What is Test Case Design Technique?

Test case design techniques refer to systematic methods or strategies used to create and identify test cases for software testing. These techniques help testers to ensure that the software application under test is performing as expected and meeting the requirements.

There are various test case design techniques, some of the popular ones include:

1. Equivalence partitioning: This technique divides the input domain of a software application into different partitions or sets, where each set represents a different behavior of the application.
2. Boundary Value Analysis: This technique focuses on the input values that are on the edge or boundary of the input domain. These values are more likely to cause errors or faults in the software application.
3. Decision Table Testing: This technique is used to test the logic of the software application. It involves creating a table that lists all possible combinations of inputs and their corresponding outputs.
4. State Transition Testing: This technique is used to test the behavior of the software application when it transitions from one state to another. It involves identifying the different states of the application and testing the behavior of the application when it moves from one state to another.
5. Error Guessing: This technique is used to identify test cases based on the tester's experience and intuition. It involves guessing the types of errors that are likely to occur in the software application and designing test cases to test those areas.

The selection of a test case design technique depends on various factors such as the type of software application, the complexity of the application, and the level of detail required for

What is software testing life cycle

Software Testing Life Cycle (STLC) is a process that describes the various stages involved in testing software. It is a structured approach to testing that ensures that software meets the expected quality standards and fulfills the customer requirements. The following are the various phases of the STLC:

1. Requirement Analysis: In this phase, the testing team reviews the requirements to understand the features and functionalities of the software. Based on the requirements, they create test cases and scenarios.

.

1. Test Planning: This phase involves creating a detailed test plan that outlines the testing objectives, test approach, test environments, and test schedules. The test plan also defines the roles and responsibilities of the testing team.
2. Test Design: In this phase, the testing team designs the test cases and scenarios based on the requirements. The test cases are designed to cover all possible scenarios and to ensure maximum test coverage.
3. Test Environment Setup: The testing team sets up the test environment, which includes the hardware, software, and network configurations required for testing.
4. Test Execution: The actual testing of the software is done in this phase. The testing team executes the test cases and scenarios and reports any defects or issues.
5. Defect Reporting: The defects or issues found during testing are reported to the development team. The development team then fixes the defects and sends the software back to the testing team for re-testing.
6. Test Closure: This is the final phase of the STLC. In this phase, the testing team prepares the test closure report, which includes the test summary, test metrics, and recommendations for future testing. The report is then submitted to the stakeholders for review.

Difference between Test Case and Test Scenario

A test case is a specific example used to test a particular aspect of a software system, while a test scenario is a high-level plan or strategy for testing the entire system or application.

****What is a Difference between Regression and Retesting?****

The difference between Regression and Retesting is explained below

Regression testing is the testing of an application that verifies that there is no adverse effect of the changes made due to defect fix or change request to other areas of application. regression testing involves test cases other than those that are related to defect or change as per client request, but it also tests features that directly or indirectly connected with the feature that was rectified by developer against the defect raised, such as integration tests, test case related to web components and their functions along with those involved directly with defect raised. Regression testing is usually carried out using automated scripts by selecting test cases and re executing the scripts to run those test cases.

Retesting is a testing to make sure that previously failed test cases due to defect are passed and the defect has been resolved. The test cases involved in retesting focus only on the failed test cases due to the defect or the scenarios or steps followed result in failed functionality. It is better to carry out manual testing in case of retesting to make sure that defect has actually been resolved.

****107) What is a difference between Debugging and Unit testing?****

Shreyas Answer: Debugging and unit testing are both carried out by developers, but in case of debugging, code in focus is verified by debugging tool line by line for any syntax or programming level issues such as runtime or compile-time exceptions or capturing and transferring data through lines of code.

Unit testing is a testing of a unit or program for validating functionality carried out by developer, essential to carry out to minimize bugs found at later stage like integration or system testing. These testing are known as White Box testing.

****109) How do you decide which test cases to consider for Regression Testing?****

Shreyas Answer: selection of test cases for regression testing for the build received from release, should based on following scenario priority for testing those test cases that were failed due to defect, on testing if they are passed. All integration tests, boundary value scenarios, testing functionality of components related to defect, end to end tests that involve change request or resolved test cases after release, should be considered for regression.

****110) What is a Difference between Integration Testing and System Testing?****

The difference between system testing and integration testing is listed below

System testing is a test level after unit tests and integration tests in software testing life cycle, where application as a whole is tested end to end for functional as well as non-functional requirements. It is performed after integration testing. Usability, performance, scalability and security testing are involved.

Integration testing is testing of interconnected units and their interfaces, the data flow between these units and to make sure that feature is working as per expected. it is carried out after unit testing. Various test approach like top down, bottom up or hybrid integration are carried out during testing.

****111) What are the various contents of a Test Plan?****

Test plan document should include following details

* Features to be tested
* Features not to be tested
* Approach like should we test manually, automated scripts, test techniques (white,black or grey box)
* Criteria for Failed tests / pass tests
* What condition for suspending test activities
* Test environment such as hardware, software, network etc.
* Man hours estimated while testing
* specific training required, Risks involved, any assumptions or dependencies

****114) What are SDLC phases?****

The Software Development Life Cycle consists of following phases

* Requirement Gathering and Analysis
* Design
* Coding
* Testing
* Deployment
* Support

****118) What are key challenges while testing a software application?****

Some of the challenges of testing a software applications are listed below

* Test entire application – It is virtually impossible to test and verify entire application with each and every scenario. Even one of the seven principles of software testing says, “Exhaustive Testing is Not Possible”.
* Relationship with developers: Tester’s responsibility to check for any defects in the software and how to break an application, it becomes tough to keep amicable relationship with developers
* Regression testing: In every build released from developer, tester has to check entire application for end to end functionality every time along with resolved defect and components or units that are integrated that may get affected due to change request or resolved defect.
* Testing always under time constraint: All the stack holders keep an eye on how the built has came out, testers has to quickly start testing the build with sanity test, smoke test and regression tests as send their observations as soon as possible.
* Which tests to execute first? – Based on available timeline, and complexity of modules, test lead has to distribute testing like user interface or feature testing using script automatically and regression and integration tests manually. Non functional testing using tools like load runner, API testing using Postman or REST assured. Regression tests using selenium and mobile testing using Appium.
* Understanding the requirements: There should be a walkthrough and review of software, domain knowledge, how to test features etc. is mandatory for each member of tester.
* Decision to stop the testing: Testing manager should explain testers when should testing be stopped, based on what critical bugs should get resolved, any blockers affecting feature pending by developers.

****123) What is pesticide paradox?****

Pesticide paradox is a principle wherein same set of test case if executed again and again over interval of time, then these test cases are not capable of identifying any new defects in the system. in order to overcome this effect, the set of test cases are regularly reviewed and revised to make them effective.

****128) What is the difference between Authentication and Authorization give an example.****

Authentication is verify logged in user’s genuineness, in real world you were required to show photo identity documents such as Aadhar card, Driving licence, PAN card or Passport in order to receive documents addressed to you from courier, know your account balance at your bank.

whereas authorization is verifies what module access rights the logged in user has. real life examples are You are allowed to explore feature of an application based on your trial period, once you become registered user by paying the registration fees, you can access more features of the software.

****129) Other than functionality what else you should test in Web Application?****

Shreyas Answer: In addition to functionality and feature testing, we should test browser compatibility, security tests for authentication and authorization rights, latency – how long it takes to load a web page, or request to come back as a response, load testing based on number of users logged in during peak hours, and scalability testing.

****30) How you know that testing is enough?****

The criteria and conditions are specified in exit document that guides us on when testing should be stopped. When most of blockers and critical bugs have been resolved and application has performance in acceptable limits, we can stop testing. This is however based on some risk that have to be undertaken and time limit.

****154) What are the contents of defect report?****

A defect report should inform description about the defect.

* Which version or build of software release being tested during which defect occured
* Which is the module name, how to access this module (path), what were the steps carried out
* What was the functionality affected, how severe you think the defect is
* Name of the tester who discovered the defect, name of the developer to whom defect has been assigned,
* Test case affected mapped to requirement specification document ids and number of test cases that may not be tested due to the defect.

****160) Where Exit and Entry Criteria are written in Test Plan?****

Entry criteria are written document without referring to which one cannot begin any of the SDLC phases. In order to begin SDLC phase Requirement document and complete flow diagram of the application as well as Test Plan should be ready and available. An exit criterion is the detailed document that decides moving from current phase of SDLC to the next phase. In order to exit SDLC phases test cases should be written and reviewed, test data has been identified and ready to use, scripts for automate the execution of test cases must be ready.

****163) If there is mistake in company Logo, what is Priority and Severity Justify your answer?****

Mistake in company logo on a product, does not impact its functionality or working, as well as buying or selling activities of product, the defect has low severity. But, this defect will impact greatly brand identity and impact how user will interact or may take as duplicate product, hence such defect is of high priority.

****171) Difference between Ad-hoc & Exploratory testing****

Shreyas Answer: Exploratory as well as Ad-hoc testing should be carried out by experienced testers for validating critical application that requires early iteration so to test if the application satisfies essential functionality or not.

Following is the difference between Ad-hoc and exploratory testing

|  |  |
| --- | --- |
| Adhoc Testing | Exploratory Testing |
| Ad-hoc is an informal test that can be carried out without referring to requirement specification documents for the application being tested. | Exploratory is formal tests are carried out by experienced Software testers based on their previous testing expertise thereby test cases are designed during test execution. There is no test case or scenario created during exploratory testing. |
| Ad-hoc  tests primarily tests negative scenarios | While exploratory tests, both positive as well as negative tests should be tested thoroughly, |

You can learn more about exploratory testing please visit https://www.qafox.com/importance-of-exploratory-testing/

****202) What is Defect cascading?****

Defect cascading is defects that are encountered by software testers, which further uncover another defect in the system while integration testing takes place. In order to locate defects that are primary this may further realize other defects in the application.

****246) What is Latent Defect?****

Latent Defect is a hidden defect in software still unknown to end user until particular task or operation is not performed by running the application. this type of defect are difficult to identify, pass through all testing, remains hidden even after production, occures only in certain scenario, can surface after release of the product.

****247) What is fault-masking?****

Fault masking is a technique wherein one defect prevents the detection of another. there are many features that may have defects but remain hidden until these feature are utilized. for eg. In case of login page, developer has provided option to log in with email address and mobile number, until user logs in with valid credentials, the defect as below, wherein forgot password when registered mobile number is entered OTP is not received on mobile (unhandled task by developer), remains un noticed. This is known as fault masking.

****252) What are commonly used testing types across industries?****

Answer: Unit testing, API testing, Integration testing, web services testing, security testing, automation testing are commonly used testing types across industries.

****259) What the basic components of defect report format are?****

Answer: The defect report consists of unique identification number by which the defect is registered in bug tracking software, project, product name, released version, product build, and module of a software where this defect was observed. Defect report will have summary of defect, description and steps followed that will reproduce the defect, Actual result and expected result, defect severity, ie. degree of negative impact on quality of software due to the defect – critical, major, minor or trivial, priority of defect is the urgency of fixing the defect, name of tester who has identified the defect, name of developer to whom the defect has been assigned to resolve. Current status of defect whether it is still new, assigned, open, resolved by developer, under test by tester, fixed or closed, based on bug life cycle.

****272) What do you mean by Defect Triage?****

Answer: Defect triage is process of analyzing defect priority based on factors such as severity, risk, and time required to fix /resolve the defect. During defect triage meet, stack holders discuss and decide the priority of fixing the defects.

****277) What are error guessing and error seeding?****

Answer: Description between Error Guessing and Error Seeding as details below.

* Error Guessing – It is a test case design technique in which testers have to imagine the defects that might occur and write test cases to represent them.
* Error Seeding – It is the process of adding known bugs in a program for tracking the rate of detection & removal. It also helps to estimate the number of faults remaining in the program.

****280)What is validation in software testing?****

Answer: The meaning of validation is to confirm if application or system designed is in compliance with the client’s requirements and its feature functions as desired and thereby meets goals and needs of end user. In short it is to verify the software build is right product.

****281) What is verification in software testing?****

Answer: The meaning of verification is to confirm that the designed product delivers and satisfies all the functionalities expected to fulfill as expected by end user. In short it is to test whether the product build is right (functions as expected)?

****282)What is API testing?****

Answer: API testing is a part of integration tests that verifies that the application programming interface communicates between two different systems and meets functionality, reliability, performance, and security as expected by client. Various API testing tools are SOAPUI, Postman and REST Assured.

****283)What is web service testing?****

Answer: Web service testing validates web services build on API works as expected. The functionality, reliability, performance, and security of application programming interface (API) provided to communicate in order to receive response of requests sent via HTTP protocol over an internet. Web services can be tested by asking requests over internet via HTTP protocol and receives response. Various web service testing tools are SOAPUI, Postman and JMeter, vRest

****284) What is Postman?****

Answer: Postman is open source API client that sends REST, SOAP and GraphQL requests and used to test RESTful API that have been created. It uses GET, PUT, POST and other HTTP methods in order to verify API functionality.

****309) Can you explain in short 7 principles of software testing?****

Answer: Seven principles of software Testing are as below

1. Testing shows presence of defects – testing any software will uncover defects in it
2. Early Testing is essential and will help defect in the initial stage of development.
3. Exhaustive testing is not possible – It is not possible to check each and every condition and criteria of application.
4. Testing is context Dependent – Different domains are tested differently, thus testing is purely based on the context of the domain or application.
5. Defect clustering – 80% of the problems are found in 20% of the modules.
6. Pesticide paradox – the set of test cases needs to be regularly reviewed and revised.
7. Absence of Error is fallacy as it will be impossible to imagine software without a defect.

****326) What Is Cyclomatic Complexity in software testing?****

Answer: Cyclomatic complexity for a section of code is measure of number of independent linear path taken by it. This is a standard of measurement of complexity of a program or code. for eg. there are multiple if statement in a program code, then total number of path will be equal to number of if statements.

****365) What is a difference between functional and non functional testing?****

Answer: Following are the difference between functional testing and non functional testing

|  |  |
| --- | --- |
| ****Functional Testing**** | ****Non Functional Testing**** |
| Verifies operations and actions on application under test | Verifies behavior of application when subjected to load or stress |
| These tests checks if features functions as expected by end user’s requirements | These type of tests checks applications are developed as per expectations of end user |
| Functional tests helps enhance user experience | Non functional tests improves performance of an application |
| Types of functional tests are unit testing, smoke tests, integration tests, regression tests | Types of non functional tests are performance, load, stress, and scalability tests |

****367) What 201 HTTP response codes describes?****

Answer: 201 – means HTTP Request is successful creation via either POST or PUT – HTTP methods.

****368) What is a status code for internal server error?****

Answer: HTTP status code for internal server error is 500

****369) Have you come across 404 response code?****

Answer: Yes, The meaning of 404 response code displayed on web page is – The server cannot find the requested page. i.e. There is no page found at server for the URL search.

****370) What is a difference between PUT and POST in case of REST API?****

Answer: PUT updates a specific resource or collection of resources, where as POST creates a new resource.

****403) Why do we use Joins?****

Answer: Joins are commands used to combine records from two or more tables based on related column and have one-to-many or many-to-may relationship between these tables. Based on relation between columns of multiple tables, records common from these tables can be retrieved using join commands.

****410)**** ****Have you heard of Oracle PL / SQL,****

Answer: PL/SQL is a extension to structured query language to accomplish more comprehensive solution for building mission critical applications running on oracle database. It has various functions, procedures, triggers, cursors and conditional looping for designing solutions using database records.

****413) What is the difference between XML and JSON?****

Answer: JSON (Javascript Object Notation) is a lightweight data interchange format easy to understand and generate where as XML is textual data format that support Unicode for different human languages, designed to carry data, not to display data.

****424) Have you heard of Headless Testing?****

Answer: Headless testing bypass or ignore GUI and communicates directly with application’s backend or service by sending  request and receives response back using HTTP protocol using internet. Some examples of headless technology are HTTP databases, web services and Java Messaging services.

****426) Give name of API testing tools other than SOAPUI and Postman****

Answer: Apache JMeter, Katalon Studio and Citrus few API testing tools

****428) Explain web service terminologies****

Answer: web service terminologies are as listed below

* ****XML**** – xml allow user to share structured data across various systems over internet
* ****WSDL**** – wsdl provides information how to access web services
* ****SOAP**** – soap is a protocol to exchange information or data to and from web services using xml format
* ****SOA**** – Service oriented architecture (SOA) is way in which software can be organized to respond marketplace requirement.
* ****REST**** – REST is a light weight option that is used along with HTTP protocol to develop web service.

how to confirm test coverage is good

The best way to confirm that your test coverage is good depends on the context and requirements of your specific project. Here are some general best practices to follow:

1. Define your testing objectives and requirements: Before you begin testing, you need to have a clear understanding of what you want to achieve and what types of tests are necessary. This includes understanding the scope of your project, the types of users, and the possible use cases.
2. Use a variety of testing techniques: It's important to use a variety of testing techniques to ensure that you are covering all aspects of your application. These may include unit testing, integration testing, functional testing, acceptance testing, and exploratory testing.
3. Measure your testing coverage: To confirm that your testing coverage is good, you should measure it against your objectives and requirements. You can use tools such as code coverage analysis, test case tracking, and defect tracking to measure the effectiveness of your testing efforts.
4. Continuously improve your testing process: Testing is an ongoing process, and it's important to continuously evaluate and improve your testing approach. This may involve adjusting your testing objectives and requirements, adding new tests, and refining your testing techniques.

Overall, ensuring good test coverage involves a combination of planning, execution, measurement, and continuous improvement. By following best practices and regularly evaluating your testing process, you can confirm that your test coverage is good and that your application is reliable and robust.

example for equivalence class partitioning

ChatGPT

Equivalence class partitioning is a testing technique used in software engineering to identify test cases that can be used to represent a group of similar inputs. The inputs in each group are considered to be equivalent in terms of their expected behavior, and therefore testing one input from each group is sufficient to test the entire group.

Let's consider an example of an online shopping application that requires users to enter their credit card information. To apply equivalence class partitioning, we can identify the following input parameters and their corresponding equivalence classes:

Credit card type: Visa, MasterCard, American Express

Equivalence classes:

Visa credit cards

MasterCard credit cards

American Express credit cards

Credit card number:

Equivalence classes:

Valid 16-digit credit card numbers

Invalid credit card numbers (e.g., too short or too long)

Expiration date:

Equivalence classes:

Valid expiration dates (e.g., within the next year)

Invalid expiration dates (e.g., expired dates)

CVV number:

Equivalence classes:

Valid 3-digit CVV numbers

Invalid CVV numbers (e.g., too short or too long)

Using equivalence class partitioning, we can identify the minimum number of test cases required to cover all possible scenarios. For example, we can select one test case from each equivalence class to test the following scenarios:

* Visa credit card with valid credit card number, valid expiration date, and valid CVV number.
* MasterCard credit card with an invalid credit card number, valid expiration date, and valid CVV number.
* American Express credit card with a valid credit card number, an expired expiration date, and an invalid CVV number.

By selecting one input from each equivalence class, we can ensure that all possible scenarios are covered and that the application behaves as expected in each scenario.

<https://www.geeksforgeeks.org/equivalence-partitioning-method/>

https://www.youtube.com/watch?v=7dqu5JLZHi4&list=PLL34mf651faNhSzU9ZkMds9Q7GoaZTDg\_&ab\_channel=SoftwareTestingMentor

# What is Software Testing?

# What is the importance of Software testing?

Saves Money

Ensures Security and performance

Testing helps in improving the quality of the software

What is the objective of Software Testing?

Helps in preventing the defect

It determines it is fit for customer or endusers

Builds confidence , as we have detected the defects , fixed

Why do we say testing is context dependent?

The approach to test different application is different, Eg Testing an Tutorials website is different from testing an Air traffic control system

What are some causes of software defects?

Unclear requirements

Unclear understanding about software purpose

Miscommunication within team

Mistakes by people during SDLC

Less skilled team

External factors like issue with third party applications and Integration.

What are levels of testing

Unit testing

Integration

System testing

Acceptance testing

What is Test Bed/Test Environment?

Test Env is a set of Hardware(servers) and Software, network ,simulators, tools or any other supported elements needed to conduct testing of software

What is Test case in Software testing?

A set of input values, preconditions , test steps , expected results and post conditions developed for a particular test condition to verify the software requirement

What is importance of Testcases

It provides a structured approach to testing

Reqmnt Traceability matrix.

Its helps other team members

What is Test plan?

A record of test planning service that helps a Test team to achieve

What is difference between Verification and Validation?

What is Retesting ?

Re execution of a specific test case for which defect has been fixed.

What is Regression testing

Regression testing is done to verify that there is no new defects introduced in the application/module by fixing defect or addition of new functionality or any code code changes.

What is Pesticide Paradox . How do you overcome it?

If same tests are repeated over and over again , eventually the same set of testcases will loose its ability to find any new bugs

As an when the product development is enhanced the set of testcases should be updated with new edge cases and scenarios to catch new bugs.

What are different Software development approaches

Waterfall

V-model

Agile Software development approach

How is V-model different from Waterfall model

# How will you approach testing for any Requirement or User story

1. Understand the Story
2. Define High level scenarios what you are going to test - How User is going to use the application
3. Elaborate those scenarios into TestCases

# How will you overcome the challenge of not having documentation for testing

* Try to find out any related document available
* Talk to project members and understand the application
* Document high level functionalities based on your discussion
* Apply exploratory testing technique

# Why is Regression testing in Agile Development Challenging and how will you handle it

* New features added and defect fixes every sprint
* Follow risk based approach to testing
* Try to automate regression test suite

What is Test Case Design Technique?

Test case design techniques refer to systematic methods or strategies used to create and identify test cases for software testing. These techniques help testers to ensure that the software application under test is performing as expected and meeting the requirements.

There are various test case design techniques, some of the popular ones include:

1. Equivalence partitioning: This technique divides the input domain of a software application into different partitions or sets, where each set represents a different behavior of the application.
2. Boundary Value Analysis: This technique focuses on the input values that are on the edge or boundary of the input domain. These values are more likely to cause errors or faults in the software application.
3. Decision Table Testing: This technique is used to test the logic of the software application. It involves creating a table that lists all possible combinations of inputs and their corresponding outputs.
4. State Transition Testing: This technique is used to test the behavior of the software application when it transitions from one state to another. It involves identifying the different states of the application and testing the behavior of the application when it moves from one state to another.
5. Error Guessing: This technique is used to identify test cases based on the tester's experience and intuition. It involves guessing the types of errors that are likely to occur in the software application and designing test cases to test those areas.

The selection of a test case design technique depends on various factors such as the type of software application, the complexity of the application, and the level of detail required for

What is software testing life cycle

Software Testing Life Cycle (STLC) is a process that describes the various stages involved in testing software. It is a structured approach to testing that ensures that software meets the expected quality standards and fulfills the customer requirements. The following are the various phases of the STLC:

1. Requirement Analysis: In this phase, the testing team reviews the requirements to understand the features and functionalities of the software. Based on the requirements, they create test cases and scenarios.

.

1. Test Planning: This phase involves creating a detailed test plan that outlines the testing objectives, test approach, test environments, and test schedules. The test plan also defines the roles and responsibilities of the testing team.
2. Test Design: In this phase, the testing team designs the test cases and scenarios based on the requirements. The test cases are designed to cover all possible scenarios and to ensure maximum test coverage.
3. Test Environment Setup: The testing team sets up the test environment, which includes the hardware, software, and network configurations required for testing.
4. Test Execution: The actual testing of the software is done in this phase. The testing team executes the test cases and scenarios and reports any defects or issues.
5. Defect Reporting: The defects or issues found during testing are reported to the development team. The development team then fixes the defects and sends the software back to the testing team for re-testing.
6. Test Closure: This is the final phase of the STLC. In this phase, the testing team prepares the test closure report, which includes the test summary, test metrics, and recommendations for future testing. The report is then submitted to the stakeholders for review.

Difference between Test Case and Test Scenario

A test case is a specific example used to test a particular aspect of a software system, while a test scenario is a high-level plan or strategy for testing the entire system or application.