Q.1) Define Software Testing and explain why it is essential in the software development process.

→ Software testing is the process of evaluating a software application to identify any discrepancies between expected and actual results. It involves executing a program or system with the internet of finding errors to ensure its quality and reliability.
Testing is essential in software development for several reasons -

- Bug Detection - Testing helps identify and rectify bugs, errors, and defects in the software, preventing issues that could lead to system failures or malfunct.

- Quality Assurance - It ensures that the software meets the specified requirements and adheres to quality standards, enhancing the overall quality of the product.

- Customer Satisfaction - Through testing results in a more reliable and stable product, contributing to customer satisfaction by delivering software that performs as expected.

- Cost-Effectiveness - Identifying and fixing defects early in the development process is more cost-effectiv than addressing them later on after the software has been released.

• Risk Management - Testing helps in assesing and mitigating risks associated with the software, ensuring a more reliable and secure product.

• Compliance - For certain industries, adherence to regulatory standards and compliance requirement is crucial. Testing ensures that the software meets these standards.

• Continous Improvement - Testing provides feedb that can be used to improve the development process, leading to better software development practices and methodologies.

Q.2) List down the phases of Software Development Life Cycle.

→ The phases of the Software Development life cycle are -

i) Requirement gathering -
This is where the development team collects and analyzes the development requirements for the software.

ii) Designing -
In this phase, the team creates a blueprint for the software. They design the architecture, modules and interfaces, ensuring that it aligns with the requirements and is scalable and maintainable.

iii) Coding / Implementation -

This is where the actual development takes place. The developers write code based on the design specifications. They follow coding standards and best practices to create a functioning software product.

iv) Testing -

Once the coding is complete, the softwar goes through various testing processes. This includes unit testing, integration testing, system testing, and user testing. The goal is to identify and fix any bugs or issues before the software is deployed

v) Deployment -

After sucessful testing, the software is deployed to the production environment. It is made available to users and stakeholders. This phase involves activities like installation, configurat and data migration.

vi) Maintenance -

Once the software is live, it requires engoing maintenance and support. This includes bug fixes, updates and enhancements based on user feedback and changing requirements.

Page No.
Date

Q.3) Define "Quality" in the context of software development. How does quality relate to customer satisfaction and business success?

→ Quality refers to the overall excellence and reliability of a software product. It encompasses several aspects, including functionality, performance, usability, security, and reliability.

Quality is closely related to customer satisfaction and business success. When a software product is of high quality, it meets or exceeds customer expectations. It performs well, is user-friendly, and meets the intended purpose effectively. This leads to increased customer satisfaction, as users are more likely to have a positive experience and achieve their desired outcomes.

Customer satisfaction plays a crucial role in business success. Satisfied customers are more likely to become repeat customers, recommend the software to others, and contribute to positive word of mouth marketing. This can lead to increased sales, customer loyalty, and a stronger market presence.

Page No.
Date

Q.4) Define Verification and Validation (V & V) in the context of software testing.

→ Verification testing :-

It includes different activities such as business requirements, system requirements, design view, and code walkthrough while developing a product.

It also known as static testing, where we ensuring that "we are developing the right product or not". And it also checks that the developed application fulfilling all the requirements given by client.

Validation testing :-

It is a testing where tester performed functional and non-functional testing. Here functional testing includes Unit Testing (UT), Integration Testing (IT) and System Testing (ST), and Non-functional testing includes User acceptance testing (UAT). It is also known as dynamic testing.

Where we are ensuring that "we have developed the product right". And it also checks that the software meets the business needs of the client.

Q.5) Define the Error.

→ An error refers to a mistake or flaw in the code or logic of a program. Errors can occur during the code phase and can lead to unexpected behaviour or incorrect results when the program is executed.

Page No.
Date

(Q.6) What are the goals of software testing?

→ • Functionality – The software should perform its intended functions accurately and reliably.

• Reliability – The software should consistently operate without failures and errors, even under varying conditions.

• Performance – The software should respond quickly and efficiently, even when subjected to heavy loads or stress.

• Security – The software should protect against unauthorized access, data breaches, and other security vulnerabilities.

• Usability – The software should be user-friendly and intuitive, allowing users to accomplish task with eas

• Maintainability – The software's code should be structure and documented well, making it easy to maintain an enhance in the future.

• Scalability – The software should be able to accomodate increasing workloads and user demands without significant performance degradation.

• Cross Compatibility – The software should work seamlessly on different devices, operating systems, and browse

- Portability – The software should be easily transferable between different environments and platforms.

Q.7) What do you mean by software Test case?

→ A test case is a document, which has a set of test data, preconditions, expected results and post condition, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

Q.8) List the different types of Black Box Testing.

→ Black Box Testing –
i) Functional Testing
ii) Non-Functional Testing.

i) Functional Testing –
- Unit testing.
- Integration testing.
- System testing.
- User Acceptance testing.

ii) Non-Functional testing –
- Performance testing.
- Usability testing.
- Compatability testing.

Page No.
Date

Q.9) List the advantages of software Testing.

→ i) Customer satisfaction.
ii) cost effective.
iii) Quality product.
iv) Low failure.
v) Bug free Application.
vi) Security.
vii) Easy Recovery.
viii) Speed up the Development process.
ix) Early defect Detection.
x) Reliable Product.

Q.10) What is Test case in software Testing ?

→ A Test case is defined format for software Testing required to check if a particular application / software is working or not. A Test case consists of a certain set of conditions that need to be checked to test an application or software i.e. in more simple terms when conditions are checked it checks if the resultant output meets with the expected output or not. A Test case consists of various parameters such as ID, condition, steps, input, expected result, actual result, status, and remarks.

Page No.
Date

Q.11) Why it is necessary to perform Unit Testing in Software Testing?

→ Unit Testing is a straightforward yet crucial part of Software development. It involves testing the most minor parts of an application, called units, to ensure they work correctly these units are often individual functions or section of codes.
The essence of unit testing lies in its ability to meticulously scrutinize, the most minor parts of an application, ensuring they coork flawlessly on their own. This early-stage testing is important for several reasons.

- Early Bug Detection.
- Code Quality Improvement.
- Facilitates Refactoning.
- Documentation.

Q.12) Explain the importance of having a strategic approach to software testing.

→ The Software testing strategy is an essential process that comprises part of the software development lifecycle. Companies that fail to implement quality control standards and adequately define the range of tests for an application can destroy brand credibility, sabotage the overall project and create a cost blowout. The test plan forms part of the project documentation. The goals, objectives and functional requirements of a software application are scoped and bounded by the project plan.

Page No.
Date

Q.13) Define Integration Testing.

→ Integration testing is the phase in software testin in which the whole software module is based tested or if it consists of multiple software modules they are combined and then tested as a group. Integration testing is conducted to evaluate to compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing.

Q.14) What do you mean by Validation Testing?

→ The process of evaluating software during the development process or at the end of the developmen process to determine whether it satisfies specified business requirements.
Validation Testing ensures that the product actually meets the client's need it can also be defined as to demonstrate that the product fulfills its intended use when deployed. It answers to the questions, are we building the right product.

Q.15) Define Verification Testing.

→ Verification testing includes different activities such as business requirements, system requirement design review, and code walkthrough while developing a product. It also known as Static testing, where we are developing the right product or not. And also checks that the developed application fulfilling all requirements by the client.

Page No.
Date

Q.16) Differentiate between Validation Testing and other types of testing.

→

Verification Testing	Validation Testing.
i) Focuses on whether the software meets its specified requirements.	Focuses on whether the software meets the needs of end-users.
ii) Typically conducted during the development process.	Typically conducted after the development process
iii) Involves testing with sample data	Involves testing with real-world data.
iv) Helps to ensure that the software is built consistently and correctly.	Helps to ensure that the correct software is built.
v) Improves the quality of the software	Reduces the risk of product recall or legal issues.
vi) Examples include unit testing, integration testing and system testing.	Examples include user-acceptance testing, alpha and beta testing, and compatability testing.

Page No.
Date

Q.17) What do you mean by System Testing ?

→ System testing is a type of software testing that evaluates the overall functionality and performanc of a complete and fully integrated software soluti It tests if the system meets the specified requirem and if it is suitable for delivery to the end-users. This type of testing is performed after the integrati testing and before the acceptance testing.

Q.18) What do you understand by software metrics.

→ A software metric is a measurement of the level at which any impute belongs to a system product or process. It is a quantifiable or countable assesment of the attributes of a software product. There are 4 functions related to software metrics :-

i) planning.
ii) organizing.
iii) controlling.
iv) Improving.

Q.19) What is Defect ?

→ A defect refers to any flaw, error, or imperfection in a software product that deviates from its intended behaviour or specification. Defects are also commonly known as bugs or issues. These can manifest in various forms, including coding errors, design flaws or misunderstandings of requirements.

Page No.
Date

Q.20) What do you mean by product matrices?

→ Product matrices are quantative measures used to assess various aspects of a software product. These metrics can include lines of a code, defect density, response time, user satisfaction scores, and more. This is known as product metrics.

Q.21) What is Process metrics?

→ Process metrics are quantitative measures used to assess and evaluate various aspects of the software development process. These metric provide insights into the efficiency, effectiveness, and quality of the processes employed throughout the software development life cycle. This is known as Process Metrics.

Q.22) What do you mean by Quality Models and Measures?

→ Quality Models and measures are used to assess and manage the quality of software products. These models provide a structured framework for evaluating various attributes and characteristic of software, ensuring that it meets specified requirements and user expectations. Quality measures are quantative indicators that help gauge the degree to which these attributes are achieved. It measures the defect density, code coverage, Response time, customer satisfaction, Mean Time between Failures (MTBF), Maintainability Index, Usability Metrics, Scalability.

Q.23) Explain the importance of having a strategic approach to software testing.

→ The main objective of software testing is to design the tests in such a way that it systematically finds different types of errors without taking much time and effort so that less time is required for the development of the software. The overall strategy for testing software includes -

- Developing continous development approach.
- Conducting formal technical reviews.
- Using effective formal reviews.
- Build Robust Software.
- Developing a test plan.
- Identifying and developing users's profile.
- Specifying the objectives of testing.
- Specifying the product requirements.

Q.24) What do you understand by SQA ?

→ Software Quality Assurance (SQA) is simply a way to assure quality in the software. It is the set of activities which ensure process, prodecures as well as standards are suitable for the project and implemented correctly. It is a process which works parallel to development of software. It focuses on improving the process of development of software so that problems can be prevented before they become a major issue.

Page No.
Date

Q.25) What is the purpose of SQA?

→ The purpose of Software Quality Assurance is –
- Improve Development Processes.
- Ensure Compliance.
- Mitigate Risks.
- Enhance Product Quality.
- Cost Reduction.
- Increase Customer Satisfaction.
- Facilitate Continous Improvement.
- Build Confidence.

Q.26) List the types of Software Reviews.

→ A software review is a task in which a group of people tries to resolve the errors and defects in particular software.
The types of software reviews are :-
- Software peer reviews.
- Software management reviews.
- Software audit reviews.

Q.27) What are the 6 Sigma principles?

→ The 6 Sigma principles involve defining project goals, measuring current process performance, analyzing data to identify root causes, making improvements, implementing controls to sustain improvements, and verifying success through ongoing monitoring. The goal is to enhance process efficiency, reduce defects, and meet customer requirements.

Page No.
Date

Q.28) What are the different feature of ISO 9000 Requirements ?

→ The Features of ISO 9000 Requirements are :-
- Document control.
- Planning.
- Review.
- Testing.
- Organizational Aspects.

Q.29) Describe the Software Development Life Cycle (SDLC) and its various phases. How does the SDLC contribute th overall quality of a software product?

→ Software Development Life Cycle (SDLC) is a process used in software development to plan, design, build, test, and maintain software systems. The phase of SDLC include :-

i) Requirements Gathering - Collecting and documenting the software requirements.

ii) System Design - Creating a high-level design and architecture for the software.

iii) Coding - Writing the actual code for the software based on the design.

iv) Testing - Conducting various tests to ensure the software functions correctly.

v) Deployment - Releasing the software to users or customers.

vi) Maintenance - Making updates and enhancements to the software based on user feedback.

When the Software Development Life Cycle (SDLC) is executed effectively, contributes to overall software quality by promoting systematic development, early defect identification, adherence to requirements, and continous improvement. It ensures that the software is not only functional but also reliable, maintable, and aligned with user expectations, thereby enhancing the user experience and satisfaction.

Q.30) Difference between Quality Assurance (QA), Quality control (QC).

→

Quality Assurance (QA)	Quality control (QC).
• It is a procedure that focuses on providing assurance that quality requested will be achieved.	It is a prodecure that focuses on fullfilling the quality requested.
• QA aims to prevent the defect.	QC aims to identify and fix.
• It's a preventive technique	It's a corrective technique.
• It's a proactive measure	It's a reactive measure.

• It is the procedure to create the deliverables.	It is the procedure to verify that deliverables.
• QA involves in Full software development life cycle.	QC involves in Full software testing life cycle.
• In order to meet the customer requirements, QA defines standards and methodologies.	QC confirms that the standards are followed while working on product.
• Its main motive is to prevent defects in the system. It is a less time-consuming activity	It's main motive is to identify defects or bugs in system. It is a more consuming activity.
• QA ensures that everything is executed in the right way, and that is why it falls under verification activity.	QC ensures that whatever we have done is as per the requirement, and that it falls under validation acti
• It requires the involvement of the whole team.	It requires the involvemen of the testing team.
• The statistical technique applied on QA is known as SPC or Statistical Process Control (SPC).	The statistical technique applied to QC is known as Sqc or statistical quality Control (SQC).
• It is performed before quality Control	It is performed only afte QA is activity is done.

Page No.
Date

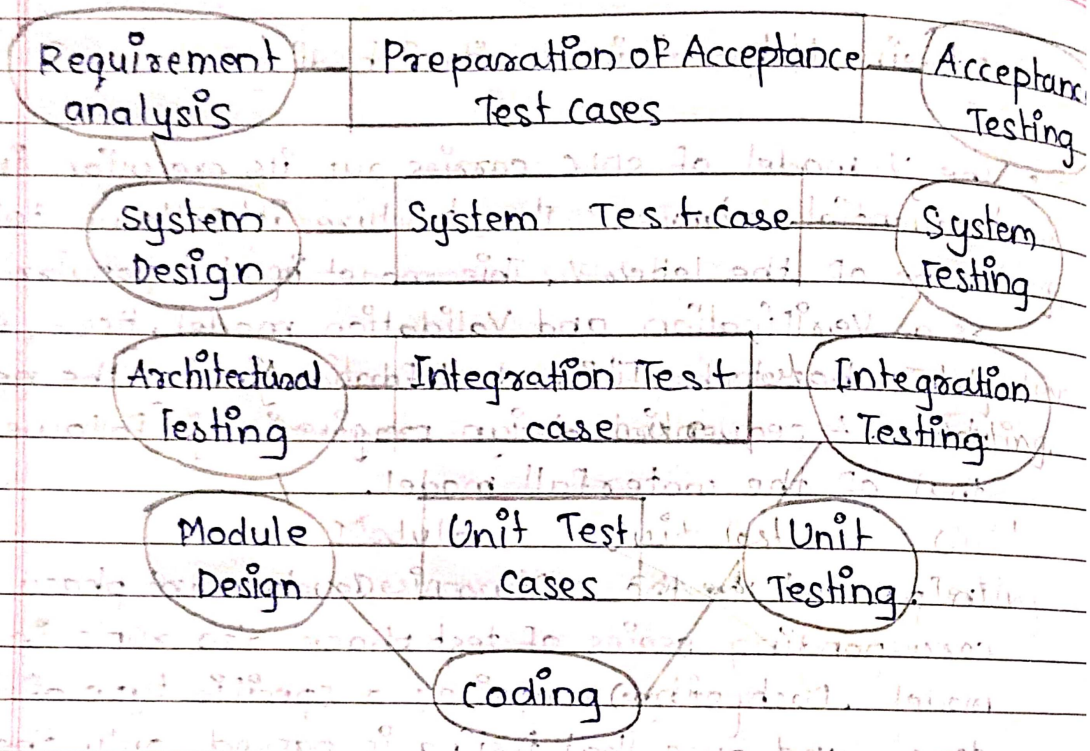
Q.31) Explain V model in detail with diagram.

→ The V-model of SDLC carries out its execution in a sequential manner. The structure it follows takes the shape of the letter V. This model is also popularly termed as a Verification and Validation model. Here, each phase has to be finished before beginning the next phase. A sequential design progression is followed like that of the waterfall model.

In parallel to the software development phase, a corresponding series of test phase also runs in this model. Each stage comprises a specific type of testing done, and once that testing is passed, only then the next phase starts.

- Verification – In the concept of verification in the V-model, static analysis technique is carried out without executing the code. This evaluation procedure is carried out at the time of development to check whether specific requirements will meet or not.
- Validation – This concept of V-model comprises of dyanmic analysis practice (both functional as well as non-functional), and testing is done by code execution. The validation of a product is done once the development is complete for determining if the software meets up the customer hope needs.

So both verification and validation are combined up and work in parallel to make the V-model fully functional.



Q.32) Explain the concepts of Inspection and Walkthrough as methods for improving software quality. Highlight the key differences between these processes.

→ Inspection – An inspection is defined as formal, rigorous, in depth group review designed to identify problems as close to their point of origin as possible. Inspections improve reliability, availability, and maintainability of software product.

Walkthrough – It is method of conducting informal group / individual review. Author describes and explains work product in a formal meeting to his peers or supervisor to get feedback. Here, validity of the proposed solution for work product is checked.

The key differences between the processes are :-

i) Formality - Inspection is a formal and structured process with predefined steps and checklists. Walkthroughs are less formal and more flexible, encouraging open discussion and collaboration.

ii) Participants - Inspections typically involve a larger team, including authors and various stakeholders. Walkthroug involve a smaller, more casual group, focusing on knowledge sharing and early feedback.

iii) Process - Inspections follow a rigorous process aimed at defect detection and adherence to standards. Walkthroughs are more about understanding and collaboration, with an emphasis on obtaining feedback and clarifying information.

iv) Timing - Inspections can be conducted at different stages of development, while walkthroughs are often used early in the process to gather input and ensure shared understanding.

Page No.
Date

Q.33) List and briefly explain three software Quality factors that are crucial in evaluating the quality of a Softwar application ?

→ • Functionality – It is a critical software quality factor that assesses the extent to which the software meets its specified requirements and performs the intended functions. It involves evaluating features, capabilities, and correctness of the software. A high-quality softwa application should provide accurate and complete functionality according to user expectations.
• Reliability – It focuses on the software's ability to perform consistently and predictaly under various conditio It includes aspects such as stability, fault tolerance, and the ability to recover from failures. A reliable software application should minimize the occurence of defects, crashes, and unexpected behaviour, ensuring a dependable user experience.
• Usability – Usability is the measure of how user-friendly and easy to use a software application is. It considers factors such as user interface design, intuitiv ness, and overall user experience. A high-quality softw application should be designed with the end-user in mind, promoting efficiency, satisfaction, and ease of learning and use.

These quality factors are interconnected, and achieving a balance among them is crucial for delivering a softw are product that not only meets technical specification but also satisfies under users needs and expectation

Page No.
Date

Q.34) Explain the different types of Software Testing with advantages and disadvantages.

→ There are two types of Software Testing -

• Manual Testing -

The process of checking the functionality of an application as per the customer needs without taking any help of automation tools is known as Manual Testing. While performing the manual testing on any application, we do not need any specific knowledge of any testing tool; rather than have a proper understanding of the product so we can easily prepare the test document.

Manual Testing can further divided into three types -

- White box Testing.
- Black box Testing.
- Gray box Testing.

• Automation Testing -

It is a process of converting any manual test cases into the test scripts with the help of automation tools, or any programming language is known as automation testing. With the help of automation testing, we can enhance the speed of our test execution because here, we do not require any human efforts. We need to write a test script and execute those scripts.

Advantages of Software Testing -

i) Improves software quality and relability.
ii) Enhances user experience.
iii) Increase confidence.

iv) Facilitates maintenance.
v) Reduce costs.

Disadvantages of Software Testing-
i) Time Consuming.
ii) Resource-intensive.
iii) Limited coverage.
iv) Unpredictable results.
v) Delays in delivery.

Q.35) Explain the different techniques used in White Box Testing in details?

→ Techniques used in White Box Testing are -

- Data Flow Testing - Data Flow Testing is a group of testing strategies that examines the control flow of programs in order to explore the sequence of variables according to the sequence of events.

- Control Flow Testing - It determines the execution order of statements or instructions of the program through a control structure. The control structure of a program is used to develop a test case for the program. In this technique, a particular part of a large program is selected by the tester to set the testing path. Test cases represented by the control graph of the program.

- Branch Testing - This technique is used to cover all branch of the control flow graph. It covers all the possible outcomes (true and false) of each condition of decision.

• Statement testing = Statement coverage technique is used to design white box test cases. This technique involves execution of all statements in the source code at least once. It is used to calculate the total number of executed statememels in the source code, out of total statements present in the source code.

• Decision Testing = This technique reports true and false outcomes of Boolean expressions. Whenever there is a possibility of two or more outcomes from the statements like do while statements, if statement and case statement, it is considered as decision point because there are two outcomes either true or false.

(Q.86) Put down the difference between White Box and Black Box Testing?

→

Black Box Testing	White Box testing.
• It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it.	It is a way of testing the software in which the tester has knowledge about the internal structure of the code or the program of software.
• Implementation of code is not needed for black box testing	Code Implementation is necessary for White box testing.
• It is mostly done by software testers.	It is mostly done by software developers.

• No knowledge of implementation is needed.	Knowledge of implementation is required.
• It can be reffered to as outer or external software testing.	It is the inner or the internal software testing.
• It is a functional test of the software.	It is a structural test of the software.
• This testing can be intiated based on the requirement specifications document.	This type of testing of software is started after a detail design document.
• No knowledge of programming is required.	It is mandatory to have knowledge of programming.
• It is the behaviour testing	It is the logic testing.
• It is applicable to the higher levels of testing of software	It is generally applicable to lower levels of software test
• It is also called closed testing	It is also called clear box.
• It is least time consuming.	It is most time consuming.
• It is not suitable or preffered for algorithm testing.	It is suitable for algorithm testing.
• Can be done by trial and error ways and methods.	Data domains along with inner or internal boundaries can be better tested.

Page No.
Date

Q.37) Write a test case design for Appointment in Hospital in detail.

→ → Install Application
→ For apollo Hospital.

→ Check For the Appointment
— Name
— Mobile no.
— Date
— city
— Disease Specialist.

→ Appointment confirmation message
→ Message regarding Appointment.

Header :-

Test -Case / ID-Name	:- Appollo Hospital
Test . Case Type	:- FTC / NFTC
Requirement No.	:- 01.
Module	:- Delta
Severity	:- Critical
Release	:- Third.
Version	:- 3.0
Pre-condition	:- You should have installed application.
Test data	:- Check hospital appointment.
Summary	:-

Page No.
Date

Body :-

Step no.	Description	Input	Expected Result :	Actual Result	Status	Remark
01	Go to the playstore	Apollo Hospital	Open appollo Hospital app.	Open appollo Hosp. app	Positive +ve	complete with +ve
02	Enter on login Button and Enter User ID & password	User - 1234 Pass - abcd	Should open the personal account	open personal account	+ve	+ve
03	Check for appointment	i) name ii) mobileno. iii) Date iv) city v) Disease vi) specialist	open name open mobile open calender open city opt. open disease list open specialist	open name open mobile open calender open city open disease list open name specialist	+ve +ve +ve +ve +ve +ve	+ve +ve +ve +ve +ve +ve
04	Enter on log out Button	click on log out	Exit From application	Exit From application	+ve	+ve
05	Close the app	click exit	Exit application	Exit application	+ve	+ve

Footer :-
Author :- Raj
Review by :- Pushkar Jane
Date :- 13/12/2023

Page No.
Date

Q.38) Write a Test Case design for Facebook Login.

→ Header :-

Test Case / ID Name :- Facebook Login
Requirement No. :- 01
Module :- Alpha
Severity :- major
Status :- positive / negative
Rebase :- Second
Version :- 1.0.
Pre-condition :- Required URL / You should have install app.
Test Data :- User Name - 1234 Password - Abcd.
Summary :-

Body :-

Step No	Description	Input	Expected Result	Actual Result	Status	Remark
01	Enter the URL of facebook login	WWW. facebook .com	Home page of facebook	Home page of facebook	Positive +ve	Positive +ve.
02	Enter on login button and enter user id & password	i) user -1234 pass-Abcd	should open the persond account	open personal account	+ve	+ve
		ii) user- Blank pass- wrong	Error	Error	-ve	-ve
		iii) user- correct pass- wrong	Error	open personal account	-ve	-ve
		iv) user- wrong pass- wrong	Error	Error	+ve	+ve
03	Enter on log out button	click on Logout	Exit from application	Exit from application	+ve	+ve.

Footer :-

Author :- Raj
Date :- 13/12/2023

Review by :- Pushkar Jane.

Page No.
Date

Q.39) Explain Strategic Approach to Software testing?

→ The main objective of an strategic approach to software testing involves planning, designing, and executing testing activities in a systematic and efficient manner to ensure the delivery of a high-quality softwar product. Here are key components of a strategic approach to software testing -

- Before testing starts, it's necessary to identify and specify the requirements of the product in a quantifiable manner.
- Specifying the objectives of testing in a clear and detailed manner.
- For the software, identifying the user's category and developing a profile for each user.
- Developing a test plan to give value and focus on rapid-cycle testing.
- Robust software is developed that is designed to test itself.
- Conduct formal technical reviews to evaluate the nature quality or ability of test strategy and test cases.
- Before testing, using effective formal reviews as a filter.
- For the testing process, developing a approach for the continous development.

Q.4a) Explain Unit testing in detail with objective, advantages & disadvantages.

→ Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers and it is performed early in the development process before the code is integrated and tested as a whole system.

The objective of Unit Testing is –

- To isolate a section of code.
- To verify the correctness of the code.
- To test every function and procedure.
- To fix bugs early in the development cycle and to save costs.
- To help the developers to understand the code base and enable them to make changes quickly.
- To help with code reuse.

Advantages of Unit Testing –

i) Early Issue Detection.
ii) Improved code quality.
iii) Facilitates Refactoring.
iv) Documentation and Examples.
v) Saves Time in the long Run.
vi) Supports continous Integration.
vii) Isolation of Defects.
viii) Encourages Modular Design.

Page No.
Date

Disadvantages of Unit Testing -

i) Incomplete Test Coverage.
ii) Time - Consuming.
iii) Dependacy on Implementation Details.
iv) Not a Silver Bullet
v) Resistance From Developers.
vi) Maintenance overhead.
vii) False Sense of Security.
viii) May Miss Integration Issues.

Q.4) Explain different types of Integration Testing in detail.

→ The types of Integration Testing are -

• Big Bang Testing -

This is when all the components are tested together at once. It can be quick, but it's risky because if any issues arise, it can be hard to pinpoint the cause.

• Top-Down Testing -

In this approach, testing starts with the highest level components and gradually moves down to the lower-level ones. It helps identify major issues early on.

• Bottom - Up Testing -

This is the popposite of top-down testing. It starts with the lower-level components and gradually moves up to the higher-level ones. It's useful for identifying issues in individual components.

Page No.
Date

• Sandwich Testing -

It combines both top-down and bottom-up approaches. Testing starts from the top-bottom simultaneously, with the aim of meeting in the middle. It helps find integration issues at different levels.

• Mock Testing -

In this testing, mock objects are used to simulate the behaviour of components that are not yet available. It allows testing to proceed even if some components are still under development.

Q.42) Compare the Validation Testing and Verification Testing with their pros & cons.

Verification Testing	Validation Testing
• We check whether we are developing the right product or not.	We check whether the developed product is right.
• Verification is also known as Static Testing.	Validation is also known as dynamic testing.
• Verification includes different methods like inspections, Reviews, and Walkthroughs.	Validation includes testing like functional testing, system testing, integration, and user acceptance testing.

• It is a process of checking the work-products (not the final product) of a develop ment cycle to decide the whether the product meets the specified requirements.	It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specifi- ed business requirements.
• Quality assurance comes under Verification testing.	Quality control comes under validation testing.
• The execution of code does not happen in the verifica- tion testing.	In validation testing, the execution of code happens.
• In verification testing, we can find the bugs in early in the development phase of the product.	In validation testing, we can find those bugs which are not caught in the verification process.
• Verification testing is executed by the quality assurance team to make sure that the product is developed according to customer's requirements.	Validation testing is executed by the testing team to test the application
• Verification is done before the validation testing	After verification testing, validation testing takes place.
• In this type of testing, we can verify that the input follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.

Verification :-
Pros -
- Early Issue Identification.
- Consistency.
- Stability.

Cons -
- Not comprehensive.
- May Miss User Expectations.

Validation :-
Pros -
- User Centric.
- Comprehensive.
- Customer satisfaction.

Cons -
- Late Identification of Issues.
- Costly to fix Defects.

Q.43) Explore the challenges that may arise during system testing, particularly when testing the system as a whol

→ System testing involves assessing the entire system's functionality to ensure it meets specified requirements. challanges during this phase may include:-
i) Integration Issues - Ensuring seamless collaboration among different components can be challenging, especially when integrating various modules or third-party streams.

ii) Data Management - Handling diverse data sets and ensuring proper data flow throughout the system can be complex, leading to issues like data corruption or loss.

iii) Dependacy Management - Identifying the and managing dependencies among different modules or system is crucial. Changes in one area may impact others, requiring careful co-ordination.

iv) Performance Bottlenecks - Discovering and addressing ~~#p~~ performance issues, such as slow response times or resource constraints, becomes critical during system testing.

v) Scalability Concerns - Verifying that the system can handle increased loads and user volumes is challenging, as it requires simulating real-world scenarios and scaling resources accordingly.

vi) User Interface (UI) Challenges - Ensuring a consistent and user-friendly interface across the entire system can be complex, especially when dealing with diverse function and user roles.

vii) Security Vulnerabilities - Identifying and rectifying security loopholes is crucial. System testing should include through security assessments to protect against potential threats

viii) Error Handling and Logging - Ensuring robust error handling mechanisms and effective logging throughout the system is challenging but esstial for diagnosing issues.

ix) Regression Testing - As the system evolves, ensuring that new changes do not negatively impact existing functionalities can be time-consuming but is crucial to maintain system stability.

x) Documentation Accuracy - keeping documentation up-to-date can be challenging, but it is essential for future reference and troubleshooting.

Q.44) What are Goals of Defect Management Process (DMP)?

→ The goals of a Defect Management process include -

i) Early detection of Defects - Detect and identify defects as early as possible in the SDLC to minimize the impact on subsequent phases.

ii) Efficient Logging and Tracking - Log defects systematically, providing detailed information about the issue to facilitate efficient tracking, analysis and resolution.

iii) Root Cause Analysis - Conduct thorough investigation to determine the root cause of defects. This helps in addressing underlying issues to prevent similar defects in the future.

iv) Timely Resolution - Aim for timely resolution of defects to prevent project delays and maintain the overall project schedule.

v) Quality Improvement – Use defect data to identify patterns and areas for improvement in the development and testing processes, contributing to overall software quality enhancement.

vi) Process Improvement - Review and enhance the defect management process based on lessons learned, evolving project requirements, and feedback from ongoing projects.

By achieving these goals, the Defect Management process contributes to overall success of a software development project by enhancing product quality, reducing risks and promoting efficient collaboration.

Q.45) Define software metrics and explain their significance in the software development life cycle.

→ Software metrics are quantifiable measures used to evaluate various aspects of the software development process, software products, and the efficiency of software teams. These metrics provide objective data that can be analyzed to assess the quality, performance and progress of software development.

Significance in the Software Development Life Cycle –

i) Quality Assessment - Metrics help assess the quality of the software by measuring factors such as defect density, code coverage, and adherence to coding standards.

ii) Productivity Measurement - Metrics like lines of code written per hour or story points completed help guage the productivity of development teams. This information is valuable for project planning, resource allocation, and optimizing workflows.

iii) Effort Estimation - Historical data and metrics are used to estimate the effort required for similiar tasks in future projects.

iv) Risk Management - Metrics related to defect trends, volatility in requirements, and other factors help identif and manage potential risks early in the development process, allowing for proactive risk mitigation strategies

v) Resource Allocation - Metrics assist in allocating resources effectively by providing visibility into where tim and effort are most needed. This is crucial for optimizin team performance and meeting project deadlines.

vi) Code Review and Maintenance - Metrics related to code complexity, maintainability, and dependencies help guide code review processes.

vii) Customer Satisfaction - Metrics related to defect resolution time, response time to customer issues, and overall product performance contribute to assessing customer.

Q.4a) What are the defect metrics ? Also explain Defect report.

Defect Metrics are quantitative measure used to assess the quality and reliability of a software product by tracking and analyzing defects or issues identified during testing and development. Here are some common defect metrics -

i) Defect Density.
ii) Defect Removal Efficiency (DRE).
iii) Open Defect count.
iv) Defect Age.
v) Defect Arrival Rate.
vi) Defect closure Rate.
vii) Defect Rejection Rate.
viii) Defect severity Distribution.
ix) Defect Trend Analysis.
x) Defect cost
xi) Defect Aging Index.

Defect Report -

A defect report is a document detailing a software issue, including a unique ID, summary, description, severity, steps to reproduce, environment details, attachments, reporter information, assigned resolution, and current status. It aids communication among team members for efficient defect management.

Page No.
Date

Q.47) What do you understand by Process Improvement in Defect Process (DMP)?

→ Process improvement in Defect Management Process (DMP) involves identifying, analyzing, and enhancing the procedures and workflows associated with managing defects or issues in a software development or operational environment. This typically includes refining how defects are reported, tracked, prioritized, and resolved throughout the development lifecycle.
Key aspects of process improvement in DMP may include :-

i) Efficiency.
ii) Visibility.
iii) Collaboration.
iv) Root Cause Analysis.
v) Automation.
vi) Metrics and Reporting.
vii) Feedback Loop.

Overall, process improvement in DMP is about creating a more effective and efficient system for identifying, addressing, and preventing defects, ultimately contributing to higher-quality software and improved development practices.

Page No.
Date

Q.48) Discuss the Relationship between code complexity and software maintainability.

→ The relationship between code complexity and software maintainability is crucial. As code complexity increases software maintainability decreases. Complex code can be harder to understand, modify, and debug, making it more prone to errors during maintenance. Simpler, well-organized code is easier for developers to comprehend and update, facilitating quicker and more accurate modifications. This, in turn, enhances software maintainability by reducing the likelihood of introducing bugs and improving the efficiency of future development efforts.

Strategies such as modularization, adherence to coding standards, and consistent documentation can help mitigate code complexity and contribute to better software maintainability in the long run.

Q.49) Explain the major activities involved in Software Quality Assurance.

→ The Major Activities involved in SQA is -

i) SQA Management Plan - Make a plan for how you will carry out the SQA throughout the project. Think about which set of software engineering activities are the best for project. check level of SQA team skills.

ii) Set the check Points - SQA team should set check points. Evaluate the performance of the project on the basis of collected data on different check points.

iii) Multi testing Strategy - Do not depend on a single testing approach. When you have a lot of testing approaches available use them.

iv) Measure Chang Impact - The changes for making the correction of an error sometimes are introduces more errors keep the measure of impact of change on project. Reset the new change to change check the compatibility of this fix with whole project.

v) Manage Good Relations - In the working environmer managing good relations with order teams involved in the project development is mandatory. Bad relatio of SQA with programmers team will impact directly and badly on project.

Q.50) Explain Software Reviews in delaits along with objective & advantages.

→ Software reviews are an essential part of the SDLC. A software review is a task in which a group of people tries to resolve the errors and defect in particular software. It plays an important role in developing any software model.

Objectives of a software review.-

- We can use software reviews to discover the expectations that clients have from us.
- We can use them to identify and eliminate any errors and defects in the software.
- We can utilize them to reduce the gap between the client's demands and the developer's supply of products.

Advantages of Software Reviews -

i) It helps in early detection of Issues in software
ii) It helps in Transfer of knowledge.
iii) Improved collaborations.
iv) It gives the quality Assurance of software.
v) Risk Mitigation.
vi) Feedback from reviews provide valuable insights for continous improvement.

Q-8) Explain Statistical Quality Assurance in details.

→ Statistical Quality Assurance (SQA) is a set of systematic and statistical methods used to monitor, analyze and improve the quality of processes and products in various industries. It involves the applica of statistical techniques to ensure consistency, reliability, and conformity to established standards, ultimately aiming to enhance overall product or service quality. SQA encompasses activities such as statistical process control, data analysis, and quality management to optimize processes and minimize variations in production or service delivery.

Methods of Statistical Quality Assurance (SQA) –

i) Methodical Approach – It adopts a systematic meth
ii) Statistical Tools – Utilize tools like control charts an hypothesis testing.
iii) Consistency – Aims for consistend and reliable outcomes by minimizing variations.
iv) Data-Driven Decisions – Relies on data analysis for informed decision making.
v) Continous Improvement – Emphasizes ongoing enhancements in processes.
vi) Compliance – Ensures adherence to standards and specifications.
vii) Customer stisfaction – contributes to customer satisfaction through quality products.
viii) Risk Reduction – Identifies and mitigates risks associated with process variations.
ix) Integration – Collaborates with quality control for a comprehensive approach.

Page No.
Date

Q.52) Explain Formal Technical Reviews (FTR) in software Quality Assurance.

→ • Formal Technical review is a software quality assurance activity performed by software engineer.
• The purpose of FTR is to enable junior to observer the analysis, design, coding and testing approach more closely.
• FTR is useful to uncover error in logic, function and implementation for any representation of the Software.
• It ensure that software meets specified requirements.
• It also ensure that software is represented according to predefined standards.
• It helps to review the uniformity in software development process.
• It makes the project more manageable.
• Each FTR is conducted as meeting and is considered successfully only if it is properly planned, controlled and attended.

Q.53) Explain ISO 9000 standards in software testing in details.

→ ISO 9000 standards are set of International quality Management standards designed to ensure that organizations meet certain quality and customer satisfaction criteria. When applied to software testing, ISO 9000 standards help establish a framework for effective quality management.

Features of ISO 9000 Requirements -

- Document control -
  All documents concerned with the development of a software product should be proper managed and controlled.

- Planning -
  Proper plans should be prepared and monitored

- Review -
  For effectiveness and correctness all important documents across all phases should be independently checked and reviewed.

- Testing -
  The product should be tested against specificat

- Organizational Aspects -
  Various organizational aspects should be addressed e.g., management reporting of the quality team.

Page No.
Date

Q.54) Provide two challenges that may arise during integration testing and suggest strategies to overcome them.

→ challenge 1 - Dependency Issues.

- Issue - It involves combining different modules or components, and dependencies between them can lead to challenges. Change in one module may impact others, causing unexpected failures.
- strategy - Use stubs or mocks to simulate the behavior of dependent modules. This allows you to isolate the module being tested and identify issues without interference from external dependencies.

challenge 2 - Data Consistency.

- Issue - Ensuring consistent data flow between integrated components can be challenging. Mismatched data formats or communication protocols may result in errors during integration.
- Strategy - Establish a standardized data format and communication protocol across modules. Implement through data validation checks during integration testing to catch inconsistencies early. Additionally use realistic test data that mirrors actual production scenarios to uncover potential issues.