***Software testing Assignment***

**Module =2 (manual testing)**

**---------------------------------------------------------------------**

Q.1 what are errors, Bug, defects, and failure

* Error = A mistake in coding is called an error. & Functionality is working but not correctly (developer) & any abnormal behavior (spelling mistake, wrong syntax. etc)
* Defects = error found by tester is called defects (tester), when the software does not behave as per use need/intention
* Bug = defects accepted the by developer team then it is called bug
* Failure = build does not meet the requirement then it is called failure. & a software does not work as per its capacity is called failure

Q.2 what is Equivalence partitioning value testing

* Aim to target groups of input as equivalent selecting one representative input to test them all
* Equivalence partitioning be used for all levels of testing
* Ep says that by testing just one value we have tested the partition (typically a mid-point value is used)
* If one value finds a bug, the other probably will too
* If one does not find a bug, the other probably won’t either
* Partition data into various class and we can select data according to class then test it reduce the number of test cases and save time for testing
* Value check
* Classify/divide/partition into multiple class

**Ex**  enters the number\_\_\_\_ \*allow from 1-300

**Normal test Divided values into test data using**

**Data equivalence class Equivalence partitioning**

1. -100 to 0 = -50 (invalid) -50

2 1 to 100 = 30 (valid) 30

3 101 to 200 = 160 (valid) 160

. 201 to 300 = 240 (valid) 240

. 301 to 400 = 355 (invalid) 355

.

300

Q.3 What is boundary value testing

* Software testing efforts on cases near the limits of valid ranges
* B.V.A method which refines equivalences partitioning
* B.V.A test that highlights error better than equivalences partitioning
* The boundary of the values
* Six parameters should all cover

- min - max

- min-1 - max-1

- min+1 - max+1

**Ex enters age \_\_\_ \*allow from digits 18-35**

Min=18(pass) max=35(pass)

Min-1 =17(fail) max-1 =34(pass)

Min +1 =19(pass) max+1 = 36(fail)

Q.4 What is integration testing?

* Integration testing is a level of the software process where individual units are combined and tested as a groups
* Expose interfaces and the integration between integrated components or system
* Performed between 2 or more modules
* Focuses data communication between multiple modules
* Integration testing performed after unit and before system testing
* Integration testing can test developers themselves or independent testers perform integration testing
* There are two levels of integration testing

- component(unit) integration

- system integration

* There are two types of integration testing methods

- big bang integration testing

- incremental integration testing

- top-down approach(stub)

- bottom-up approach(driver)

Q.5 mentions what big bang testing is.

* In Big bang integration testing all components or modules is integration simultaneously, after which everything is tested as whole
* Big bang testing has the advantage that everything is finished before integration testing starts
* Here all components are integrated together at once, and then tested
* Don’t wait for integration (finished before integration testing starts)

**Advantages:**

* Convenient for small system

**Disadvantages:**

* Fault localization id difficult
* Some interface links to be tested could be missed easily

Q.6 what is 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 model
* Functional system testing functionality is as below:
* Accuracy = provision of right or agreed on results or effects
* Interoperability = ability to interact with specified systems
* Compliance = adhere to application standards, conventions, or laws
* Auditability = ability to provide adequate and accurate audit data
* Suitability = presence and appropriateness of functions for specified tasks
* There are two types of test approach
* Requirement-based functional testing
* Process-based testing
* Requirement based testing
* Testing against requirements and specifications
* High-level system design
* User documentation
* System requirements functional specification
* Starts by using the most appropriate back-box testing techniques
* May support this with white-box techniques
* Risk-based approach
* Business process-based testing
* Expected user profiles
* Business scenarios
* Use case
* Therefore, test cases should be based on real business processes

Q.7 What is non-functional testing

* Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing, and portability testing
* May be performed at all test levels (not just non-functional system testing)
* It is testing of how the system works
* Concentrates on user expectation
* **Ex**
* Web-based testing = in the website the number of user/customer will increase, how the website will handle to every customer/user
* Desktop-based testing = in desktop-based testing check memory, and different other issues
* Mobile-based testing = in mobile, automatically will switch off without any reason
* Game-based testing = confirms the workability and stability of the software

Q.8 explain the difference between functional testing and non-functional testing

**Functional testing non-Functional testing**

* Functional testing is executed first - non-functional testing should be performed after

Functional testing

* Manual and automation testing tools can - using tools will be effective for this testing

Be used for functional testing

* Functional testing describes what the - non-functional testing describes how software

Product does works

* Easy to do manual testing - tough to do manual testing
* Concentrates on user requirement - concentrates on user expectation
* Business requirement are the input - performance parameter like speed, scalability

to functional testing are input to non-functional testing

* Type of functional testing - type of non-functional testing
* **Unit testing - performance testing**
* **Smoke testing - load testing**
* **Sanity testing - volume testing**
* **Integration testing - stress testing**
* **White box testing - security testing**
* **Black box testing - installation testing**
* **User acceptance testing - compatibility testing**
* **Regression testing - migration testing**
* **End to end testing**
* **Experience-based testing**

Q.9 what is exploratory testing

* Exploratory testing is not random testing but it is ad-hoc testing with purpose of find bugs
* Is highly teachable and manageable
* No document
* No plan
* Informal testing
* Tester should know application functionality
* Intension break to application and find out corner defects
* Make use of experience or test patterns
* Testing is often not recorded
* We have to explore the application, understand completely and test it. (without document)
* Understand app, identify possible scenarios use it for testing

Exploratory testing drawback

* You might misunderstand any feature as bug or bug as feature since you do not have requirement
* Time-consuming
* If there is ant bug in app you will never know about it

Q.10 What is ad-hoc testing

* Testing the app randomly without any test cases or any business requirement any test cases or any business requirement document
* Informal testing type with the aim to break system
* This testing is usually an unplanned activity
* Intension break to application and find out corner defects
* Random testing
* No plan
* No documentation
* A tester should know the application functionality
* Three types of ad-hoc testing

Buddy testing = one developer team and another testing

Pair testing = tester and scriber (tester but take notes)

Monkey testing = randomly testing the product

Q. 11 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 system
* Known as a white box or glass box
* Knowledge of how the software is implemented(code), how it is work
* White box testing is the detailed investigation of internal logic and structure of the code

EX = web-based testing

* Analyze the logic by reading the code
* Types of Coverage
* Statement coverage /segment coverage
* Decision coverage /branch coverage
* Condition coverage

Q. 12 what is black box testing? What are different black box testing techniques?

* Don’t show the code
* Testers have no knowledge of how the system or component is structed inside the box
* System architecture and does not have access to the source code
* The tester will interact with the system’s user interface by providing input and examining output without knowing how and where input is worked
* different black box testing techniques
* equivalence partitioning (equivalence class partitioning)
* boundary value analysis (B.V.A)
* decision table
* state transition
* use case

Q.13 Difference between smoke and sanity?

**Smoke testing sanity testing**

* check the critical functionality of - check the main and new functionality of the

application

* it is the initial stage - it is done stable builds or after 30 builds
* build either stable or unstable - sanity builds relatively stable
* performed by both testers or - performed by only tester

developer

* done every time is new build - planned when not enough time in-depth

release testing

* part of acceptance testing - part of regression testing
* smoke testing is usually document - sanity testing usually not documented or

or scripted unscripted

- smoke testing is like general health - sanity testing is like specialized health check up

Check up

Q.14 Different between QA v/s QC v/s Tester

**QA** **QC Tester**

-process-oriented activities - product-oriented activities - product-oriented activities

-preventive activities - corrective process - preventive activities

-QA is process related - QC is actual testing

Software

-focuses on building in - focuses on testing for

Quality Quality

-QA is entire life cycle - QC for testing part in SDLC

-it is subset of software test - QC is can be considered as the - testing is the subset of Quality

Life cycle (STLC) subset of quality assurance control

Q.15 What is GUI testing?

* Graphical user interface
* GUI testing involves checking the screen with the control like menu, button, icons, and all types of the bar – toolbar, menu bar, dialog box and windows, etc.
* Process of the user interface of an application
* GUI is frontend testing
* What do check in GUI testing
* Check all the GUI elements for size, position, width, length.
* Check you can execute the intended functionality of the application using the GUI \
* Check error message are display correctly
* Check font used in application is readable
* Check alignment of the text is proper
* Check that the images have good clarity
* Check that the images are proper aligned
* Heading whether it is proper aligned or not
* Color of the hyper link
* Interface attractive or not
* Testing of the spelling
* The color of the font
* Font whether readable or not
* UI elements like buttons, text box, check box, radio button, drop downs etc.

Q.16 What is component(unit) testing

* Component testing is smallest testable part of software
* Some time know unit testing, module testing, program testing
* Unit testing is performed by using white box testing methods
* The testing of individual software components
* Functional testing or non-functional
* Unit tests find problems early in the development cycle

Q.17 What is load testing

* Gradually increase the load on app then check speed the of the app( 1,2,3,4,to 100 gradually load on the app)
* It’s a performance testing to check system behavior under load
* Load testing is a kind a performance testing which determines a system performance under real-life load conditions
* Need for load testing
* Some extremely popular sites have suffered serious downtime when they get massive traffic volumes
* Ex an airline website was not able to handle 10000+ user during aa festival offer
* Why load testing
* Load testing gives confidence in the system & its reliability and performance
* Load testing helps identify the bottlenecks in the system under heavy user stress scenario before they happen in a production environment
* Goals of load testing
* Loading testing identifies the problems before moving the application to market or production
* Response time for each transaction
* Network delay between the client and the server
* Software design issues
* Pre-requisites for load testing
* Server machines
* Memory
* Disk storage
* Load machines configuration
* Network configuration
* Operating system
* Server software

Q.18 What is stress testing

* Suddenly increase/decrease the load on the app and check the speed of the app
* Performance under heavy load like putting large numbers beyond storage capacity
* Stress testing is done to make sure that the system would not crash under crunch situations
* It also checks whether the system demonstrates effective error management under extreme conditions
* Need for stress testing
* During the festival time an online shopping site may witness a spike in traffic or when it announces a sale
* To check whether the system works under abnormal conditions
* Goal of stress testing
* The behavior of the system after failure
* The main purpose of stress testing is to make sure that the system recovers after failure which is called recoverability
* Types of stress testing
* Application stress testing
* Transactions stress
* Systemic stress testing
* Exploratory stress testing

Q.19 explain types of performance testing

* **Load testing**
* **Stress testing**
* **Volumes testing**
* **Scalability testing**
* **Spike testing**

Performance problems

* Load a long time
* Load time is normally the initial time it takes an application to start
* This should generally be kept to a minimum
* Poor response time
* Response time is the time it takes from when a user input data into the application until the application outputs a response to that input

Performance testing examples

* Web based testing & desktop-based testing
* Check the page on slow connections
* Check data-based query time
* Mobile based testing
* Check data-based query time
* Check CPU and memory usage under peak load conditions

Q.20 What is Alpha testing

* It is always performed by the developer at the software development site
* Something it is also performed by the independent testing team
* Alpha testing is not open the market and public
* It conducted for software application and project
* It performed in a virtual environment
* Always performed in a virtual environment
* It always performed within the organization
* It is a form of acceptance testing
* It comes under the category of both white-box testing and black-box testing
* Alpha testing is definitely performed and carried out at the developing organization locations with the involvement of the developer

Q.21 What is beta testing

* It always performed by the customers at their own site
* Not performed by the independent testing team
* Always open to the market and public
* Usually conducted for software product
* Performed in a real-time environment
* Always performed outside the organization
* It is also the form of acceptance testing
* Beta testing can be considered “pre-release”
* It is a only kind of black box testing
* It is a performed in real time environment
* Beta testing is always performed at the time when software product and project are marketed
* It is also considered as the user acceptance testing (UAT) which is done at customers or user area

Q.22 what is priority?

* Priority describes the importance of the defect
* Defect priority status is the order in which defect should be fixed
* Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect.
* The priority status is set based on the customer’s requirements.
* Priority can be the following types
* **Po (High)**: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.
* The defect must be resolved immediately as it affects severely cannot be used until it is fixed
* **P1(Medium)**: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
* It can wait until a new version/build is created
* **P2(Low)**: The defect is an irritant that should be repaired, but repair can be deferred until after a more serious defect has been fixed.
* Developer can fix it in later releases

Q.23 what is the severity

* Severity describes the seriousness of defect and how much impact on business work flow
* Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software.
* For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.
* Severity can be of the following types:
* **Critical**: The defect that results in the termination of the complete system or one or more components of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.
* The main functionality is not working. Workflow is broken they cannot proceed further
* Ex: fund transfer is not working is net banking
* **Major (High)**: The defect that results in the termination of the complete system or one or more components of the system and causes extensive corruption of the data. The failed function is unusable but an acceptable alternative method exists to achieve the required results then the severity will be stated as major.
* **Moderate (Medium)**: The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.
* **Minor (Low)**: The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.
* It won’t cause any major break down of the system
* Ex: look and feel issues, spelling, alignment

Q.24 What is bug (defect) life cycle

* The duration or time span between the first-time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Defect Life Cycle’.
* Bug(Defect) Life Cycle

New

Assigned

---------- > Open --------------------------🡪 duplicated, rejected, deffered, not a bug

| Fixed

| Pending reset

Reopened ----🡪 Retest

Verified

Closed

* **New**: When a new defect is logged and posted for the first time. It is assigned a
* status as NEW.
* **Assigned**: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
* **Open**: The developer starts analyzing and works on the defect fix
* **Fixed**: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
* **Pending retest**: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”
* **Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* **Verified**: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
* **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.
* **Closed**: If the bug is no longer exists then tester assigns the status “Closed.”
* **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* **Deffered**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deffered” is assigned to such bugs
* **Not a bug**: If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

Q.25 mentions what are the categories of defect

* **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 the 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:

Buttons like Save, Delete, and Cancel not performing their intended functions

A missing functionality (or) a feature not functioning the way it is intended to

Continuous execution of loops

* **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.
* 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

Q.26 What is bug categories

* **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug it called duplicate
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* **Deffered**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release it called deffered
* **Not a bug**: If it does not affect the functionality of the application then it called not a bug

Q.27 Write agile manifesto principles

* **Customer satisfaction through early and continuous software delivery** – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
* **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.
* **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
* **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical team are aligned.
* **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
* **Enable face-to-face interactions** – Communication is more successful when development teams are co-located.
* **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
* **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
* **Attention to technical detail and design enhances agility** – The right skills and good design ensure the team can maintain the pace, constantly improve the product, and sustain change.
* **Simplicity** – Develop just enough to get the job done for right now.
* **Self-organizing teams encourage great architectures, requirements, and designs** – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
* **Regular reflections on how to become more effective** – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

Q.28 What is 7 key principles? explain in detail?

* **1. Show presence of error** = important to design test which find as many defects as possible
* **2. exhaustive testing is impossible** = for this reason risk and priorities used to concentrate important aspects to test
* Testing including all combinations of input and preconditions is not possible
* **3. Early testing** = testing activities should start early as possible. focused on defined objectives
* **4. Defects clustering** = a small number of modules contain most of the defects discovered pre-release testing most operating failure
* defects are not evenly spread in a system they are clustered
* **5. The pesticide paradox** = if the same tests are repeated over again the same set of test cases will no longer find any new defects
* If you keep running the same set of tests over and over again chances are no more new defects
* **6. Testing is context dependent** = different kind of sites are tested differently
* Ex: medical testing same testing in gaming that not possible in medical we all thing try to understand to small of small thing in gaming we should testing hardware, ram support to software that is context dependent
* **7. Absence of error fallacy** = all the test are done if there are no defects that not mean the software app are defects free or good in that cases your testing design does not match the software design
* We executed test design to catch most defects to see if the software matched the user’s requirements, then

Test cases upgrade

Software requirement

Factors

Test again software

Q.29 difference between verifications and validation

Verifications Validation

* Verifications check whether we are - validation check whether we are building the

Building the right product product right

* Focus on document - focus on actual software
* Process right or not - take place after verification are completed

Q.30.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
* Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means re-testing those part of application, which are unchanged

Q.31 what are Regression Testing Techniques

* Unit regression = testing only the change/modification done by developer
* Regional regression testing = testing modified along with impacted modules
* Full regression = testing main feature and remaining part of the application
* (E.x) dev has done changes in many modules instead of identifying we perform one round of full regression

Q.32 what is traceability value testing

* Make it obvious to the client that the software is being developed as per the requirements.
* To make sure that all requirements included in the test cases
* To make sure that developers are not creating features that no one has requested
* Easy to identify the missing functionalities.
* If there is a change request for a requirement, then we can easily find out which test cases need to update.
* The completed system may have “Extra” functionality that may have not been specified in the design specification, resulting in wastage of manpower, time and effort.

Q.33 what is the difference between the SDLC (software development life cycle) and SDLC (software development life cycle)?

| SDLC (software development life cycle) | STLC (software testing life cycle) |
| --- | --- |
| 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. |
| On both the development and testing process | On only testing process |
| In SDLC, more number of members (developers) are required for the whole process. | In STLC, less number of members (testers) are needed. |
| In SDLC, the 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. |
| A good quality software product | A bug free 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. |
| Development life cycle | Testing life cycle |

Q.34 explain the difference between Authorization and Authentication in web testing what are the common problems faced in web testing?

* Authentication: Accepting an invalid username/password
* Authorization: Accessibility to pages though permission not given

Q.35 explain what test plan is? what is the information that should be covered.

* A test plan is document that describe the test scope, test strategy, objectives, deliverables and resources requirement
* the test plan strategy along with efforts and cost estimates for the project.
* Moreover, the resources, test environment, test limitations and the testing schedule are also determined.
* The Test Plan gets prepared and finalized in the same phase.
* Below information should be covered
* Overview
* Scope
* Inclusions
* Test environment
* Exclusions
* Test strategy
* Defect reporting procedure
* Roles/responsibilities
* Test schedule
* Pricing
* Tools
* Approvals
* Risk and mitigations
* Test deliverables
* Test tools selections
* Test formations
* Test effort estimation