Software Quality Engineering

Error:

A mistake made by a programmer during coding is called an error.

Defect:

An error found during the unit testing in the development phase is called a defect.

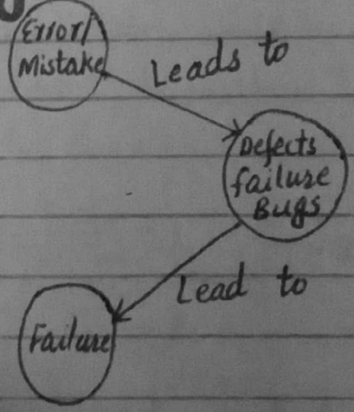
Bug:

Error found during the testing phase is called a bug.

Failure:

When an error is found at an ~~end~~ user's end is called as the failure.

Why Software Testing:

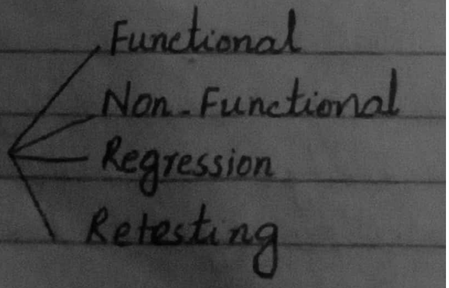


1. Quality measure
2. Problem Identify

Types of Testing:

1. Black box
2. White box
3. Regression
4. Retesting

Black box Testing:



Black box / closed Testing

- Functional
  - Unit
  - Integration
  - Smoke
  - Accaptance
- Regression
- Non-Functional
  - Load
  - stress
  - strain
  - Volume
  - Security
  - Performance
- Retesting

Black box Testing:

Definition: is defined as a software testing methodology in which the tester analyze application functionality without a thorough knowledge of its internal design

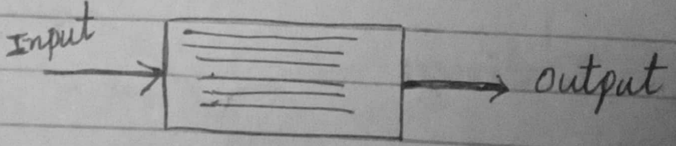
→ It is carried out by tester.

→ It is the least time consuming.

→ Black box testing is not considered for algorithm testing.

→ It is the behavior testing of the software.

→ Also known as data-driven testing. functional testing, closed box testing.



White box testing:

Definition: is defined as a software testing methodology in which the tester's knowledge of the application's internal workin is leveraged during testing.

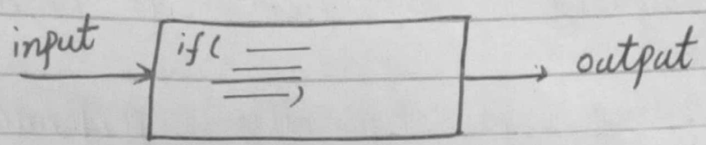
→ It is carried out by software developer.

→ It is most time consuming.

→ White box testing is well suitable for algorithm testing.

→ It is the logic testing of the software

→ It is also known as clear box, code-based testing, structural testing and trans-parent testing.



Functional Testing:

Functional testing verifies each function or feature of the software.

→ Functional testing can be done manually

Non-Functional Testing:

Non-Functional testing verifies non-functional aspects like performance, usability, reliability etc.

→ Non-functional testing is hard to perform manually.

Regression Testing:

This technique involves retesting the entire system or a part of the system to ensure that the existing functionality of the software is still working as expected after making changes.

→ Regression testing can be performed manually or using automated tools.

→ Regression testing is performed afte making any change in the software system, including bug fixes, enhanceme or new features.

Re-testing:

Re-testing involves re-testing a specific part of the software that was previously identified as having a defect.

→ Re-testing is typically performed after a bug has been fixed, and the software has been reworked to address the defect.

→ The objective of re-testing is to ensure that the specific issue that was previously identified has been resolved and the software now works as expected.

What is test case:

A test case refers to the action required to verify a specific feature or functionality in software testing.

→ Test cases are more detailed with several parameters.

→ Test cases are Low-Level actions

→ It focuses on "What to test" and "How to test"

→ It requires more time compared to test scenarios.

→ They are hard to maintain.

What is Test Scenario?

The test Scenario provides a small description of what needs to be performed based on the use case.

→ Test scenario provides a small description, mostly one-line statements.

→ Test scenario's are high-level actions

→ It focuses more on "What to test"

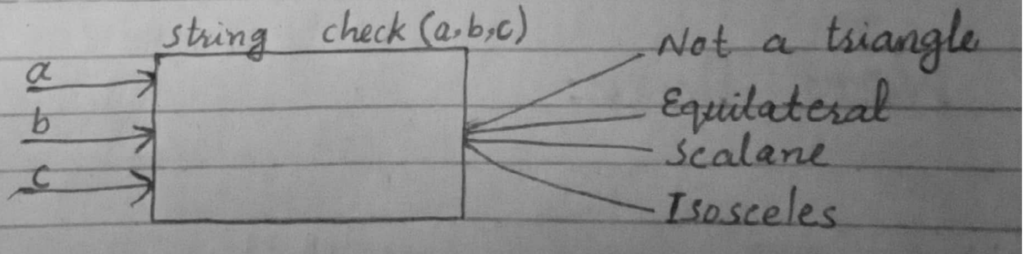
→ Test scenario's require less time.

→ They require less time to maintain.

Unit Testing:

Unit testing focuses on a specific unit of code, such as a function, method or class. It aims to test this unit independently of other parts of the application.

Triangle problem:



Conditions:

$C_1 : 1 \leq a \leq 200$

$C_2 : 1 \leq b \leq 200$

$C_3 : 1 \leq c \leq 200$

$C_4 : a < b+c$

$C_5 : b < a+c$

$C_6 : c < a+b$

$C_7 :$ $a = b$ && $b = c$ Equilateral

$C_8 :$ $a \neq b$ && $a \neq c$ && $b \neq c$ Scalane

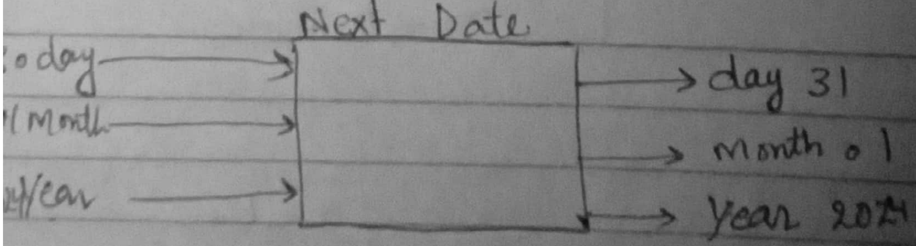
$C_9 :$ $a = b$ or $b = c$ or $a = c$ Isosceles

```
If((a≥1 && a≤200) && (b≥1 &&b≤200)&&
  (c≥1 && c≤200))

  if (a<b+c && b<a+c && c<a+b)

    if (a=b && b=c && a=c)
      {
        'Equilateral'
      }
    Else if (a=b OR b=c OR a=c)
      {
        'Isosceles'
      }
    Else if (a≠b && a≠c && b≠c)
      {
        'Scalane'
      }
  Else
    Not a Triangle
Else
  Invalid
```

Next Date:



```
Leap year = 0, Day, Month, Year        c1: 1 ≤ Day ≤ 31
                                       c2: 1 ≤ Month ≤ 12
                                       c3: 1992 ≤ Year ≤ 2024
if(Month  1  3  ~~8~~5  7  8  10)
    if (Day < 31)
    {
      Day = Day+1;
    }
  else if (Day == 31)
    {
      Day = 1 , month = month +1
    }
  else if (Month  4  6  9  11)
    if (day < 30)
      day = day+1
    else if (day == 30)
    {
      day=1 , month = month+1;
    }
  else if ( month == 12)
    if (day < 31)
      day = day +1
    else if (day == 31)
      day = 1 , month = 1 , year++
    else if (month == 2)
      if (day < 28)
        day ++;
      else if (day == 28 && leapyear == 0)
        day = 1   month++;
    else if (day = 28   leap)
        Day++;
  if (day 29 && leap == 1)
        day = 1
  else if (day 29 && leap == 0)
        error.
```