**ACCEPTANCE TESTING**

**Acceptance testing is done by end users. Here, they use the s/w for the business for a particular period of time and check whether the s/w can handle all kinds of real-time business scenarios / situations.**

For Acceptance testing, let us consider the example shown below.

**WIPRO**

**FED - EX**

Fed-ex with its requirements asks Wipro to develop the s/w and Wipro agrees to give the s/w in 2 releases like below,

**25crores 18crores**

**Jan 2010 Sept 2010 Sept 2010 Feb 2011**

On September 8th, test manager tells the project manager that there is a critical bug in the application which will take another 5days to fix it.

But the project manager says you just deliver the application and by the time they implement in Fed-ex, it takes another 25days so we can fix the bugs or otherwise we will have to pay the penalty for each day after the said release day. **Is this the real scenario ? – No**. Then what happens, we will see now in 3 cases which really and who really does the acceptance testing.

**CASE 1 :-** here, we will discuss how the acceptance testing is done or how the test engineer testing becomes the acceptance testing here.

**Product to Customer**

**Functional Integration System**

**Testing Testing Testing**

**CODING**

**WBT**

Usually, the actual flow of testing will be like above. But, here a small difference we see where the system testing or end-to-end testing becomes the acceptance testing. To understand this, follow the sequence below,

Fed-ex gives the requirements and Wipro develops the s/w and do all testing and gives it to Fed-ex

Are the Fed-ex going to use the s/w as soon as they get from Wipro ? – NO, certainly not. Then what do they do ? – Observe,

Fed-ex, they have some group of Test Engineers and after they get the s/w, this team starts testing it. So, now we can understand that though the test engineer do the testing but it is done at customer level. This end-to-end testing is called ACCEPTANCE TESTING.

**The difference between Wipro test engineers and Fed-ex test engineers are,**

* The Wipro testing do Functional Testing, Integration Testing and System testing. But at Fed-ex, the testing team do only end-to-end testing / system testing.

The difference between end-to-end testing of Wipro and Fed-ex is,

* Fed-ex engineer is a domain expert
* Fed-ex engineer understands the business well
* Fed-ex engineer tests for real time data
* Fed-ex engineer is the one who gave the requirements.

To understand this, we see the example below. If the application format is like below,

**PARCEL 1 Docket ID Produced**

**CANCEL**

**SEND**

**NEW PARCEL**

**USERNAME**

**PASSWORD**

**FROM Address**

**TO Address**

**WEIGHT**

**NEW PARCEL**

**CHANGE ADDRESS**

**DISPATCH**

**CANCEL**

**….**

**…**

**LOGOUT**

In the above example, after the product is given to Fed-Ex Test Engineers, they do testing and they know after the application has been filled above, it should produce an message saying “Parcel 1 Docket ID Produced”. If this is not happening, they give back the application for fixing bugs. Now, the Fed-Ex checks whether this feature is there or not in the requirement. If it is there and Wipro have not done fix it, then Penalty Counts for Wipro from that day, whereas the TE at Wipro will not be knowing this and thus arises the difference in testing at Wipro and Fed-Ex.

Thus, the TE become END-USERS here and this testing is known as Acceptance Testing.

**CASE 2 :-**

In this case, we see how the employees are becoming end-users and do acceptance testing.

**If Bug is there, send it back to Wipro Employees doing dummy implementation or Testing**

**S/w given to Fed-Ex**

The s/w is developed and tested at Wipro’s place and then sent to Fed-ex. At Fed-Ex, they have less TEs and so it is not possible for them to do Acceptance testing. So, out of 400 employees of Fed-ex, Fed-ex gives the s/w to 40 employees and installs the product at their systems and asks them to start using the s/w and come up with bugs or issues.

Now, the 40 employees, they do dummy implementation (i.e, they implement the data into the application and also have the data written manually). Now, the employee here becomes the end-users and come up with bugs and issues when using the s/w.

These issues are verified against requirements and now penalty is charged for Wipro ( sometimes, penalty is charged on an hourly basis ).

If the bug found is not as per requirement, then Fed-Ex can go for CR or RFE.

CR – Change Request – i.e, if the requirement has not been specified properly, then Fed-Ex gives the correct requirement and requests for change.

RFE – Request For Enhancement – if Fed-Ex feels that a particular module can be enhanced and developed in a better way, then they can send the CRS as RFE and Wipro goes on to make the necessary changes.

Thus, **Acceptance Testing can also be defined as – *end-to-end testing done by engineers sitting in customer’s place. Here, they take real time scenarios and check whether the s/w works or not. Here also, we are able to take real time business scenarios because the end-users know how the business flow works.***

**We are getting more and more builds for Acceptance Testing means,**

* The product quality which is delivered to customers is not good. Development and testing both are not good.
* After receiving the s/w, customer is getting more and more ideas, so he is asking for more and more changes
* The requirement which was given in the beginning is not clear.

**CASE 3 :-**

Here, the Fed-ex customers become the end users.

Here, the s/w is developed and tested and implemented at Fed-ex production servers and thousands of users start using the s/w. This comprises the **1st release.** When using the s/w, Fed-ex comes up with more number of features and enhancements and sends the CRS to Wipro who make the additional changes and modules and give it to Fed-ex.

Thus, what is happening here is – the requirements are collected by Fed-ex from customers and end-users and then the s/w is developed.

**The number of cycles depends on,**

* Number of features
* Complexity of features
* How new features affect old features

**Hot fix –** in production environment, whenever the client finds critical bugs – developers fix the bugs – small team of TEs test it – reinstall the s/w – client starts using the new s/w. This entire process is known as **Hot fix**. It takes few hours to 1day.

**For ex,** if the login feature itself is not working at the production environment, then the client immediately sends it for fixing which is done asap.

**SLA – Service Level Agreement**

**Interim Release –** ( short release ).

Between 2 major releases – there is a short release of enhancements – this comes up when the client requires a small bunch of features very urgently. Out of 70developers, around 10come out and out of 30 TEs, around 3 come out – they develop and test the s/w – client does 1 short round of Acceptance testing – before adding it to the production environment – this interim could take just around 15days to 1month.

**\*\*\* SMOKE TESTING or SANITY TESTING or DRY RUN or SKIM TESTING or BUILD VERIFICATION TESTING \*\*\*** (*Very very important interview question)*

**Testing the basic or critical features of an application before doing thorough testing or rigorous testing is called as smoke testing.**

It is also called Build Verification Testing – because we check whether the build is broken or not.

Whenever a new build comes in, we always start with smoke testing, because for every new build – there might be some changes which might have broken a major feature ( fixing the bug or adding a new feature could have affected a major portion of the original software).

In smoke testing, we do only positive testing – i.e, we enter only valid data and not invalid data.

Do we have separate testing (or) do we have to do it in between FT, IT, ST ? Then, where actually do we do smoke testing? Observe,

**AT**

**FT IT ST**

**CODING**

**WBT**

**In Between Smoke Testing ? – NO In Between Smoke Testing? – NO**

**Production**

From the above diagram, it may be confusing when we actually do smoke testing

Now, we have to understand that smoke testing is done in all testing before proceeding deep into the testing we do.

The below example will make us understand better when to do smoke testing,

Developers develop application and gives it for testing. The testing team will start with FT. suppose we assume that 5days we are given for FT. on the 1st day, we check one module and later 2nd day we go for another module. On the 5th day, we find a critical bug, when it is given to the developer – he says it will take another 3days to fix it. Then we have to stretch the release date to extra 3days.

Then how do we overcome this ? – Observe how smoke testing works here. In the above scenario, instead of testing module by module deeply and come up with critical bug at the end, it is better to do smoke testing before we go for deep testing i.e, in each module – we have to test for basic (or) critical feature and

**REQUIREMENTS Given 5days for Functional Testing**

**Functional Integration System**

**Testing Testing Testing**

**...**

**…**

**Developers develop application**

**1 2 3 4 5**

**CRITICAL BUG Send it to developer Fix it in 3 days**

then proceed for deep testing. The scenario will be like this as shown in the figure below,

**Functional Testing Integration Testing System**

**Testing**

**Developers develop**

**the application and send it to testing team**

**Smoke Testing for FT Smoke Testing for IT Smoke Testing for System Testing**

**Check for basic (OR) critical features (SMOKE TESTING) Find critical bug**

**Development team**

**Fix the bug ( so that the release date will not be**

**stretched )**

Question arises – how do we know which is the critical feature? – we will come to know which is the critical feature or basic feature when we proceed with the testing.

**Smoke Testing in Acceptance Testing**

**Acceptance**

**Testing**

**System**

**Testing**

**Copy the file from Testing server to End-User server**

**Smoke testing here involves whether the entire product works fine and goes for Acceptance Testing**

Let us now consider **3cases** where in which we see where we do smoke testing in different kinds of testing.

***CASE 1***

**FT IT ST**

**CODING**

**WBT**

**GIVE FOR TESTING Smoke Testing Major Bug**

**Back to development team to fix the bug**

Major bug in the initial stages is an insult to the developer.

Sometimes, the below procedure is also followed,

**CODING**

**WBT Smoke Testing in WBT Testing Team**

If this is done, then the testing team need not do smoke testing as the bugs are already fixed in WBT. But depending upon the project or the organization, normally this is not followed.

***CASE 2***

Production Team

**Acceptance**

**Testing**

**SMOKE TESTING – Whether Installation is working fine or not**

In the above scenario, smoke testing in production team means after we do Acceptance Testing, Smoke Testing involves whether the s/w developed is installed fine or not.

**Important Points to Remember**

* When we are doing smoke testing, we do only positive testing (only valid data is entered)
* Here, we test only basic or critical features
* Here, we take basic features and test for important scenarios
* Whenever the build comes to the customer, before the customer / client does Acceptance Testing, he also does Smoke Testing before doing Acceptance Testing
* When the product is installed in production, we do quick smoke testing to ensure product is installed properly.

**Why we do Smoke testing ?**

* Just to ensure that product is testable
* Do smoke testing in the beginning – catch bugs in basic features – send it to development team so that development team will have sufficient time to fix it.
* Just to ensure that product is installed properly.

In early stages of product development, doing smoke testing fetches more number of bugs. But, in later stages of product development, if you do smoke testing – the number of bugs that you are going to catch in smoke testing will be very less. Thus, gradually the effort spent on smoke testing is less.

**Note :-** *only for information purpose (NOT FOR STUDYING)*

Smoke testing is classified into two types,

* ***Formal Smoke Testing*** – the development team sends the s/w to the test lead. The test lead then instructs the testing team to do smoke testing and send report after smoke testing. Once, the testing team is done with smoke testing, they send the smoke testing report to the Test lead.
* ***Informal Smoke Testing*** – here, the test lead says the product is ready and to start testing. He does not specify to do smoke testing. But, still the testing team start testing the product by doing smoke testing.

**INTERVIEW QUESTIONS**

***1) Difference between Smoke Testing and Sanity Testing and Dry Run***

***Ans) Sanity Testing***

* ***Narrow and deep testing. Unscripted***
* ***Take some very very important features and do deep testing***
* ***It is manually done***

***Smoke Testing***

* ***Scripted. Shallow and wide testing***
* ***Take all important features and do high-level testing***
* ***Build comes – write automation scripts and run the script. Thus test done automatically.***

***Dry Run - A dry run is a testing process where the effects of a possible failure are intentionally mitigated. For example, an aerospace company may conduct a "dry run" of a takeoff using a new aircraft on a runway before the first test flight.***

***If he still expects more differences – then just tell – “* for study purposes, i had visited wikipedia website and also www.allinterview.com . So, i know this answer. Can u please tell me which website i should visit to get to know the exact answer. That’s how i could answer this question “.**