**Assignment Module -2**

* **What is Exploratory Testing?**

Exploratory testing is a concurrent process where test design, execution and logging happen simultaneously.

Testing is often not recorded

Makes use of experience, heuristics and test patterns

The focus of exploratory testing is more on testing as a thinking activity.

Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits

It is not random testing but it is Adhoc testing with purpose of find bugs

It is structured and rigorous.

It is highly teachable and manageable.

It is not a technique but it is an approach.

* **What is traceability matrix?**
* Test conditions should be able to be linked back to their sources
* in the test basis, this is known as traceability.
* To protect against changesyou should be able to trace back from every
* system component to the original requirement that caused its presence.
* Types of Traceability Matrix
* Forward Traceability – Mapping of Requirements to Test cases
* Backward Traceability – Mapping of Test Cases to Requirements
* Bi-Directional Traceability - A Good Traceability matrix is the references from test cases to basis documentation and vice versa.
* **What is Boundary value testing?**
* Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of validranges.
* Boundary value analysis is a methodology for designing test cases that

concentrates software testing effort on cases near the limits of valid ranges

* Boundary value analysis is a method which refines equivalence
* partitioning.
* Boundary value analysis generates test cases that highlight errors better
* than equivalence partitioning.
* The trick is to concentrate software testing efforts at the extreme ends
* of the equivalence classes.
* At those points when input values change from valid to invalid errors
* are most likely to occur.
* Boundary Value Analysis (BVA) uses the same analysis of partitions as
* EP and is usually used in conjunction with EP in test case design
* **What is Equivalence partitioning testing?**
* Aim is to treat groups of inputs as equivalent and to select one representative input to test them all
* EP can be used for all Levels of Testing
* Equivalence partitioning is the process of defining the optimum
* number of tests by:
* Reviewing documents such as the Functional Design Specification and
* Detailed Design Specification, and identifying each input condition within
* a function,
* Selecting input data that is representative of all other data that would
* likely invoke the same process for that particular condition.
* If we want to test the following IF statement: If value is between 1 and

100 (inclusive) (value >=1 and value <=100) Then...

* We could put a range of numbers as shown in the below figure.
* The numbers fall into a partition where each would have the same, or
* equivalent, result i.e.an Equivalence Partition (EP) or Equivalence
* Class
* EP says that by testing just one value we have tested the partition
* (typicallya mid-point value is used). It assumes that:
* If one value finds a bug, the others probably will too
* If one doesn't find a bug, the others probably won't either
* In EP we must identify Valid Equivalence partitions and Invalid
* Equivalence partitions where applicable (typically in range tests)
* The Valid partition is bounded by the values 1 and 100
* Plus there are 2 Invalid partitions
* **What is Integration testing?**
* interfaces and in the interactions between integrated components
* or systems
* Integration Testing is a level of the software testing process
* where individual units are combined and tested as a group.
* The purpose of this level of testing is to expose faults in the interaction
* between integrated units. Test drivers and test stubs are used to assist in
* Integration Testing.
* Integration testing tests integration or interfaces between components,
* interactions to different parts of the system such as an operating system, file
* system and hardware or interfaces between systems.
* Integration testing is done by a specific integration tester or test team.
* Components may be code modules, operating systems, hardware and
* even complete systems
* There are 2 levels of Integration Testing
* Component Integration Testing
* System Integration Testing
* **What determines the level of risk?**
* A properly designed test that passes, reduces the overall level of Risk in a system
* Risk, factor that could result in future negative
* consequences; usually expressed as impact and likelihood
* When testing does find defects, the Quality of the software system increases when those defects are fixed
* The Quality of systems can be improved through Lessons learned from previous projects
* Analysis of root causes of defects found in other projects can lead to process Improvement
* Process Improvement can prevent those defects reoccurring
* Which in turn, can improve the Quality of future systems
* Testing should be integrated as one of the Quality assurance activities
* A Risk could be any future event with a negative consequence

You need to identify the risks associated with your project

* Risks are of two types:
* Project Risks
* Product Risk
* **What is Alpha testing?**
* It is always performed by the developers at the software developmentsite.
* Sometimes it is also performed by Independent Testing Team.
* Alpha Testing is not open to the market and public
* It is conducted for the software application and project.
* It is always performed in Virtual Environment.
* It is always performed within the organization.
* It is the form of Acceptance Testing.
* Alpha Testing is definitely performed and carried out at the developing
* organizations location with the involvement of developers.
* It comes under the category of both White Box Testing and Black
* Box Testing.
* **What is beta testing?**
* Beta Testing is always open to the market and public.
* It is usually conducted for software product.
* It is performed in Real Time Environment.
* It is always performed outside the organization.
* It is also the form of Acceptance Testing.
* Beta Testing is performed and carried out by users or you

can say people at their own locations and site using customer data.

* It is only a kind of Black Box Testing.
* Beta Testing is always performed at the time when software product
* and project are marketed.
* It is always performed at the user’s premises in the absence of the
* development team.
* It is also considered as the User Acceptance Testing which is done at customers or user area.
* Beta testing can be considered pre-release testing.
* Pilot Testing is testing to product on real world as well as collect data
* on the use of product in the classroom.
* **What is component testing?**
* A minimal software item that can be tested in

isolation. It means “A unit is the smallest testable part of

software.”

* Component Testing – The testing of individual software components.
* Unit Testing is a level of the software testing process where individualunits/components of a software/system are tested. The purpose is tovalidate that each unit of the software performs as designed.
* Unit testing is the first level of testing and is performed prior to

Integration Testing.

* Sometimes known as Unit Testing, Module Testing or Program Testing
* Component can be tested in isolation – stubs/drivers may be employed
* Unit testing frameworks, drivers, stubs and mock or fake objects are usedto assist in unit testing.
* Functional and Non-Functional testing
* Unit tests are typically written and run by software developers to
* ensure that code meets its design and behaves as intended with debuggingtool.
* Below we look at some of what extreme programming brings to the worldof unit testing:
* Tests are written before the code
* Rely heavily on testing frameworks
* All classes in the applications are tested
* Quick and easy integration is made possible.
* **What is functional system testing?**
* Functional System Testing : A requirement that specifies a function that
* a system or system component must perform
* A Requirement may exist as a text document and/or a model
* There is two types of Test Approach
* Requirement Based Functional Testing
* Process Based Testing
* Functional System Testing Functionality As below:
* Accuracy Provision of right or agreed results or effects
* Interoperability Ability to interact with specified systems
* Compliance Adhere to applicable standards, conventions, regulationsor laws
* Auditability Ability to provide adequate and accurate audit data
* Suitability Presence and appropriateness of functions for specified tasks
* **What is Non-Functional Testing?**
* Non-Functional Testing: Testing the attributes of a component
* or system that do not relate to functionality, e.g. reliability,
* efficiency, usability, interoperability, maintainability and
* portability
* May be performed at all Test levels (not just Non Functional Systems
* Testing)
* Measuring the characteristics of the system/software that can be
* quantified on a varying scalee.g. performance test scaling
* Non-functional testing includes, but is not limited to, performance
* testing, load testing, stress testing, usability testing, maintainability
* testing, reliability testing and portability testing.
* It is the testing of “how” the system works. Non-functional testing maybe performed at all test levels.
* The term non-functional testing describes the tests required to
* measure characteristics of systems and software that can be
* quantified on a varying scale, such as response times for performancetesting.
* To address this issue, performance testing is carried out to check &
* fine tune system response times. The goal of performance testing is

to reduce response time to an acceptable level

* Hence load testing is carried out to check systems performance at

different loads number of users accessing the system

* **What is GUI Testing?**
* Graphical User Interface (GUI) testing is the process of testing the system’s
* GUI of the System under Test. GUI testing involves checking the screens
* with the controls like menus, buttons, icons, and all types of bars – tool bar,
* menu bar, dialog boxes and windows etc.
* WHAT DO YOU CHECK IN GUI TESTING?
* Check all the GUI elements for size, position, width, length and acceptance of
* characters or numbers. For instance, you must be able to provide inputs to the
* input fields.
* Check you can execute the intended functionality of the application using the GUI
* Check Error Messages are displayed correctly
* Check for Clear demarcation of different sections on screen
* Check Font used in application is readable
* Check the alignment of the text is proper
* Check the Colour of the font and warning messages is aesthetically pleasing
* Check that the images have good clarity
* Check that the images are properly aligned
* Check the positioning of GUI elements for different screen resolution.
* **What is Adhoc testing?**
* Adhoc testing is an informal testing type with an aim to break thesystem.
* It does not follow any test design techniques to create test cases.
* In fact is does not create test cases altogether!
* This testing is primarily performed if the knowledge of testers in thesystem under test is very high.
* Testers randomly test the application without any test cases orany business requirement document.
* Adhoc Testing does not follow any structured way of testing and it israndomly done on any part of application.
* Main aim of this testing is to find defects by random checking.
* Adhoc testing can be achieved with the testing technique

called Error Guessing.

* Error guessing can be done by the people having enough experience onthe system to “guess” the most likely source of errors.
* **What is load testing?**
* Its a performance testing to check system behaviour under load.
* Testing an application under heavy loads, such as testing of a web site under arange of loads to determine at what point the system’s response time degradesor fails.
* Load testing is a kind of performance testing which determines a system’s
* performance under real-life load conditions. This testing helps determine howthe application behaves when multiple users access it simultaneously.
* This testing usually identifies
* The maximum operating capacity of an application
* Determine whether current infrastructure is sufficient to run the application
* Sustainability of application with respect to peak user load
* Number of concurrent users that an application can support, and scalability to
* allow more users to access it.
* It is a type of non-functional testing. Load testing is commonly used for the
* Client/Server, Web based applications – both Intranet and Internet.
* **What is stress testing?**
* System is stressed beyond its specifications to check

how and when it fails.

* Performed under heavy load like putting large

number beyond storage capacity, complex database queries,

continuous input to system or database load.

* Stress testing is used to test the stability & reliability of the system.
* This test mainly determines the system on its robustness and error

handling under extremely heavy load conditions.

* It even tests beyond the normal operating point and
* evaluates how the system works under those extreme

conditions.

* Stress Testing is done to make sure that the system would not
* crash under crunch situations.
* Stress testing is also known as endurance testing.
* **What is white box testing and list the types of white box testing?**
* White Box Testing: Testing based on an analysis of the internal
* structure of the component or system*.*
* Structure-based testing technique is also known as ‘white-box’

or ‘glass-box’ testing technique because here the testers require

* knowledge of how the software is implemented, how it works.
* In white-box testing the tester is concentrating on how the software

does it.

* For example, a structural technique may be concerned with

exercising loops in the software.

* Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.
* Structure-based techniques are also used in system and acceptance testing, but the structures are different.
* For example, the coverage of menu options or major business transactions could be the structural element in system or acceptance testing.
* **Branch Condition testing**
* Branch Condition Testing requires that the True and False of each Boolean operand is tested
* **Branch Condition Combination testing**
* Branch Condition Combination Coverage would require all combinations of boolean operands to be evaluated
* **Modified Condition Decision testing**
* Modified Condition Decision Coverage requires test cases to show that each Boolean operand can independently affect the outcome of the decision
* **Dataflow testing**
* Data flow testing aims to execute sub-paths from points where each variable in a component is defined to points where it is referenced.
* **Linear Code Sequence and jump testing**
* LCSAJ testing requires a model of the source code which identifies control flow jumps.

**What is black box testing? What are the different black box testing techniques?**

* Testing, either functional or non-functional,without reference to the internal structure of the componentor system.
* Specification-based testing technique is also known as ‘black-box’

or input/output driven testing techniques because they view the

* software as a black-box with inputs and outputs.
* The testers have no knowledge of how the system or component is
* structured inside the box. In black-box testing the tester is
* concentrating on what the software does, not how it does it.
* Specification-based techniques are appropriate at all levels of testing
* where a specification exists.
* For example, when performing system or acceptance testing, the requirements specification or functional specification may form the basisof the tests.
* **There are four specification-based or black-boxtechnique:**
* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing
* Use-case Testing
* Other Black Box Testing
* Syntax or Pattern Testing
* **Mention what are the categories of defects?**
* Security Defects: Application security defects generally involve

improper handling of data sent from the user to the application. Thesedefects are the most severe and given highest priority for a fix.

* Examples:
* Authentication: Accepting an invalid username/password
* Authorization: Accessibility to pages though permission not given
* User Interface Defects: As the name suggests, the bugs deal with
* problems related to UI are usually considered less severe.
* Examples:
* Improper error/warning/UI messages
* Spelling mistakes
* Alignment problems
* Data Quality/Database Defects: Deals with improper handling of data in the database.
* Examples:
* Values not deleted/inserted into the database properly
* Improper/wrong/null values inserted in place of the actual values
* Critical Functionality Defects: The occurrence of these bugs hampers
* the crucial functionality of the application. Examples: - Exceptions
* Functionality Defects: These defects affect the functionality of
* the application.
* Examples:
* All JavaScript errors
* Buttons like Save, Delete, Cancel not performing their intended functions
* A missing functionality (or) a feature not functioning the way it is intended to
* Continuous execution of loops
* **Mention what bigbang testing is?**
* In Big Bang integration testing all components or modules is

integrated simultaneously, after which everything is tested as a whole.

* Big Bang testing has the advantage that everything is finished beforeintegration testing starts.
* The major disadvantage is that in general it is time consuming and

difficult to trace the cause of failures because of this late integration.

* Here all component are integrated together at once, and thentested.
* Advantages:
* Convenient for small systems.
* Disadvantages:
* Fault Localization is difficult.
* Given the sheer number of interfaces that need to be tested in thisapproach, some interfaces links to be tested could be missed easily.
* Since the integration testing can commence only after “all” the modulesare designed, testing team will have less time for execution in the testingphase.
* Since all modules are tested at once, high risk critical modules are notisolated and tested on priority. Peripheral modules which deal with userinterfaces are also not isolated and tested on priority.

**What is the purpose of exit criteria?**

* Purpose of exit criteria is to define when we STOP testing either at the:
* End of all testing
* End of phase of testing
* Exit Criteria typically measures:
* Thoroughness measures, such as coverage of requirements or of code or riskcoverage
* Estimates of defect density or reliability measures.
* Cost.
* Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
* Schedules - such as those based on time to market.
* **When should "Regression Testing" be performed?**
* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix
* **What is 7 key principles? Explain in detail?**

1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy

* Testing shows presence ofDefects:
* Testing can show that defects are present, but cannot prove that thereare no defects.
* Testing reduces the probability of undiscovered defects remaining

in the software but, even if no defects are found, it is not a proof ofcorrectness.

* Exhaustive Testing is Impossible:
* Testing everything including all combinations of inputs

and preconditions is not possible.

* So, instead of doing the exhaustive testing we can use risks and

priorities to focus testing efforts.

* Early Testing:
* Testing activities should start as early as possible in the software

or system development life cycle, and should be focused on

defined objectives.

* Testing activities should start as early as possible in the developmentlife cycle
* Defect Clustering:
* A small number of modules contain most of the defects

discovered during pre-release testing, or are responsible for the

most operational failures.

* The Pesticide Paradox:
* If the same tests are repeated over and over again, eventually the sameset of test cases will no longer find any new defects.
* To overcome this “pesticide paradox”, the test cases need to be

regularly reviewed and revised, and new and different tests needto be written to exercise different parts of the software or systemto potentially find more defects.

* Testing is Context Dependent :
* Testing is basically context dependent.
* Different kinds of sites are tested differently.
* Absence of Errors Fallacy:
* If the system built is unusable and does not fulfil the user’s needs andexpectations then finding and fixing defects does not help.
* **Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| QA | QC | Tester |
| * Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | * Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements. | * Activities which ensure the identification of bugs/error/defects in the Software |
| * Focuses on processes and procedures rather than conducting actual testing on the system | * Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process. | * Focuses on actual testing. |
| * Process oriented activities. | * Product oriented activities | * Product oriented activities. |
| * Preventive activities. | * It is a corrective process. | * It is a preventive process |
| * It is a subset of Software Test Life Cycle (STLC). | * QC can be considered as the subset of Quality Assurance. | * Testing is the subset of Quality Control. |

* **Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| Smoke | sanity |
| * Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | * Sanity Testing is done to check the new functionality/bugs have been fixed |
| * Smoke testing is usually documented or scripted | * Sanity testing is usually not documented and is unscripted |
| * Smoke testing is a subset of Acceptance testing | * Sanity testing is a subset of [Regression Testing](https://www.guru99.com/regression-testing.html) |
| * This testing is performed by the developers or testers | * This testing is performed by the developers or testers |

* **Difference between verification and Validation**

|  |  |
| --- | --- |
| Verification | validation |
| * Verification process includes checking documents, design, code, and program | * Validation process includes testing and validation of the actual product. |
| * Verification does not involve code execution | * Validation involves code execution. |
| * Verification uses methods like reviews, walkthroughs, inspections, and desk-checking | * Validation uses methods like [black box testing](https://www.guru99.com/black-box-testing.html), white box testing, and non-functional testing. |
| * Verification finds the bugs early in the development cycle | * Validation finds the bugs that verification can not catch. |

* **Explain types of Performance testing?**
* stress testing
* System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
* Stress Testing is done to make sure that the system would not crash under crunch situations.
* Stress testing is also known as endurance testing.
* Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.
* Load testing
* Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.
* Endurance testing
* It is done to make sure the software can handle the expected load over a long period of time.
* Spike testing
* Spike testing is a type of stress testing that evaluates software performance when workloads are substantially increased quickly and repeatedly. The workload is beyond normal expectations for short amounts of time.
* Volume testing
* Volume testing determines how efficiently software performs with large projected amounts of data. It is also known as flood testing because the test floods the system with data.
* Scability testing
* Scalability testing is used to determine if software is effectively handling increasing workloads. This can be determined by gradually adding to the user load or data volume while monitoring system performance. Also, the workload may stay at the same level while resources such as CPUs and memory are changed.
* **What is Error, Defect, Bug and failure?**
* A mistake in coding is called error.
* error found by tester is called defect
* defect accepted by development team then it is called bug
* build does not meet the requirements then it is failure.
* **Difference between priority and severity?**

|  |  |
| --- | --- |
| Severity | priority |
| * Defect Severity is specified as the degree of impact that a defect has on the operation of the product. | * Defect Priority has specified the order in which the developer should fix a defect. |
| * Severity means the seriousness of the defect in the product functionality. | * Priority means how soon the bug should be fixed. |
| * The test engineer determines the severity level of the defect. | * Priority of defects is decided in discussion with the manager/client. |
| * It is driven by functionality. | * It is driven by business value. |

* **What is Bug Life Cycle?**
* **It** is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it’s not reproduced. it is related to the bug found during testing.
* New**:**  When a defect is logged and posted for the first time. It’s state is given as new.
* Assigned:  After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It’s state given as assigned.
* Open: At  this state the developer has started analyzing and working on the defect fix.
* Fixed: When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’ and the bug is passed to testing team.
* Pending retest:  After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
* [Retest](http://tryqa.com/what-is-retesting/)**:**  At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
* [Verified](http://tryqa.com/what-is-verification-in-software-testing-or-what-is-software-verification/)**:** The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.
* Reopen**:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.
* Closed**:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.
* Duplicate**:** If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate**“.**
* Rejected**:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.
* Deferred**:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are [priority](http://tryqa.com/what-is-the-difference-between-severity-and-priority/)of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
* Not a bug:  The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and feel of the application like change of colour of some text then it is not a bug but just some change in the look of the application.4
* **Explain the difference between Functional testing and NonFunctional testing**

|  |  |
| --- | --- |
| Functional testing | NonFunctional testing |
| * Functional testing verifies each function/feature of the software. | * whereas Non Functional testing verifies non-functional aspects like performance, usability, reliability, etc |
| * Functional testing has a goal to validate software actions | * whereas Non Functional testing has a goal to validate the performance of the software. |
| * A Functional Testing example is to check the login functionality | * whereas a Non Functional testing example is to check the dashboard should load in 2 seconds |
| * Functional describes what the product does | * whereas Non Functional describes how the product works. |

* **What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

|  |  |
| --- | --- |
| SDLC | STLC |
| * SDLC defines all the standard phases which are involved during the software development process, | * whereas the STLC process defines various activities to improve the quality of the product. |
| * In SDLC, the development team creates the high and low-level design plans, | * while In STLC, the test analyst creates the System, Integration Test Plan |
| * In SDLC, real code is developed, and actual work takes place as per the design documents, | * whereas in STLC testing team prepares the test environment and executes test cases. |
| * The SDLC life cycle helps a team complete the software’s successful development | * , while the STLC phases only cover software testing. |

* **What is the difference between test scenarios, test cases, and** test script?

|  |  |  |
| --- | --- | --- |
| Test scenario | Test cases | Test script |
| A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility. | Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks. | A set of sequential instruction that detail how to execute a core business function |
| Test Scenario is ‘What to be tested’ | Test Case is ‘How to be tested’ | One script is written to explain how to simulate each business scenario |

* **Explain what Test Plan is? What is the information that should be covered**
* A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project.
* Think of it as a blueprint for running the tests needed to ensure the software is working correctly – controlled by test managers.
* **What is priority?**
* Priority is defined as the order in which the defects should be resolved. The priority status is usually set by the testing team while raising the defect against the dev team mentioning the timeframe to fix the defect. The Priority status is set based on end users requirement.
* For example: If the company logo is incorrectly placed in the company's web page then the priority is high but it is of low severity.
* **What is severity?**
* The severity of the bug or the defect A problem or a Defect's severity in testing refers to how much of an impact it has on the software program under test. A higher severity rating indicates that the bug/defect has a greater impact on system functionality. The severity level of a bug or defect is generally determined by a Quality Assurance engineer.
* **Difference between priority and severity?**

|  |  |
| --- | --- |
| Severity | priority |
| * Defect Severity is specified as the degree of impact that a defect has on the operation of the product. | * Defect Priority has specified the order in which the developer should fix a defect. |
| * Severity means the seriousness of the defect in the product functionality. | * Priority means how soon the bug should be fixed. |
| * The test engineer determines the severity level of the defect. | * Priority of defects is decided in discussion with the manager/client. |
| * It is driven by functionality. | * It is driven by business value. |

* **Bug categories are…**
* **Functional:** Errors are considered functional defects or functional bugs when the software refuses to work according to the functional requirements. Functional testing is performed to discover the presence of such errors.
* **Performance:** Performance bugs are defects that hamper a software's performance. This can be related to the speed, response time, overall stability, and resource consumption of the software. And, yes, these types of defects are discovered through performance testing. An example of a performance bug is when a system's loading time or response time is a lot longer than that mentioned in the requirements.
* **Usability**: Usability defects are related to the ease-of-use of an application. A software/ application/ system needs to be well designed with its content placed in the right position and in the correct order to maximize the user experience and over usability.
* **Compatibility:** Systems with compatibility defects are incapable of performing uniformly on different operating systems, hardware, browsers, and specific network configurations. Software testers perform compatibility testing to discover the presence of any such bugs.
* **Security:** Security bugs or security defects are the deficiencies that may result in a potential security attack. Security testing is perhaps the test that is performed most often.
* **Advantage of Bugzila .**
* Open source, free bug tracking tool.
* Automatic Duplicate Bug Detection.
* Search option with advanced features.
* File/Modify Bugs By Email.
* Move Bugs Between Installs.
* Multiple Authentication Methods (LDAP, Apache server).
* Time Tracking.
* Automated bug reporting; has an API to interact with system.
* **What are the different Methodologies in Agile Development Model?**

#### Kanban

* Kanban is a simple, visual means of managing projects that enables teams to see the progress so far and what’s coming up next. Kanban projects are primarily managed through a Kanban board, which segments tasks into three columns: “To Do,” “Doing,” and “Done.”

#### Scrum

* Scrum is similar to Kanban in many ways. Scrum typically uses a Scrum board, similar to a Kanban board, and groups tasks into columns based on progress. Unlike Kanban, Scrum focuses on breaking a project down into sprints and only planning and managing one sprint at a time. Scrum also has unique project roles: Scrum master and product owner.

#### Extreme Programming (XP)

* Extreme Programming (XP) was designed for Agile software development projects. It focuses on continuous development and customer delivery and uses intervals or sprints, similar to a Scrum methodology. However, XP also has 12 supporting processes specific to the world of software development:

#### Feature-driven development (FDD)

* Feature-driven development is another software-specific Agile framework. This methodology involves creating software models every two weeks and requires a development and design plan for every model feature. It has more rigorous documentation requirements than XP, so it’s better for teams with advanced design and planning abilities.

#### Dynamic Systems Development Method (DSDM)

* The Dynamic Systems Development Method (DSDM) was born of the need for a common industry framework for rapid software delivery. Rework is to be expected, and any development changes that occur must be reversible. Like Scrum, XP, and FDD, DSDM uses sprints. This framework is based on eight fundamental principles:

#### Crystal

* Crystal is a family of Agile methodologies that includes Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Red, etc. Each has a unique framework. Your choice depends on several project factors, such as your team size, priorities, and project criticality.

#### Lean

* Lean development is often grouped with Agile, but it’s an entirely different methodology that happens to share many of the same values.
* **Explain the difference between Authorization and Authentication in Web testing.What are the common problems faced in Web testing?**
* Authentication verifies the identity of a user or service, and authorization determines their access rights.
* Integration. Integration testing exposes problems with interfaces among different program components before deployment. ...
* Interoperability. ...
* Security. ...
* Performance. ...
* Usability. ...
* Quality Testing, Exceptional Services.