Difference of static and dynamic testing

Static testing :- means verify the documentation / mainly focused on the project documentation.

Static testing :- nothing but testing the documentation whether which is written in correct or not

static testing can be done by using Review, Walkthrough, Inspection

----Technique----

1. Review :- Read complete document whether it is complete or not----// only other will be there

Conductness on document to ensure correctness or completeness.

Types of

1. Requirement Review

2. Design Review---conduct on design document

3. Code Review

4. Test plan Review---even testing part document

5. Test case Review

Anybody can do the review--manager, developer, tester, business analyst

2. Walkthrough

It is basically a informal Review.// there will not be any proper plan / any proper schedule--//means two or more people will be there

Conduct walkthrough at any time any place

Author reads the document or code and discuss peers.

It is not preplanned and can be done whenever required.

Also walkthrough does not have minutes of the meet.

3. Inspection --- More formal --means it’s a preplanned--send invite to team member

It s a most formal review

In which at least 3-8 people will sit in the meeting 1-reader, 2-Writter, 3-Moderatore plus concerned.

Inspection will have proper schedule which will be intimated via email to the concerned developer/ tester.

Writer :- Note down all the clarification and issue everything

Moderator :- it just like a Anker mediator between other and rest of team

Dynamic testing:-

Mainly focus on the software--> Mainly focus on the software which we have developed

Actual software we are going to verify or validate.

----Technique----

1. Unit testing :- Developer will concentrate on testing the code what they have written

2. Integration testing :- They will Integrate multiple module and multiple component in the software and they will check the data flow between the component

3. System testing :- Normally tester doing this testing in this testing lot of testing will be conducted during system testing security testing , performance testing, UI testing

Tester will be doing this testing.

4. UAT Testing :- done by the tester along with the customer.(User Acceptance Testing )

They will set up the exact environment where the customer will go to work install the software they will test the software customer point of view. Also execute some of the test cases in the software.

Start dynamic testing :- coding/software part will be ready

Concept of QA, QC, QE

QA :- Quality Assurance :- High level management people will comes under the QA

Entire process design by this people. This people make shore rest of the people following

Process is properly or not.

QA involved every stages in SDLC

Throughout development process involved in QA

QC :- Quality Control

It is a actual testing of software they are only involved in testing phase in SDLC

Test the software whether it is working according to the customer requirement or not

QC is a subset part of QA

QE :- Quality Engineer

Quality Assurance

Quality Control

It is a process related testing

It is a actual testing of the software

P-People -->QC--> Talking about people (Tester)

P-Process--->QA – always define the process and talk about the process

P-Product--->

Focuses on building in quality ---> to follow certain process properly before release the software.

Focuses on testing for quality ---> QC is related to testing process

QA is preventing defect --> preventing means define the process

QC is detecting defect-->

While testing the software find the defect and report to the developer

It is a process oriented

It is a product oriented

QA for entire software development life cycle

QC for testing part in SDLC.

Testing : - Mainly focus on the quality of the products

They will write the code to developed the software

QE means automation tester ---> They also write code for testing purpose to test the software comes under Quality engineer

QE is a team which contain a automation tester

Level of Testing -----> 4 level of testing-----technique of dynamic testing

1. Unit testing

2. Integration Testing

3. System Testing

4. User Acceptance Testing

1. Unit testing :-

It is done by the developer at code level, test the small module in the application

is component level testing . Small piece of code will be tested

small component is called as unit. It comes under white box testing .

A unit testing is a single component or module of a software.

Unit testing conduct on a single program or single module

Unit testing is a white box testing technique

Unit testing is conducted by developer

-------Unit testing technique ----

Basic path testing --- Test the program each and every line should be covered

Control structure testing

Conditional coverage---Developer will verify the condition in positive and negative input

Loop coverage ---repeat some statement multiple times we have to keep in loop statement

Mutation Testing -- It is a repetition testing –Testing the code with multiple side of data

2. Integration Testing

While combine multiple module and multiple unit in a single unit and then they will

check the data flow and communication between those modules

It comes under white box testing.

--- Integration Testing-----

Integration Testing perform between 2 or more module.

Integration Testing focuses on checking data communication between multiple modules

Integration Testing is white box testing technique.

Integration Testing done by the tester at a UI level

In Integration Testing at least 2 module will be integrated

I/O----A----O/P---------I/O-----B-----O/P

Output of A is the input of B, Input of B is depend on Output of A.—checking the data

flow between two or more module

Every software every application will be having different module

Types of module 1 login, 2 composing emails, 3 deleting, 4 outbox, 5. Send emails

Types of Integration Testing

1. Incremental Testing:- Incrementally adding the module and testing the data flow between the modules
2. Two approaches

Top down bottom up

Top down Integration Testing : - Ensure that the module is added as a child of the previous module.

(Compose email is a parent module) it’s a top module, (send email is a child module, deleted email is a another child module) all are child module

Bottom up Integration Testing :- Ensure that the module is added as a child of the parent module of the previous module.

Sandwich/ hybrid approach :- It is a combination of top down & bottom up approach

2. Non Incremental Integration Testing

Adding all the modules in single shot, and test the data flow between modules

Drawback

We might miss of the data flow between some of the modules.

If you find any defect we can’t understand the root cause of defect.

3. System Testing

Testing the overall functionality of application whether it is working or not each and

Every requirement which are mention in the software is working or not

This is a actual area testers will be evolved

------- System Testing------

Testing over all functionality of the application with respective client requirements

It is black box testing technique

This testing is conducted by testing team

After completion of component and integration level testing’s we start system testing

Before conducting system testing we should know the customer requirements

----System testing focuses on below aspects-----

User interface testing (GUI)--- Amazon it is a UI testing

Functional testing ----Test functionality of the application –(login, check balance, withdraw balance, transfer money, update adhar card , pan card , mobile)—this all are functionality check those functionality are properly working or not

Non functional testing –security, performance, installation testing , compatibility, load , stress testing --

Once the functionality is stable then start the non functional testing

Performance means :- speed of the application

Usability testing

User help, user manual contain how to use a product to user, customer

What is GUI testing

GUI testing is process of testing the user of an application.

Graphical user interface includes all the elements such as menus, checkbox, buttons, colors, fonts, sizes, icons, content and images.

GUI testing mainly focus on front end part / Testing the UI of the application

User always interact with the application through front end, and actual operation will be done at back end as a user I cannot seen what changes in back end

Usability Testing :-

During this testing validates application provided context sensitive help or not to the user

Checks how easily the end users are able to understand and operate the application is called as usability testing

Every application and software must provide the help to customer / user

Functional testing :-

Functionality is nothing but the behavior of application

1 . Object properties testing

Check the properties of objects presents on the application

Ex. Enable, disable, visible, focus…….

2. Database/ Back end Testing

Mainly focus on DML (data manipulation language) operation

Database testing are of two things --- black box and white box testing

Data base testing is one of the example of gray box testing because we do black box testing and we also do white box testing there

Table & column level validations (column type, column length, number of columns )

Relation between the tables (normalization)

Functions

Procedures

Triggers

Indexes, View etc…..

Error handling testing

Tester verify the error messages while performing incorrect actions on the application

Error messages should be readable

User understandable / simple language

Calculation and manipulation testing

Tester should verify the calculation working is properly or not

What is cookies and session ?---- it is only web based application

Cookies

Temporary files are created by browser while browsing the pages through internet

Session

Session will be created on server sides

Open the application the time slot will be started at server side. Session will be expire after some time (if your idle for some time )

Non functional testing

Customer expectations and requirements both are different

Non functional testing will performed by separate skill set is required

Normal tester and functional tester cannot be done non functional testing

Requirement is nothing but the some feature to be there in the application

Expectation :-

My application work on different environment

My application should respond very faster

My application should be safe from the anauthorized users or hackers

My application should able to install in multiple environment

Performance testing :- speed of the application

Load Testing :- Gradually increasing the load(multiple member) of the application slowly, then check the speed of the application.

Stress Testing :- Suddenly increase / decrease the size with large number of load on the application and check the speed of the application

Volume testing :- How much data is able to handle by the application

Security testing – How secure our application

Authentication – Users are valid or not

Authorization / Access control – Permission of the valid user we are going to verify

Recovery testing

Whether your application recovering the data or not

Check the system change abnormal to normal

Normal to abnormal ---abnormal to normal it is a recovery mechanism

Compatibility testing

Application should be compatible with different environment

Forward Compatibility--

Backward Compatibility –

Hardware Compatibility—Configuration Testing // our software is working multiple platform or not it support (Unix , windows, Linux) or not

It is working or not on 4GB, 8G RAM or not

Installation testing

Just we have to test installation process

Check the screen are clear to understand

Screens navigation

Simple or not

Un installation testing – It should remove all the files and folders related to that software

Sanitation and garbage testing

Garbage Testing—

If any application provides extra features and functionality then we consider them as a bug

We have to remove unwanted stuff/requirement from our software

|  |  |
| --- | --- |
| Functional Testing | Non Functional testing |
| Validates functionality of the software | Verify the performance, security , reliability of the software |
| Functionality describe what software does | Non functionality describe how software work |
| Concentrate on user requirement | Concentrate on users expectation |
| Functional testing takes place before non functional testing | Non functional testing performed after completion of functional testing |

4. User Acceptance Testing

After completion of system testing tester along with the user will conduct user

acceptance testing

White Box Testing :- Performed by developer / need to programming

Black Box Testing :- Performed by tester. / No need to programming / pass the input & check the output / no need to know the internal logic of program

Software testing terminology -----important to interview

1. Regression testing

Build contains a multiple module

Module no. 1 will change that should not impact of other module

Regression testing conducts on modified build to make sure that there will not be impact on existing functionality because of changes like adding / deleting / modifying features

Imp line---Existing functionality should not be broken because of new changes or bug fixes

---Unit Regression Testing----

Testing only the changes/modifications done by the developer

---Regional Regression Testing—

Testing the modified module along with the impacted modules

Impact analysis meeting conduct to identify impacted modules with QA & Dev

--Full Regression Testing—

Testing the main feature and remaining part of the application

Eg. Dev has done changes in many modules instead of identifying impacted modules, we perform one round of full regression.

Impact Analysis – we will identify the impacted modules because of new changes /modification/adding/ deleting all this reason some functionality will be impacted

Identify those functionality and test those functionality is called as regional regression testing

Re –Testing

Means we have to conduct the testing again and again. // executing test cases again and again

Verifying the functionality again and again in upcoming build.

Whenever the developer fix the bug. Tester will test the bug fix is called as re-testing

Tester closed the bug if it worked otherwise re-open and send to developer

To ensure that defect which were found and posted in the earlier build were fix or not in the current build

Ex. Build 1.0 was released . Test team found some defect (d\_1.0.1, d\_1.0.2, d\_1.0.3, d\_1.0.4 )

--Build 1.1 was released. Now testing the defects d\_1.0.1, d\_1.0.2, d\_1.0.3, d\_1.0.4 in this build is re- testing.

Smoke Testing Vs Sanity Testing

Smoke & sanity –both are fail then reject he build developer will create another build and they provide to tester

Very basic functionality we are going to test , don’t test the application in detail

Smoke – Build is (stable, installation, pages properly) or not & basic navigations working or not / basic level testing performing before testing the in detail testing , not focus on measure feature of application

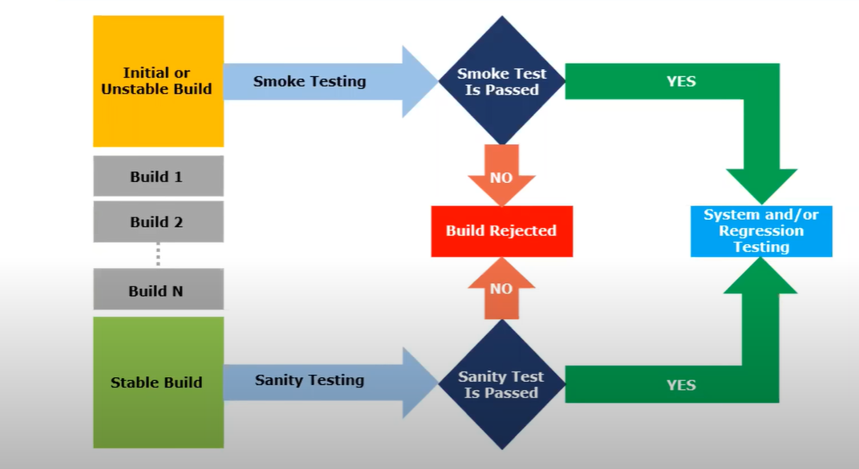
Basic installation successful but basic navigation will fail, some time URL will be ping

Once open the URL sometimes home page will not be displayed

Developer don’t test the functionality

Smoke Testing Vs Sanity Testing

|  |  |
| --- | --- |
| Smoke Testing | Sanity Testing |
| Smoke testing is done to make sure the build we received from development team is testable/ stable or not | Sanity testing is done during the release phase to check for the main functionalities of the application without going deeper |
| Smoke testing is performed by both developer and tester | Sanity testing is performed by only tester |
| Smoke testing build may be either stable or unstable | Sanity testing build is relatively stable |
| It is done on initial builds | It is done on stable builds |
| It is a part of basic testing | It is a part of regression testing |
| Usually it is done every time there is a new build release. | It is plan when there is no enough time to do in depth testing. |



Exploratory Testing

No documentation is available but the application is ready

We have to explore the application, Understand completely and test it

We do exploratory testing when the application is ready but there is no requirements.

Test engineer will do exploratory testing when there is no requirement

Drawbacks /Disadvantages

You might misunderstand any feature as a bug or any bug as a feature since you do not have any requirement.

Time consuming

If there is any bug in application, you will never know about it.

Adhoc Testing ---

Testing application randomly without any test cases or any business requirement document

Adhoc testing is an informal testing type with an aim to break the system

Tester should have knowledge of application even though he doesn’t have requirements and test cases

This testing is usually an unplanned activity

Break the application it is a fundamental job of tester .

In this 3 testing there is no any documentation FRS, BRS ,

|  |  |  |
| --- | --- | --- |
| Adhoc testing | Monkey testing | Exploratory testing |
| No documentation | No documentation | No documentation |
| No plan | No plan | No plan |
| Informal testing | Informal testing | Informal testing |
| Tester should know application functionality | Testers doesn’t know application functionality | Testers doesn’t know application functionality |
| Random testing | Random testing | Random testing |
| Intention is to break the application / find out corner defects | Intention is to break the application / find out corner defects | Intention is to learn or explore functionality of application |
| Any application | Gamming application | Any application which is new to tester |

Positive testing

Testing the application with valid input

It checks whether an application behaves as expected with positive input

Ex. Enter any number—99999---is called as positive test case

Negative testing

Testing the application with invalid input is called negative testing

It checks whether an application behaves as expected with negative input.

Ex. Enter any number – andfbjdf@@#$ ----should not accept any character or special symbol is called negative test case

Positive vs negative test cases

Requirement –

If a text box is listed as feature and in FRS it is mention as text box is accepts

6-20 character and only alphabets ----minimum 6 and maximum 20 character

Positive test cases

Text box accepts all alphabets

Text box accept 6 characters

Text box accept up to 20 char length

Text box accept any value in between 6 to 20 char length

Negative test cases

Text box should not accept less than 6 characters

Text box should not accept more than 20 characters

Text box should not accept special characters

Text box should not accept numerical

End –To-End Testing

Testing the overall functionality of the system including the data integration among all the module is called End-To-End testing

1. Login

2. ADD New Customer

3. Edit Customer

4. Delete Customer

5. Logout

Globalization and Localization testing

Globalization Testing

Perform to ensure the system or software application can run in any cultural or local environment

Different aspects of the software application are tested to ensure that it supports every language and different attributes

It test the different currency format, mobile number format and address format are supported by the application

Ex. Facebook , twitter support many of the language and it can be accessed by people of different countries. Hence it is a globalized product

Localized Testing

Performed to check system or software application for a specific geographical or cultural environment

Localized product only support the specific kind of language and is usable only in specific region

It test the different currency format, mobile number format and address format is working properly or not.

Ex. Baidu.com – it supports only a chainless language an can be access only by people of few countries. Hence it is a localized product

China is not used any globalized software or system

Test design technique -----important to ISTQB certification Exam

Used to reduce the data and increase the coverage

Main uses to prepare the test data which will covered each and every area in feature /functionality.

Advantages of test design technique

-----Reduce the test Data:--

-----More Coverage: ---

5 types Test design technique

1. Equivalence class partition (ECP)

2. Boundary value analysis BVA)

3. Decision making table

4. State transition

5. Error guessing

1. Equivalence class partition (ECP)

Partition data into various classes and we can select data according to class then test

It reduce the number of test cases and save the time for testing

Value Check

Classify/divide /partition ata in multiple classes

|  |
| --- |
|  |

Enter a number \* Allow digits from 1---2000

|  |
| --- |
|  |

name Allow only alphabets

Test data using ECP

A…Z----Valid -----ABXZ

a…z----Valid - ---- abca

Number---123---Invalid ---12344

Special characters—Invalid ---@#$a&\*#

Spaces---- 333----Invalid ----- ab g A bc

--Upper and lower case both are covered---

2. Boundary value analysis BVA)

Mainly focus on the boundary of the value

|  |
| --- |
|  |

Enter age Allow digit from 10 ----30

We will just test the boundaries not test the all functionality, we no need to verify all the values

Only 6 test cases are present in Boundary value analysis ---2 are invalid & 4 are valid

1. min-1=9—Invalid

2. min=10--- Valid

3. min+1=11----Valid

4. max=30--- Valid

5. max-1= 29 --- Valid

6.min+1=11----Invalid

Input domain testing

Value will be verified in the text box / input field

We use ECP and BVA

3. Decision making table

--We can reduce the data same time increase the coverage-----

---If we have more number of conditions/actions then we use decision table technique ---imp

Decision table is also called as cause effect table

This technique will be use if we have more conditions and corresponding actions

Decision table technique, we deal with combinations of inputs

To identify the test cases with decision table we consider conditions and actions

Take an example of transferring money to an account which is already approved and added

Here the condition to transfer money are

Account already approved

OTP (one time password) matched

Sufficient money in the account

Here the actions performed are

Transfer money

Show a message as insufficient money

Block the transaction in case of suspicious transaction

4. State transition

State transition technique changes in input conditions change the state of the Application

This testing technique allows the tester to test the behavior of an AUT.

The tester can perform this action by entering various input conditions in a sequence

State transition technique testing team provide positive as well as negative input test values for evaluating the system behavior.

5. Error guessing

Error guessing is one of the testing technique used to find the bug in software application based on testers prior and experience.

In error guessing we don’t follow any specific role

It depends on tester analytical skill and experience

-----Blindly check the application and test the error----imp point

Example

Submitting a form without entering values

Entering invalid values such as entering alphabets in the numeric field

STLC—Software testing life cycle----

It talks about only testing process.

--STLC is also part of SDLC---

1. Requirement Analysis

2. Test Planning

3. Test Design

4. Test Execution

5. Defect / bug reporting and tracking

6. Test closure

--STLC is a subset of SDLC process

---- Test Planning----we will prepare a small documents

Mainly focuses on three thing

What to test

How to test ----All kind of testing we have to test smoke, unit, sanity, integration

When to test ---we have to specify the time ---time line , schedule

Test estimation ----Will estimate the time for testing

Test engineer responsibility

Understand the requirement and write the test cases , what is the test cases

---use case is part of functional requirement

Use case--- means describe the requirement, clearly understand the functionality

Test plan content

Test plan is a document that describe the test scope, test strategy, objectives, schedule, deliverables and resources are required to perform testing for a software product.

Overview –what is a project and why we are preparing the test plan document and description

---Most imp in interview ---Scope— What to test what not to test –again depends on the requirement document

--inclusions

--test environment

--Exclusions

Test strategy –what kind of testing we are going the conduct --.smokes, sanity, regression, retesting, function testing this are the strategy –Like manual and Automation

Defect reporting processor---what we are going to report the bug and what is the exact process to follow to report the defect from tester to developer

Roles /responsibilities---Testing team clearly define the test plan document. Like test engineer, test lead, project manager everybody in team review the document all the part of roles and responsibilities.

Test schedule—when we have to conduct what kind of testing, clearly we have to specify the days also along with what kind of activity

Test deliverable – After completion of Each and every phase of testing we are going to deliver certain document. Test plan documents are deliverable, test case documents are deliverable defect report, test execution report is a deliverable

Pricing

Entry and exit criteria—when we have to start your testing and when we have to exist your testing. This criteria clearly mention in the test plan document

Suspension and resumption criteria ---sometime we will immediately stop testing suppose something not working, something broke

Tools—Excel document bug tracking tools, like automation testing tools, testing management tools

Risk and mitigation—Suppose while project will going on there will be some risk , risk can be come from any side

Mitigation in the since another plan team member they can leave from long time that is a risk for project , we can replace the person means mitigation. We have to also mentain the back up machine

Approval –who will approve the test plan, test cases, defect reporting, test case execution

We have all this specify clearly in test plan document

----Use case, Test scenario and Test case---very very important -----

Use Case—

-picture or data flow diagram by which we can understand the requirement more clear

---Also specify a FRS document along with a text--

---Project manager and product owner write the test cases, FRS document and requirement document they will create the use cases.---imp

Part of FRS document

- Describe the test requirement

-it contain Three items

1. Actor— Which is a user, which can be a single person, or group of people interacting with a process

2. Action ---Which is to reach the final outcome

3. Goal/outcome ---Which is the successful user outcome

Test Scenario

A possible area to be tested (what to test )

Depends/ based upon a use cases we will drive the test scenario

Tester create the test scenario and test cases

Test Case

Step by step action to be perform to validate functionality of AUT (How to test )

Test case contains test steps, expected result & actual result

Which contain multiple steps

By using test scenario we will write test cases

Use Case Vs Test Case

Use Case---Describe the functional requirements, prepared by business Analyst (BA)

Test Case---Describers the test steps/Procedure prepared by test engineer.

Test environment

Test environment is a platform specially build for test case execution on the software product

It is created by integrating the required software and hardware along with proper network configuration

Test environment simulates production / real time environment

Another name of test environment is test bed

--Test Execution

During this phase test team will carry out the testing based on the test plans and test case prepared

Entry Criteria----Test cases , test data and Test plan

---Activities

Test cases are executed based on the test plan

Status of test cases are marked like passed, failed, blocked, run, other.

Documentation of test result or long defects for test case is done

All blocked and failed test cases are assigned bug ids

Retesting once the defects are fixed

Defects are traced file are closure

------Deliverable

Provides the defect and test case execution report with completed result

---Guidelines for Test Case Execution

The build being deployed to the QA environment it is the most important part of test case execution cycle

Test execution is done in quality assurance (QA) environment

Test executions happen in multiple cycles

Test execution phase consists executing the test cases + test script( if automation )

--application should working on test environment not on development environment

Automation test cases---manual effort will be reduced ---main used

It will reduce the lot of time and lot of effort in each and every cycle of testing

---Defects/Bugs---VIM—for each and every interview

Any mismatched functionality found in a application is called as bugs/defect /issue

Any mismatched functionality between expected and actual result

---Variation between expected and actual result---

During test execution test engineers are reporting the mismatches or defects to developer through templates or using tools

Issue is a technical—generic term

Error and mistake both are different

Defect and bugs both are same

---Error---is a programming related---developer used this term

---Mistake—Is a human mistake term ---if as a tester we can write some incorrect test cases that list to missing of the defects

Defect reporting tools

Clear quest --- defect tracking tools is used only for defect reporting

Devtrack-- defect tracking tools is used only for defect reporting

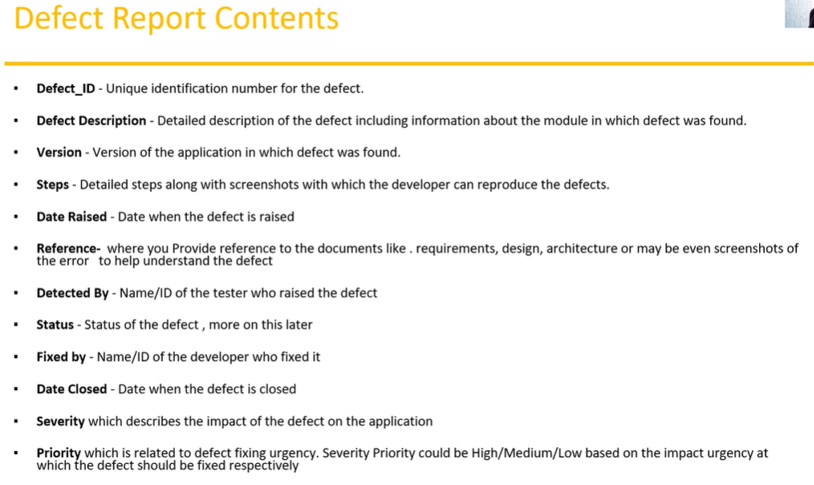
Jira ---- it is test management tools we can track each and every activity during the testing

Quality centre --- it is test management tools we can track each and every activity during the testing

Bug jilla --- defect tracking tools is used only for defect reporting

---Test management tools and defect tracking tools are different –

---Developers also used Jira tools to track every activities.



Defect classification

Defect categorization

|  |  |
| --- | --- |
| Severity | Priority |
| Blocker | P1 |
| Critical | P2 |
| Major | P3 |
| Minor |  |

Defect severity

Severity describe the seriousness of defect how much the impact on workflow business workflow

Blocker (show stopper)—it is a technical term

show stopper—we can call as a blocker

Defect severity can be categorized into four class

Blocker (show stopper) ---This defect indicates nothing can proceed further

Ex.—Application crashed , login not working

--Critical –Mainly basic functionality are not working, customer business workflow is broken . They cannot proceed further

Ex –fund transfer is not working in net banking

Ex –Ordering product in ecommerce application is not working

--Major – It causes some undesirable behavior, but the feature and application is still functional

Ex – after sending email there is no confirm message

Ex –After booking cab there is no confirmation

--Minor---It won’t cause any major break down of the system ---the defect will not impact on any business or any break down

Ex – look and feel issues, spelling and alignment

Tester will assign the severity

Even priority also tester will assign

Defect priority

Priority describe the importance of defect

Defect priority states the order in which a defect should be fixed

Defect priority can be categorized into three class

-(P0)--High---Defect must be resolve immediately as it affect the system severely and cannot be used until it is fixed

-(P1)-Medium—It can wait until a new versions/builds is created

-(P2)-Low –developer can fix it in later releases

After reporting defect priority can be changed by test lead or test manager or product manager

Test should provide the severity and priority but priority can be changed

Severity cannot be change it is a tester responsibility ---developer and business analyst this people not touch the severity

High severity, priority and Low severity, priority

priority

|  |  |  |
| --- | --- | --- |
|  | High | Low |
| High | Login is taking to the blank page | About us link is going to blank page |
| Low | After user is log into application he can see home page but there is a spelling mistake in home page | User open contact page email id has spelling mistake |

Examples of severity and priority

Defect resolution ---One of the column should be specified in bug report and defect report

This column will be developed by developer not a tester ---Developer has to select this file

Developer has to change the resolution

After receiving a defect report from the testing team, development team conducts a review meeting to fixed defect. Then they send the resolution time to testing team for further

--Accept

--Reject

--Duplicate

--Enhancement

--Need more information

--Not reproducible

--Fixed

--As designed