pass or fail for any input it is defined for. Therefore, it determines whether the system will behave correctly for test execution.

The oracle might be:

* A program which takes the similar input and produces similar output.
* A documentation that produces specific outputs for specific inputs.
* An algorithm that can be used to calculate correct outputs for given inputs.
* A human domain expert who can analyse whether the output is correct or not.
* It is a mechanism used for determining whether the test case in a program has passed or failed.

An oracle has three capabilities as mentioned below:

* A *generator*, which provide expected results for each test.
* A *comparator*, compare condition and actual results.
* An *evaluator*, determine whether the actual and expected results are as per the specification mentioned in a test case.

**Levels Of Testing** \*

* Component Testing
* Integration Testing
* System Testing
* Acceptance Testing

