**Hot Fix and Defect Leakage:**

**Hot Fix:** It’s a process of fixing the defect asap (as soon as possible)

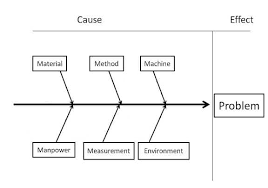
**Defect Leakage:** The test engineers miss the defect and bcoz of that, the defect is found in production server is known as defect leakage.

We cannot give 100% assurance to the client that the same defect will not occur again. So to overcome this problem we go for Route Cause Analysis (RCA)

**How to stop the defect leakage next time?**

Ans: We make use of a concept called (RCA)

**Route Cause Analysis (RCA):**



Here we try to identify the source which is causing the problem and we provide the solution at the source level.

Providing the solution at the source level will ensure that the same problem/defect will not occur next time

In order to do RCA we make use of FISH BONE diagram or CAUSE EFFECT diagram or ISHIKAWA diagram

**Exploratory Testing:**

When there is no Requirement Document (RD) or RD is there but not understandable or there is no time to read the RD, then we will explore the application, understand the application and then we come up with the scenarios and test the application with these scenarios.

Alternate Definition: Exploratory Testing is a simultaneous process of learning and testing the application when there is no Requirement Document (RD) or RD is there but not understandable or there is no time to read the RD.

**Disadvantages:**

1. Sometimes we may consider the defect as a function or vice versa.

**Pesticide Paradox:** When you execute the same test cases repeatedly on the application, the application will build the immunity towards these test cases bcoz of which some of the defects in the application will not be discovered easily.

If you test the application in the same manner again and again, you might miss to identify some defects.

**Functional and Non-functional testing:**

**Functional Testing:** Black box test engineers look at the functional requirements and test the functionality of each and every component.

**Non-functional Testing:** Here performance test engineers look at the non-functional requirements and tests the application. They will verify the stability and response time of an application

e.g.: Performance Testing comes under Non-functional testing

**Adhoc Testing(Monkey/Random Testing):**

In Adhoc testing, we come up with creative scenarios and then we test the application with these creative scenarios.

* Adhoc testing is done when you know the application very well.
* Exploratory testing is done when you don’t know the application.
* Adhoc testing is not mandatory. After performing all types of testing, if you do not find many number of defects or if you are not satisfied with the number of defects found or if you have enough time then only perform adhoc testing.

In adhoc testing, do you write test cases and then test the application?

Ans: No, we will test the application directly but if we find defects then only we create test cases for the defects found.

**Globalization Testing:**

Whenever the application is developed in multiple languages, we will call that as **globalization development.**

Testing the application which is developed in multiple languages is called as **globalization testing.**

**Types of globalization:**

**1.Internationallization[I18N]:** Here we check whether the right content is displayed or not in the application.

E.g.: If the application is developed in Chinese, then all the content of that website/application should be in Chinese language only.

**2.Localization[L10N]:** Here we test the website or application based on the country’s standard.

E.g.: Let’s consider date field India is DD:MM:YYYY and in US, its MM:DD:YYYY. If the application is accessed in India, the date field should allow the date format in DD:MM:YYYY and if the same application is accessed in US, the date field should allow the date format in MM:DD:YYYY

**Types of Application:**

1. **Web Application:** Which opens in the browser. E.g.: Facebook, Gmail…etc.
2. **Client/server application:** It’s the one which requires a dedicated client software to access the application

e.g.: Yahoo messenger, Tata Docomo 3G...etc.

**There are 2 categories:**

1. **Thin Client:** Here the major process happens at server side.
2. **Thick Client:** Here the major process happens at client side.
3. **Standalone application:** It’s the application which do not require internet connection i.e. which runs locally on your personal computer.

E.g.: Excel, adobe reader, vlc media player…etc.

**Compatibility Testing:**

Testing the application on different software and hardware platforms is called as compatibility testing.

Note: Testing on different browsers or different versions of the browsers and different operation systems (windows 10, windows 7, Linux…etc.)

Compatibility testing is not mandatory, it depends on the requirements.

**Compatibility Testing Defects:**

* Text is not aligned properly in the web page.
* Some of the objects’ functionality is broken.
* Color of the web page looks different in particular browsers.
* The font size of the text has changed.
* Some of the images are not displaying properly.

**How do you perform compatibility testing on standalone application?**

Testing the application on both software and hardware platform.

**Test case:** It is a detailed step by step procedure to test the functionality of the features/components in an application.

**Test case Design Techniques:**

1.Error Guessing

2.Equivalance Partitioning

3.Boundary Value Analysis

**1.Error Guessing:** Give the inputs such that the application throws an error (give invalid inputs).

**2.Equivalance Partitioning:**

**Type-1:** If the input is a range of value, then design the test case by selecting one valid and two invalid values

E.g.: If (amount>=200 and amount<=6000)

{

Transfer the amount;

}

Else

{

Throw an error message;

}

**Alternate Definition:** If the input is a range of values then divide the values into blocks and design the test case by selecting one valid and two invalid values

E.g.: If the input range for amount field is 200 to 6000 then we will divide this into blocks as,

{200, 1000, 2000, 3000, 4000, 5000, 6000} now choose one valid and two invalid inputs.

**Type-2:** If the input is a set of values, then design the test case by selecting one valid and one invalid values.

E.g.:

If amount field has to accept these values {100, 200, 300, 400, 500,1000, 2000, 10000}

If product id is {430,254,100,211,455}

**Tyep-3:** If the input is a Boolean value, then design the test case for both true and false conditions.

E.g.: Gender field (Male and Female)

**3.Boundary Value Analysis :**

If the input is a range of value between A and B then design the test case by selecting the values such as A, A+1, A-1 and B, B+1, B-1.

**E.g.:** If (amount>=200 and amount<=6000), here A=200 and B=6000

{

Transfer the amount;

}

Else

{

Throw an error message;

}

**Advantages of Test Cases:**

* This could be a good learning source for new joiners.
* For better consistency.
* For better coverage.

**Test case Design Template:**