Page:
Date: / /

Practical No:-6

Name:- Ishwari Rajiv Narkhede
PRN :- 1841034
Batch :- B4
Branch:- computer Engineering
Aim :- Learn about different techniques & testing a software and design unit test cases to verify the functionality and locate bugs, if any.

Theory :-

Software Testing -

Testing software is an important part of the developement life cycle of a software. It is an expensive activity. Hence, appropriate testing methods are neeessary for ensuring realiability of a program. According to ANSI/IEE 1059 standard, the definition of testing is the process of analyzing a software item, to detect the differences bet^{n} existing and required conditions, i.e. detects/ errors/ bugs and to evaluate the features of the software item.

Testing Frameworks -

Following are the different testing frameworks:

1) junit - For Java unit test

Page :
Date : / /

2) Selenium - is a suite of tools for automating web applications for software testing purposes, plugin for firefox.

3) HP QC - Is the HP web-based test management tool. It formiliarizes with the process of defining releases, specifying requirements, planning, tests, executing tests, tracking defects, alerting on changes and analyzing results.

4) IBM Rational - Rational s/w has a solution to support business sector for designing, implementing and testing software

* Need for software testing -

There are many reasons for why we should test s/w, such as -

1) software testing identifies that s/w faults. The removal of faults helps reduce the no. of system failures.

2) software testing can also improves the other system qualities such as maintainability, usability, and testability.

3) In order to meet the condition that the last few years of the 20th century systems had to be shown to be free from the "Millennium bug".

4) In order to meet the industry specific standards such as the aerospace, missile and railway signaling standards.

5) In order to meet the different legal requirements.

* Test cases and test suite :-

A test cases describes an input descriptions and an expected output description. Input are of two types: preconditions and the actual inputs that are identified by some testing methods.

The set of test cases is called a test suite.

* Types of Testing :-

1) Unit Testing :-

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.

e.g.,

a) In a program we are checking if loop, method or function is working fine

b) Misunderstood or incorrect, arithmetic precedence

c) Incorrect initialization.

2) Integration Testing :-

The objective is to take unit tested components and build a program structure that

has been dictated by design. Integration testing in which a group of components is combined to produce output.

Integration testing is of four types:-

1) Top-down 2) Bottom-up 3) sandwich 4) Big-Bang.

a) Black-Box testing :-

It is used for validation. In this we ignore internal working mechanisms and focus on what is the output?

B) White-Box testing:-

It is used for verification. In this we focus on internal mechanism i.e. how the output is achieved.

3) System Testing :-

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input & output without focusing on internal working.

In this, we have security testing, recovery testing, stress testing and performance testing. e.g.

This include functional as well as non-functional testing.

Page :
Date : / /

Example..
Write a prg to calculate the square of a no. in the range 1-100.

```
#include<stdio.h>
int main()
{
    int n,res;
    printf("Enter a no:");
    scanf("%d",&n);
    if(n>=1 && n<=100)
    {
        res=n*n;
        printf("\nsquare of %d is %d \n",n,res);
    }
    else if(n<=0||n>100)
        printf("Beyond the range");
    return 0;
}
```

Output :-

Inputs	Outputs
I1 : -2	O1 : Beyond the range
I2 : 0	O2 : Beyond the range
I3 : 1	O3 : Square of 1 is 1.
I4 : 100	O4 : Square of 100 is 10000.
I5 : 101	O5 : Beyond the range.
I6 : 4	O6 : square of 4 is 16.
I7 : 62	O7 : square of 62 is 3844.

Test Cases -
T1 : {I1, O1}
T2 : {I2, O2}

Page :
Date : / /

T3 : {I3, O3}
T4 : {I4, O4}
T5 : {I5, O5}
T6 : {I6, O6}
T7 : {I7, O7}

* Design a test suite for the following problem:
The absolute begineers Inc seems to have been fascinated by our work. Recently they have entrusted you with a task of writing a web-based mathematical s/w. As part of this software, your team mate has written a small module, which computes area of simple geometric shapes. A portion of the module is shown below.

1) function square(side) { return side*side }
2) function rectangle (side1, side2) { return side1*side2; }
3) function circle (rad) { return Math.PI * rad * rad }.
4) function right_triangle (base, hgt) { return 1/2 * base * hgt }

* Learning objectives:-
1) Get familiarized with unit testing.
2) Verify implementation of functional requirement by writing test cases.
3) Analyse returns results of testing to ascertain the current state of project.

* Limitations:-
This workspace attempts to provide a very simple version of a testing framework. Real life testing frameworks are much more extensive

and provide a lot of options like creating test cases from user requirements, automatic reporting of bug when a test case fails, and so on. Neverthless, this workspace is expected to make a student familiar to testing and some of its templates and reports

Conclusion :-

In this practical, we learnt about software testing and its different techniques used for testing a software and we also designed unit test case that verified functionality and located the bugs.