Jiwaji University
SOS in Computer Science and Applications
B.C.A. (VI Semester)
Paper-604 Software Testing
Unit 2

Topic:

Other Black Box Test Design Approaches

Compatibility Testing?

By Arun Singh

Other Black Box Test Design Approaches

There are alternative methods to equivalence class partitioning/boundary value analysis that a tester can use to design test cases based on the functional specification for the software to be tested. Among these are cause and effect graphing, state transition testing, and error guessing.

Cause - and - Effect Graphing

Cause Effect Graph is a black box testing technique that graphically illustrates the relationship between a given outcome and all the factors that influence the outcome.

It is also known as Ishikawa diagram as it was invented by Kaoru Ishikawa or fish bone diagram because of the way it looks.

Circumstances - under which Cause-Effect Diagram used

- To Identify the possible root causes, the reasons for a specific effect, problem, or outcome.
- To Relate the interactions of the system among the factors affecting a particular process or effect.
- To Analyze the existing problems so that corrective action can be taken at the earliest.

Benefits :

- It Helps us to determine the root causes of a problem or quality using a structured approach.
- It Uses an orderly, easy-to-read format to diagram cause-and-effect relationships.
- It Indicates possible causes of variation in a process.
- It Identifies areas, where data should be collected for further study.
- It Encourages team participation and utilizes the team knowledge of the process.
- It Increases knowledge of the process by helping everyone to learn more about the factors at work and how they relate.

Steps for drawing cause-Effect Diagram:

Step 1 : Identify and Define the Effect

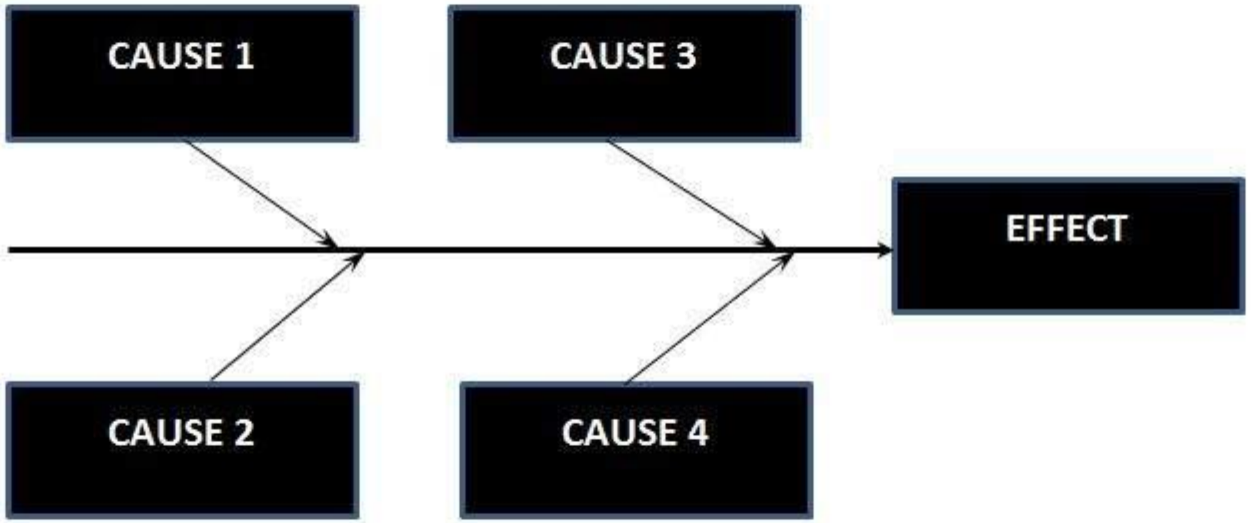
Step 2 : Fill in the Effect Box and Draw the Spine

Step 3: Identify the main causes contributing to the effect being studied.

Step 4 : For each major branch, identify other specific factors which may be the causes of the EFFECT.

Step 5 : Categorize relative causes and provide detailed levels of causes.

Cause Effect - Flow Diagram



State transition testing

Objectives of State Transition Testing:

The objective of State Transition testing is:

To test the behavior of the system under varying input.

To test the dependency on the values in the past.

To test the change in transition state of the application.

To test the performance of the system.

Transition States:

Change Mode:

When this mode is activated then the display mode moves from TIME to DATE.

Reset:

When the display mode is TIME or DATE, then reset mode sets them to ALTER TIME or ALTER DATE respectively.

Time Set:

When this mode is activated, display mode changes from ALTER TIME to TIME.

Date Set:

When this mode is activated, display mode changes from ALTER DATE to DATE.

State Transition Diagram:

State Transition Diagram shows how the state of the system changes on certain inputs.

It has four main components:

States

Transition

Events

Actions

Advantages of State Transition Testing:

State transition testing helps in understanding the behavior of the system.

State transition testing gives the proper representation of the system behavior.

State transition testing covers all the conditions.

Disadvantages of State Transition Testing:

State transition testing can not be performed everywhere.

State transition testing is not always reliable.

Error Guessing Technique

The test cases for finding issues in the software are written based on the prior testing experience with similar applications. So, the scope of the test cases would generally depend upon the kind of testing Test Analyst was involved in the past. The Error Guessing technique does not follow any specific rules.

For Example, if the Analyst guesses that the login page is error-prone, then the testers will write detailed test cases concentrating on the login page. Testers can think of a variety of combinations of data to test the login page.

To design test cases based on the Error Guessing technique, the Analyst can use past experiences to identify the conditions.

This technique can be used at any level of testing and for testing the common mistakes like:

Divide by zero

Entering blank spaces in the text fields

Pressing the submit button without entering values.

Uploading files exceeding maximum limits.

Null pointer exception.

Invalid parameters

The achievement rate of this technique does mainly depends upon the ability of testers.

Purpose Of Error Guessing In Software Testing

The main purpose of this technique is to guess possible bugs in the areas where formal testing would not work.
It should obtain an all-inclusive set of testing without any skipped areas, and without creating redundant tests.
This technique compensates for the characteristic incompleteness of Boundary Value Analysis and Equivalence Partitioning techniques.

Factors Used To Guess The Errors

Error Guessing technique requires skilled and experienced tester. It is mainly based on intuition and experience**.**

Following factors can be used to guess the errors:

Lessons learned from past releases
Tester’s intuition
Historical learning
Previous defects
Production tickets
Review checklist
Application UI
Previous test results
Risk reports of the application
Variety of data used for testing.
General testing rules
Knowledge about AUT

Error Guessing Example

Suppose there is a requirement stating that the mobile number should be numeric and not less than 10 characters. And, the software application has a mobile no. field.

Now, below are the Error Guessing technique:

What will be the result if the mobile no. is left blank?

What will be the result if any character other than a numeral is entered?

What will be the result if less than 10 numerals are entered?

Advantages of Error Guessing technique

Proves to be very effective when used in combination with other formal testing techniques.

It uncovers those defects which would otherwise be not possible to find out, through formal testing. Thus, the experience of the tester saves a lot of time and effort.

Error guessing supplements the formal test design techniques.

Very helpful to guess problematic areas of the application.

Drawbacks of Error Guessing technique

The focal shortcoming of this technique is that it is person dependent and thus the experience of the tester controls the quality of test cases.

It also cannot guarantee that the software has reached the expected quality benchmark.

Only experienced testers can perform this testing. You can’t get it done by freshers .

What is Compatibility?
Compatibility is nothing but the capability of existing or living together. In normal life, Oil is not compatible with water, but milk can be easily combined with water.

What is Compatibility Testing?
Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.
Compatibility Testing is a type of Non-functional testing

Types of Compatibility Tests
Let's look into compatibility testing types

Hardware: It checks software to be compatible with different hardware configurations.
Operating Systems: It checks your software to be compatible with different Operating Systems like Windows, Unix, Mac OS etc.
Software: It checks your developed software to be compatible with other software. For example, MS Word application should be compatible with other software like MS Outlook, MS Excel, VBA etc.
Network: Evaluation of performance of a system in a network with varying parameters such as Bandwidth, Operating speed, Capacity. It also checks application in different networks with all parameters mentioned earlier.
Browser: It checks the compatibility of your website with different browsers like Firefox, Google Chrome, Internet Explorer etc.
Devices: It checks compatibility of your software with different devices like USB port Devices, Printers and Scanners, Other media devices and Blue tooth.
Mobile: Checking your software is compatible with mobile platforms like Android, iOS etc.
Versions of the software: It is verifying your software application to be compatible with different versions of the software. For instance checking your Microsoft Word to be compatible with Windows 7, Windows 7 SP1, Windows 7 SP2, Windows 7 SP3.

Backward compatibility Testing is to verify the behavior of the developed hardware/software with the **older versions** of the hardware/software.

Forward compatibility Testing is to verify the behavior of the developed hardware/software with the **newer versions** of the hardware/software.

Tools for Compatibility Testing

1.Browser Stack - Browser Compatibility Testing: This tool helps a Software engineer to check application in different browsers.

2.Virtual Desktops - Operating System Compatibility: This is used to run the applications in multiple operating systems as virtual machines. n Number of systems can be connected and compare the results.

How to do Compatibility Testing

1.The initial phase of compatibility testing is to define the set of environments or platforms the application is expected to work on.

2.The tester should have enough knowledge of the platforms/software/hardware to understand the expected application behavior under different configurations.

3.The environment needs to be set-up for testing with different platforms, devices, networks to check whether your application runs well under different configurations.

4.Report the bugs. Fix the defects. Re-test to confirm Defect fixing.

Conclusion:

The most important use of compatibility testing is to ensure whether developed software works under different configurations (as stated in requirements documentation). This testing is necessary to check whether the application is compatible with the client's environment.