**ASSIGNMENT**

1. **What is exploratory testing?**

* Exploratory testing is a type of software testing where test cases are not created in advance but testers checks system on the fly. They may note down ideas about what to test before test execution. the focus of exploratory testing is more on testing as thinking activities

1. **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.”
* **ERROR:** A discrepancy between a computed, observed, or measured value or condition and true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.…
* **Failure:** The inability of a system or component to perform its required functions within specified performance requirements. **See EX.** Bug, crash, exception, and fault. crash, exception, and fault.
* **Bug:** A fault in a program which causes the program in an unintended or unanticipated manner. **See EX.** Anomaly, defect, error, exception, and fault. Bug is terminology of tester.

1. **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.

1. **What is boundary value testing?**

* Boundary value analysis is a methodology for designing test case that concentrate software testing effort on cases near the limit of valid range.
* Boundary value analysis is method which refines equivalence partitioning.
* Boundary value analysis generate test cases that highlight error better than equivalence partitioning.
* The trick is to concentrate software testing effort at the extreme end of equivalence classes.
* At those point when input value change from invalid error are most likely to occur.

1. **What is equivalence partitioning testing?**

* Aim is to treat group of inputs as equivalent and to select one representative input to test them all.
* Equivalence partitioning can be used for all level of testing.
* If we want to test the following if statement:

“If value is bet 1&100(inclusive)

E.g., Value >=1 & value <=100) then”

1. **What is integration testing?**

* Integration testing performed to expose defects in the interfaces and in the interactions between integrated components or system.
* Integration testing is level of the software testing process where individual units are combined and tested as a group.
* Integration testing is done by a specific integration tester or test team.
* Components may be code modules, operating system, hardware and even complete systems.
* There are 2 levels of integration testing

1) component integration testing

2) system integration testing

1. **What is alpha testing?**

* It is always performed by the developers at the software development site.
* 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 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.

1. **What is beta testing?**

* It is always performed by the customers at their own site.
* It is not performed by independent testing team.
* 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 (field 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 (UAT) which is done at customers or users’ 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.

1. **What is component testing?**

* 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 level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate 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.
* Unit testing frameworks, drivers, stubs and mock or fake objects are used to 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 debugging tool.
* Unit tests find problems early in the development cycle.
* Unit testing is performed by using the white box testing method.

1. **What is functional system testing?**

* Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.
* Functional testing is executed first.
* Manual testing or automation tools can be used for functional testing.
* Business requirements are the inputs to functional testing.
* Functional testing describes what the product does.
* Easy to do manual testing.
* Types of functional testing are:

1. Unit testing
2. Smoke testing
3. Sanity testing
4. Integration testing
5. White box testing
6. Black box testing
7. User acceptance testing
8. Regression testing
9. **What is non-functional testing?**

* Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.
* Non-functional testing should be performed after functional testing.
* Using tools will be effective for this testing.
* Performance parameters like speed, scalability are inputs to non-functional testing.
* Non-functional testing describes how good the product works.
* Tough to do manual testing.
* Types of non-functional testing are:

1) Performance testing

2) Load testing

3) Volume testing

4) Stress testing

5) Security testing

6) Installation testing

7) Penetration testing

8) Compatibility testing

9) Migration testing

1. **What is GUI testing?**

* Graphical user interface (GUI) testing is the process of testing the systems 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 color of the font and warning messages is aesthetically pleasing.
* Check that the image have good clarity.
* Check that the image are properly aligned.
* Check the positioning of GUI elements for different screen resolution.

1. **What is Ad-hoc testing (Error testing)?**

* Ad-hoc testing is an informal testing type with an aim to break the system.
* 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 the system under test is very high.
* Testers randomly test the application without any test cases or any business requirement document.
* Ad-hoc testing does not follow any structured way of testing and it is randomly done on any part of application.
* Main aim of this testing is to find defects by random checking.
* Ad-hoc testing can be achieved with the testing technique called error guessing.
* Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.
* The error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.
* Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weakness.
* This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective.
* It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won’t be able to find.
* Using experience to postulate errors.
* Use error guessing to complement test design techniques.

1. **What is Load testing?**

* Load testing is to test the system behavior under normal workload conditions, and it is just testing or simulating with the actual workload.
* Load testing identifies the bottlenecks in breaking the system under various workloads and checks how the system reacts when the load is gradually increased.
* Load testing does not break the system.

1. **What is Stress testing?**

* Stress testing is to test the system behavior under extreme conditions and is carried out till the system failure.
* Stress testing determines the point of the system to reveal the maximum point after which it breaks.
* Stress testing tries to break the system by testing with overwhelming data or resources.

1. **What is white box testing and list the types of white box testing?**

* White box testing based on an analysis of the internal structure of the component or system.
* Structure based testing technique is also knowns as ‘white box’ or ‘glass box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.
* **Types of coverage:**

1. **Statement coverage**
2. **Decision coverage**
3. **Condition coverage**
4. **Statement/ segment coverage:**

* The statement coverage is also knowns as line coverage or segment coverage.
* The statement coverage covers only the true conditions.

1. **Decision/ branch coverage:**

* Decision coverage also known as branch coverage or all edges coverage.
* It covers both the true and false conditions unlikely the statement coverage.
* Aim is to demonstrate that all decision have been run at least once.
* With an IF statement, the exit can either be TRUE OR FALSE depending on the value of the logical condition that comes after IF.

1. **Condition coverage:**

* This is closely related to decision coverage but has better sensitivity to the control flow.
* However, full condition coverage does not guarantee full decision coverage.
* Condition coverage reports the true of false outcome of each condition.
* Condition coverage measures the conditions independently of each other.

1. **What is Black box testing? What are the different Black box testing techniques?**

* The tester have no knowledge of how the system on component is structured inside the box.
* In black box testing the tester is concentrating on what the software does not how it does it.
* The technique of testing without having any knowledge of the interior working of the application is black box testing.
* The tester is oblivious to the system architecture and does not have access to the source code.
* **Black box testing techniques**

1. Equivalence partitioning
2. Boundary value analysis
3. Decision table
4. State transition testing
5. Use case testing
6. Other black box testing
7. **Mention what are the categories of defects?**

* Defect can be categorized in to different types basing on the core issue they address.

1. **Data Quality/ Database Defects:**

* Deals with improper handling of data in the database.
* EX. Value not deleted/ inserted into the database properly improper/ wrong/ null values inserted in place of the actual values.

1. **Critical Functionality Defects:**

* The occurrence of these bugs hampers the crucial functionality of the application
* Ex. Exceptions

1. **Functionality Defects:**

* These defects affect the functionality of the application.
* Ex. All JavaScript errors
* Buttons like save, delete, cancel not performing their intended functions.
* A missing functionality a feature not functioning the way it is intended to
* Continuous execution of loops

1. **Security Defects:**

* Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
* Ex. **Authentication:** Accepting an invalid username/password.
* **Authorization:** Accessibility to pages though permission not given.

1. **User Interface Defects:**

* As the name suggests, the bugs deal with problems related to UI are usually considered less severe.
* Ex. Improper error/warning/UI messages (GUI)
* Spelling mistakes
* Alignment problems

1. **Mention what big bang testing is?**

* In big bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* The major disadvantage is that in general it is time consuming and difficult to trace the cause of failure because of this late integration.
* Here all component are integrated together at once, and then tested.
* **Advantage:** Convenient for small systems.
* **Disadvantage:** Fault localization is difficult.
* Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase.
* 1) Top-down approach
* 2)Bottom-up approach

1. **What is the purpose of Exit criteria?**

* Successful testing of integrated application.
* Executed test cases are documented.
* All high prioritized bugs fixed and closed.
* Technical documents to be submitted followed by release notes.

1. **When should “Regression Testing” be performed?**

* Ideally regression testing should be performed whenever your code base has been modified or altered.
* Frequent partial regression testing will help developer fix reported defects on the time.
* Most common reason to run regression test is the introduction of new functionality.
* Regression testing can save developers lot of time with timely detection of bugs that would otherwise cause the projects a lot of pain in long time.

1. **What is 7 key principles? Explain in detail?**

* According to ISTQB (International Software Testing Qualification Board) the seven principle of software testing are mentioned below:

1. **Exhausting testing is not possible**
2. **Defects clustering**
3. **Pesticide paradox**
4. **Testing shows presence of defects**
5. **Absences of error fallacy**
6. **Early testing**
7. **Testing is context dependent**
8. **Exhausting testing is not possible:**

* Exhausting testing is not possible instead we need optimal testing based on

risk assessment.

* Testing everything is not possible.

1. **Defect clustering:**

* Small numbers of modules contain most of the defects discovered during pre-release testing or are responsible for most operational failures.

1. **Pesticide paradox:**

* If the same test are repeated over and over eventually the same set of test cases will no longer find any new defects.
* To overcome this problem test case needs to regularly reviewed and revised and new different parts of the software or system to potentially find more defects.

1. **Testing shows presence of defects:**

* Testing shows that defects are present but can not improve that there are not defects.

1. **Absences of error fallacy:**

* If the system built is unusable and does not full fill the users need and expectations then.
* Finding and fixing defects does not help.

1. **Early testing:**

* Testing activities should start as early as possible in the software or system or system.
* Development cycle and should be focused on defined objective.

1. **Testing is context dependent:**

* Testing is context dependent which is basically means that the way your test e-commerce site will be different from the way you test commercial off the application.
* Ex. Critical software is tested differently than the e-commerce website.

1. **Difference between QA v/s QC v/c TSTERS**

**QA**

* Activities which ensure the implementation of processes, procedures and standards in contest to verification of developed software and intended requirement
* Focuses on processes and procedures rather than conducting actual testing on the system
* process oriented activities
* preventive activities
* It is a subset of software test life cycle (STLC)

**QC**

* Activates which ensure the verification of development software with respect to documented (or not in some cases) requirement
* Focuses on Actual testing by executing software with intend to identify bug/ defect through implementation of procedures and process
* product oriented activities
* Qc can be considered as the subset of quality Assurance

**Testers**

* Software testing refers to the activities that are preformed on program after it has been written.
* Software testing initiates verification of application of application functionality as per requirements
* It is based on product oriented activities .
* Testing is subset of quality control.

1. **Difference between smoke and sanity**

**Smoke Testing: -**

* Check the critical functionality
* It’s done in initial stage
* It checks the stability
* Part of acceptance testing
* General health check up
* Done by tester and developer
* Its checks the system end to end
* 20 test cases it should take 30 min to test

**Sanity testing: -**

* Check the new functionality
* It is done after 30 build
* It checks the sanity/rationality
* part of regression testing
* Advance health check up
* Done by tester
* It checks only a particular function of entire system

1. **Difference between verification and validation**

**Verification**

* It is process of checking if product is developed as per specification
* It tests requirement architecture design and code of software product
* It does not require executing the code
* A few activities involved in verification testing are requirement verification and code
* It targets internal aspect such as requirement design software architecture data base and code

**Validation**

* It processes of ensuring that the product meets the need and expectation of stake holders
* It tests the usability functionalities and reliability of product
* It emphasizes executing codes to test the usability and functionality of end product
* The commonly used validation activities in software testing are usability, performance , system and security testing
* It targets product ready to be deployed

1. **Types of performance testing**

There are mainly six types of performance testing is done which are mentioned below

1. load testing
2. stress testing
3. endurance testing
4. spike testing
5. volume testing
6. Scalability testing
7. **What is error, bug ,defect and failure ?**

**Error**

An error is define as a human action that produce incorrect result

**Defect**

A flaw in a component or system that can cause the component or system to fail to perform its required function

**Failure**

It is defined as the deviation of component or system from its expected delivery service or result

**Bug**

A fault in program which causes the program to perform in an untitled or unanticipated manner

1. **Difference between priority and severity**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Severity in Testing** | **Priority in Testing** |
| Definition | Severity is a term that denotes how severely a defect can affect the functionality of the software. | Priority is a term that defines how fast we need to fix a defect. |
| Parameter | Severity is basically a parameter that denotes the total impact of a given defect on any software. | Priority is basically a parameter that decides the order in which we should fix the defects. |
| Relation | Severity relates to the standards of quality. | Priority relates to the scheduling of defects to resolve them in software. |
| Value | The value of severity is objective. | The value of priority is subjective. |
| Change of Value | The value of Severity changes continually from time to time. | The value of Priority changes from time to time. |
| Who Decides the Defect | The testing engineer basically decides a defect’s severity level. | The product manager basically decides a defect’s priority level. |
| Types | There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical. | There are 3 types of Priorities: High, Medium, and Low. |

1. **What is bug life cycle ?**

A computer bug is an error flaw mistake , failure , or a a fault in a computer program that prevents it from working correctly 

1. **Difference between functional and non-functional testing**

| * **Functional Testing** | * **Non-functional Testing** |
| --- | --- |
| * It verifies the operations and actions of an application. | * It verifies the behavior of an application. |
| * It is based on requirements of customer. | * It is based on expectations of customer. |
| * It helps to enhance the behavior of the application. | * It helps to improve the performance of the application. |
| * Functional testing is easy to execute manually. | * It is hard to execute non-functional testing manually. |
| * It tests what the product does. | * It describes how the product does. |
| * Functional testing is based on the business requirement. | * Non-functional testing is based on the performance requirement. |
| * **Examples:**   + Unit Testing   + Smoke Testing   + Integration Testing * **4.** Regression Testing | * **Examples:**   + Performance Testing   + Load Testing   + Stress Testing * **4.** Scalability Testing |

1. **What is difference between SDLC and STLC**

| * **SDLC** | * **STLC** |
| --- | --- |
| * SDLC is mainly related to software development. | * STLC is mainly related to software testing. |
| * Besides development other phases like testing is also included. | * It focuses only on testing the software. |
| * SDLC involves total six phases or steps. | * STLC involves only five phases or steps. |
| * In SDLC, more number of members (developers) are required for the whole process. | * In STLC, less number of members (testers) are needed. |
| * In SDLC, development team makes the plans and designs based on the requirements. | * In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. |
| * Goal of SDLC is to complete successful development of software. | * Goal of STLC is to complete successful testing of software. |
| * It helps in developing good quality software. | * It helps in making the software defects free. |
| * SDLC phases are completed before the STLC phases. | * STLC phases are performed after SDLC phases. |
| * Post deployment support, enhancement, and update are to be included if necessary. | * Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts. |
| * Creation of reusable software systems is the end result of SDLC. | * A tested software system is the end result of STLC. |

1. **what is difference between test cases, test scenarios and test scripts**

**Test cases**

* A test case is a high-level document with instructions on the specific functionality of the software product to be tested.
* A test case is the software development life cycle’s ‘What to test’ component.
* Test cases are written in simple English.
* A test case is a document with instructions on testing the specific functionality of an application.
* Test scenarios serve as an outline for writing test cases
* Test cases are primarily used in Manual Testing.
* It ensures end-to-end test coverage with assumed data types.
* Test cases take a lot of time and resources to document.

**Test scripts**

* Test Script is a step-by-step instruction to test each software product’s functionality (test case)
* Test script is the software development life cycle’s ‘How to test’ component.
* Test scripts are written in programming languages like VB Script, Python, Java, etc.
* Test Script is a program that runs various test data on the functionality of an application.
* Test Script is a program that runs various test data on the functionality of an application.
* Test Case serves as an outline for writing test scripts.
* Test scripts are widely used in Automation
* It can serve as a reusable component. A single script can run tests with various types of data.
* The time to execute Test scripts is less than the time to write test cases.

**Test scenarios**

* A test scenario contains high-level documentation which describes an end to end functionality to be tested.
* It focuses on more “what to test” **than** “how to test”.
* Test scenarios are a one-liner. So, there is always the possibility of ambiguity during the testing.
* Test scenarios are derived from test artefacts like BRS, SRS, etc.
* It helps in an agile way of testing the end to end functionality
* Test scenarios are high-level actions.
* Comparatively less time and resources are required for creating & testing using scenarios.

1. **explain what is test plan is ? what is the information that should be covered .**

Test plan is determine as a document describing the scope , approach , resources and schedule.

Test plan should include following things

* 1. Test Strategy and Objectives  
     Identify the main purpose of testing (in light of the product requirements) and what a successful completion of a testing cycle looks like. The two important factors in terms of quality assurance are usually coverage and velocity. The test strategy will define which one comes first (usually based on the business objectives).
  2. Schedule, Estimation, and Deliverables  
     This depends whether your test cycle occurs during or after the cycle, but it is important to be able to determine how long will it take to qualify the tested functionality. And no less important is to understand “what will I get” as a user by the end of the testing cycle — is there any traceability matrix report I can rely on in order to support the decision making process? Predictability is key in this area, as the business would always want to know when we should be ready to ship.
  3. Resources Required to Perform Testing  
     Given different coverage needs, it is important to understand what will it take — for example — to complete a testing cycle within x amount of days/hours. Resources include people (testing team) and technology (testing platforms/product units, testing tools — frameworks and other orchestration/execution tools).

1. **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 defects against the developer team mentioning the timeframe to fix defect.
* The priority is based on end user requirement.

1. **What is severity?**

* The impact of the bug or defect on the application is known as the severity of how blocker defects and what is the impact of the defect on whole system ‘s functionality.
* The severity is the parameter set by the tester while opening bug/ defect and essentially in control the tester.

1. **Bugs categories are…**

Bugs categories are mention below

1. Database defect
2. Critical defect
3. Functionality defect
4. Security defect
5. User interface defect

**Database defect**

* Deal with improper handling of the data in the database.
* Value not inserted in to database properly
* Improper values inserted in place of the actual value

**Critical defect**

* The occurrence of the bugs hampers the crucial functionality of the application
* Ex exceptions

**Functionality defects**

* These defects affect the functionality of the application
* Ex java script

**Security defects**

* Application security defect genrally involve import handling of data sent from the user to the improper handling of data sent from the user to the application .
* These defects are the most severe given highest priority for a fix .
* Authentication
* Authorisation

**User interface defects**

* + - As the name suggest the bugs deals with problem related to ui are usally considered less severe
    - **ex improper error**
    - **spelling mistake**
    - **alignment problem**

1. **Advantages of Bugzilla**

**The Advantages of Bugzilla are:**

* It is an open-source widely used bug tracker;
* It is easy in usage and its user interface is understandable for people without technical knowledge;
* It easily integrates withtest management instruments;
* It integrates with an e-mailing system;
* It automates documentation.

1. **Difference between priority and severity**

## Difference Between Severity and Priority in Testing

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Severity in Testing** | **Priority in Testing** |
| Definition | Severity is a term that denotes how severely a defect can affect the functionality of the software. | Priority is a term that defines how fast we need to fix a defect. |
| Parameter | Severity is basically a parameter that denotes the total impact of a given defect on any software. | Priority is basically a parameter that decides the order in which we should fix the defects. |
| Relation | Severity relates to the standards of quality. | Priority relates to the scheduling of defects to resolve them in software. |
| Value | The value of severity is objective. | The value of priority is subjective. |
| Change of Value | The value of Severity changes continually from time to time. | The value of Priority changes from time to time. |
| Who Decides the Defect | The testing engineer basically decides a defect’s severity level. | The product manager basically decides a defect’s priority level. |
| Types | There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical. | There are 3 types of Priorities: High, Medium, and Low. |

1. **what are the different methodology in Agile development model**

Agile methodology is a “step by step” dynamic focused on short-term visibility but never losing the long-term product goal.

## Main Agile methodologies:

### **1. Scrum**

### Scrum is the most used of the many frameworks underpinning Agile methodology.

* **Scrum is characterised by cycles or stages of development, known as sprints**
* It is usually used in the management of the development of software products but can be used successfully in a business-related context.

### **2. Kanban**

* The word Kanban is of Japanese origin and its meaning is linked to the concept of “just in time”.
* the Kanban method is organised on a board or table (Kanban board), divided into columns, showing every flow within the software production project.
* As the development evolves, the information contained in the table changes, and whenever a new task comes into play, a new “card” is created.
* This methodology is also useful in individual business departments, such as HR, marketing, etc., bringing the desired visibility over all the team’s tasks.
* The Kanban method **requires communication and transparency** so that the members of any team all know exactly what stage development is at and can see the status of a project at any time.

### **3. Extreme Programming**

* This is a typical Agile development framework, developed by Kent Beck, and can be adapted to development companies of various dimensions.
* **Extreme Programming**  methodology is based around the idea of discovering “the simplest thing that will work” without putting too much weight on the long-term product view.
* Teamwork is extremely important in XP, since, when there is a problem, it is solved by the whole team of managers, developers or customers, bringing them together to promote conversation and engagement and break down barriers to communication.

### **4. Lean Development**

* Lean development is a methodology that comes directly from Lean Manufacturing, created by Toyota, and applied to software development.
* This method offers a conceptual framework and follows values, principles and good development practices that can be applied to an Agile development approach.
* **Lean development forces the team to ruthlessly remove any activity that does not bring ultimate value to the product.**

### **5. Crystal**

* This is a family of Agile methodologies, and **Crystal is one of the most flexible frameworks, giving tremendous freedom to the team to develop their own processes.**
* It focuses way more on individuals and how they interact rather than on the process or the tools so communication is an essential key aspect.
* Crystal has variants such as **Crystal Clear** (up to an 8-person team), **Crystal Yellow**(up to a 10 to 20-person team), **Crystal Orange** (up to a 20 to 50-person team) and **Crystal Red**(for big teams with 50 to 1000 people).
* Crystal focuses on principles such as People, Interactions, Community, Skills, Talent and Communication, aiming to deliver the best possible software development process.
* It’s a light methodology in terms of documentation, where teams can find their own ways over preferred work modalities, removing management overheads and creating a “free” process.

1. **explain the difference between authorization and authentication in web testing , what are the common problems faced in web testing ?**

* **Authentication :- a**ccepting an invalid user name / password
* **Authorization :- a**ccessibility to pages through permission not given

In web testing some common problems are mentioned below

1. interoperability
2. security
3. performance
4. responsiveness
5. **when to use usability testing**

* usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you’ve begun the strategy work around a brand new site or app.
* This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants.
* Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed  to achieve the project goals with as little disruption as possible.

1. **what is the procedure for G UI testing ?**

GUI is a software testing type that checks the graphical user interface of the software

* Check all the GUI element size position , width , length etc
* Check for error massages are displayed correctly
* Check for clear demarcation of different section on screen
* Check font used in application is readable
* Check the alignment of text is proper
* Check the colour of the font and warning messages is aesthetically pleasing
* Check all the images have good celerity
* check all the images are properly aligned
* Check the positioning of GUI elements for different screen resolution