Is there an | (40) What | (16,487 un | Scrum Ma | Contact U | (20+) Face | GoTo | Password

github.com/password_reset

Reset your password

Check your email for a link to reset your password. If it doesn't appear within a few minutes, check your spam folder.

Return to sign in

Type here to search

82°F

12:11 PM
7/21/2024

Use the arrows to **rotate the object** to face in the **direction of the hand.** (2 of 2)

Submit

42217e446b29d7f36.0701218901

Audio

Restart

Send password reset email

[GitHub | Change | (40) Wh | (16,487 | Scrum | Contact | (20+) F | GoTo | Passwo

github.com/password_reset

Reset your password

Check your email for a link to reset your password. If it doesn't appear within a few minutes, check your spam folder.

Return to sign in

Type here to search

82°F

12:14 PM
7/21/2024

System Testing

Testing the behavior of the whole software/system as defined by scope is known as system testing

Purpose of system testing is to find as many defects as possible

It is often carried out by specialist testers that form an independent test team

The system test environment should correspond to the production environment as much as possible to minimize the risk of environment-specific failures not found by previous testing cycles

System Testing

System testing should investigate both functional and non-functional requirements of the system

Black box techniques (Specification – based) should be used to system test the functional requirements

White box techniques (Structure-based techniques) ma then be used to assess the thoroughness of the testing

Test cases for System Testing

System testing includes test cases based on:

- Software Requirements specifications (SRS)
- Business processes
- End user scenarios
- High level description of system behaviors
- Interactions with the operating system
- Risks involved for the system

Acceptance Testing

- Acceptance testing starts after system testing is done and all major defect are resolved
- Acceptance testing is most often responsibility of customer but other stakeholders are also involved
- The main goal in acceptance testing is to establish confidence in the system
- Test environment for acceptance testing should be in most aspects representative of production environment (as-if production)
- Finding defects should not be the main focus of acceptance testing, its main focus should be to determine if the system is fit for purpose.

Acceptance Testing

Acceptance testing occurs at more than just a single test level.

For example:

- A Commercial off the shelf (COTS) software product may be acceptance tested when it is installed or integrated
- Acceptance testing of the usability of a component may be done during component testing
- Acceptance testing of a new functional enhancement may come before system testing

Forms of Acceptance Testing

Acceptance Testing can be classified into different forms like:

- User acceptance testing
- Operational acceptance testing
- Contract acceptance testing
- Compliance acceptance testing
- Alpha testing
- Beta testing

User Acceptance Testing

Focuses on functionality and validates fitness-for-use of the system

Business users and other stakeholders should be heavily involved

UAT Test environment should be very close to the production environment

Operational Acceptance Testing

Operational acceptance testing validates if the system meets operational requirements, it is also known as "Production Acceptance Testing"

Some of the tests to include in operational acceptance testing are:

- Testing backup/restore
- High availability & Disaster recovery
- User management
- Testing maintenance tasks
- Periodic checks of security vulnerabilities

Contract Acceptance Testing

It is performed against the contracts acceptance criteria

Acceptance criteria for contract acceptance testing is formally defined before signing the contract

app.goto.com is sharing your screen. Stop sharing Hide

Compliance Acceptance Testing

Compliance acceptance testing is also know and regulation acceptance testing.

Compliance acceptance testing is performed against the regulations (Government, legal, safety, medical) which must be adhered to.

Alpha Testing

Testing the application outside of Business Owner and Development Team is known as Alpha testing.

Alpha Testing exposes the software to people and environments outside of the project team and done at the developers site.

Alpha Testing is mostly done for COTS (Commercial Off the Shelf) software to ensure internal acceptance before moving the software for beta testing.

Alpha testing provides feedback to the developers directly from the customers / potential customers

Beta Testing or Field Testing

Beta Testing is done after Alpha Testing.

Beta Testing is done by the potential or existing users, customers and end users at the external site without developers involvement.

Beta testing is done to acquire feedback from mass market

Test Types

Test type is focused on a particular test objective. There are four main test types.

- The testing of a function to be performed by the software(Functional Testing)
- Testing non-functional quality characteristic like performance, load, stress testing (non-functional testing)
- Testing the structure or architecture of the software/system (structural testing or white-box testing)
- Testing related to any defects fixes or software changes (Re-testing and regress testing)

github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

Already have an account? Sign in →

Welcome to GitHub!
Let's begin the adventure

Enter your email*

→ anfn'paJ@YMAI

Continue

Create a password*

Email is invalid or already taken

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

app.goto.com is sharing your screen. Stop sharing Hide

Type here to search

82°F

1:00 PM
7/21/2024

Testing of function (Functional Testing or Black-box testing)

Functional testing tests the function of the system or component

Functionality of the system is usually described in documents such as software requirements specifications, use cases or a functional specification

Functional testing considers the specified behaviour of the system and is often referred as black box testing

Functional Testing may be performed at all levels of testing

Perspectives of doing Functional Testing

Requirements based

- Uses functional requirements specification as basis for designing tests
- Prioritization done based on risk criteria mentioned in requirements document

Business-process based

- Uses knowledge of the business process as a basis for designing tests
- Business processes describe the scenarios involved in day to day business use of system
- Use cases are very useful basis for designing test cases

Perspectives of doing Functional Testing

Requirements based

- Uses functional requirements specification as basis for designing tests
- Prioritization done based on risk criteria mentioned in requirements document

Business-process based

- Uses knowledge of the business process as a basis for designing tests
- Business processes describe the scenarios involved in day to day business use of system
- Use cases are very useful basis for designing test cases

Testing of software product characteristics (Non-functional testing)

1. Non-functional testing tests the non functional attributes like reliability, efficiency, maintainability, usability etc.
2. Non functional testing is done with an aim to find how well or how fast an operation is performed by software.
3. It may be performed at all test levels
4. It tests how well the system works
5. Some of the non functional testing types are:
   - *Performance testing*
   - *Load testing*
   - *Stress testing*
   - *Usability testing*
   - *Maintainability testing*
   - *Reliability testing*
   - *Portability testing*

Types of Functional Testing

- Unit testing
- Integration testing
- System testing
- Acceptance testing

- Smoke testing
- Sanity testing
- Regression testing
- Re-testing
- UI testing

- GUI testing
- White-box testing
- Black-box testing

- Alpha testing
- Beta testing

Testing of software structure/ architecture (Structural Testing or White box testing)

Structural testing tests the structure or system architecture of software,

Structural testing is concerned about what is happening inside the box and that is why structure testing is also referred as white box testing

Structural testing is done with the help of code coverage tools which asses the percentage of executable elements that have been exercised or covered

If coverage is not 100%, then additional test cases are need to written to cover those missed items

Testing related to changes (Confirmation and regression testing)

Testing related to changes tests the software whenever there are changes done in the software.

For Example:
- When you fix the defects/bugs
- When you implement new functionality in software

Testing related to changes has two types of testing
- Confirmation testing (re-testing)
- Regression testing

Testing related to changes (Confirmation and regression testing)

Testing related to changes tests the software whenever there are changes done in the software.

For Example:
- When you fix the defects/bugs
- When you implement new functionality in software

Testing related to changes has two types of testing
- Confirmation testing (re-testing)
- Regression testing

Confirmation testing or Re-testing

Confirmation testing or Re-testing comes in picture when the test fails and a defect is logged against that test

Once the defect is fixed and a new build is obtained with fixed defect re-testing is done to make sure that defect has been fixed and test passes now

While re-testing it is important to follow exactly same steps and use same input, data and environment.

Regression Testing

If a defect has been fixed it might have affected other areas of code. Regression testing is done to find the "unexpected side effects" of defect fixes

Purpose of regression testing is to verify that any modifications in software have not caused unintended side effects in software.

Regression tests are executed whenever the software changes either due to defect fix or addition of new functionality

All the test cases in regression test suites are executed every time the new software version is available so regression test suite is an ideal candidate for automation

Impact analysis and regression testing

Usually maintenance testing consists of two parts

- Testing any changes
- Regression testing to show that any changes done in software does not affect the rest of the software

Impact analysis is the main activity in maintenance testing

- Impact analysis is done together with stakeholders to decide what parts software might be unintentionally affected and need careful regression
- Risk analysis helps to decide where to focus regression testing

Impact analysis and regression testing

Usually maintenance testing consists of two parts

- Testing any changes
- Regression testing to show that any changes done in software does not affect the rest of the software

Impact analysis is the main activity in maintenance testing

- Impact analysis is done together with stakeholders to decide what parts software might be unintentionally affected and need careful regression
- Risk analysis helps to decide where to focus regression testing

What is static testing?

Testing of a component or system at specification or implementation level without executing the code is known as static testing (Walkthrough, Technical Reviews, Inspection)

During static testing software products are examined manually or with some tools but there is no execution done

Static testing helps to verify the software deliverables for which dynamic testing techniques cannot be applied (like, design document, SRS, test plan etc.)

More about Static Testing

Static testing technique provides a powerful way to improve quality of software

Objective of static testing technique is to improve the software quality by assisting developers to recognize issues early in SDLC

Static testing is not a replacement for dynamic testing

All organizations should implement static testing techniques to improve software quality

More about Static Testing

Static testing technique provides a powerful way to improve quality of software

Objective of static testing technique is to improve the software quality by assisting developers to recognize issues early in SDLC

Static testing is not a replacement for dynamic testing

All organizations should implement static testing techniques to improve software quality

Static Testing vs Dynamic Testing

Static Testing

- Static testing examines software deliverable without execution
- Static testing finds the cause of failures
- Static testing technique is know as verification

Dynamic Testing

- Dynamic testing verifies the software by executing it
- Dynamic testing finds the software failures
- Dynamic testing technique is known as validation

Next chapter

Test Development Process & Test Design Techniques

Introduction to Test Development Process

It is important to know what you are trying to test, inputs and expected outcome before you actually start test execution.

Let's learn about

1. Test Conditions (Documented in test design specification)
   - An item or event of a component or system that could be verified by one or more test cases, e.g., a function, transaction, feature, quality attribute, or structural element.
2. Test Procedures or Test Scenario (Documented in test procedure document)
   - A document specifying a sequence of actions for the execution of a test. Also known as test script or manual test script.
3. Test Cases (Documented in test case specification)
   - A set of input values, execution preconditions, expected results and e
     postconditions, developed for a particular objective or test condition,
     exercise a particular program path or to verify compliance with a spe
     requirement.

Formality of test documentation

Testing can be performed with varied level of formality

It could be very informal with minimal documentation or very formal with extensive test documentation

Right level of formality depends on context of testing

- A commercial safety-critical application needs very formal approach as compared to a family website which will be used by only few people.

Level of formality is also influenced by the organization, its culture, maturity of developm
testing process etc.

In this session we will cover the formal approach of test documentation

Test Development Process

The test development process consists of 3 main phases

- Test analysis
- Test design
- Test implementation

Test Analysis - Identifying Test Conditions

The process of looking at test basis to derive test information is known as Test analysis.

Test basis is anything like system requirement, SRS, design documents, code, business process etc.

In test analysis we find out what could be tested or test conditions, test condition is something which can be tested

The chosen test conditions depend on the test strategy or detailed test approach. For example, they might be based on risk, models of the system, likely failures, compliance requirements, expert advice

The test conditions are then converted into detailed test cases, test design techniques are u figure out good set of tests

Test conditions should be able to be linked back to the source documentation to identify traceability and Requirements Coverage

Test Design – Specifying Test Cases

During Test design phase we create test cases

During test design you need to be very specific, you need exact and detailed input conditions, pre conditions and test outcomes

Your test cases should include

- Input values & Test Data
- Pre conditions
- Expected results

Expected result should be defined prior to execution

Test cases are prioritized to ensure that high priority test cases are executed first

Test cases need to be detailed

Test Implementation – Specifying test procedures or scripts

In test implementation the test cases are grouped into test suites and ordered into proper sequence

- Functional Test Suite
- Regression Test suite

During test implementation you also specify the test procedure – Test procedure is a document which specifies the steps to be taken in running a set of tests, it is also referred as test script.

Test procedures are then formed into test execution schedule or super script which specifies which procedures are to be run when and by whom.

Test design techniques categories

There are three categories of Test design techniques

- Specification-based (Black-box) testing technique
- Structure-based (White-box) testing technique
- Experience-based testing technique

Specification-based (Black-box) testing technique

Specification based testing technique is also known as black box testing technique

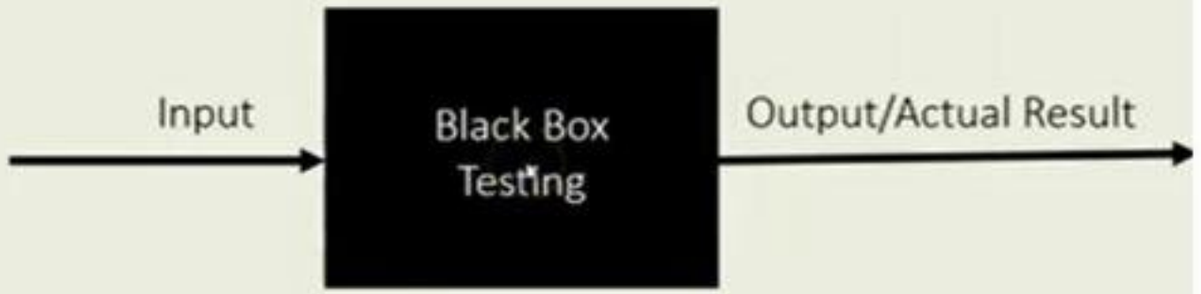
This is dynamic testing technique

This is input-output driven testing technique as you are not concerned about internal software structure.

In this technique tester is concerned about what the software does not how it does

You test both functional and non functional characteristics of software using this technique

Black Box Testing Technique



If Output = Expected result, then your test passes

Structure-based (White-box) Testing Technique

Structure based testing technique is also know as white box or glass box testing technique

It is also a dynamic testing technique

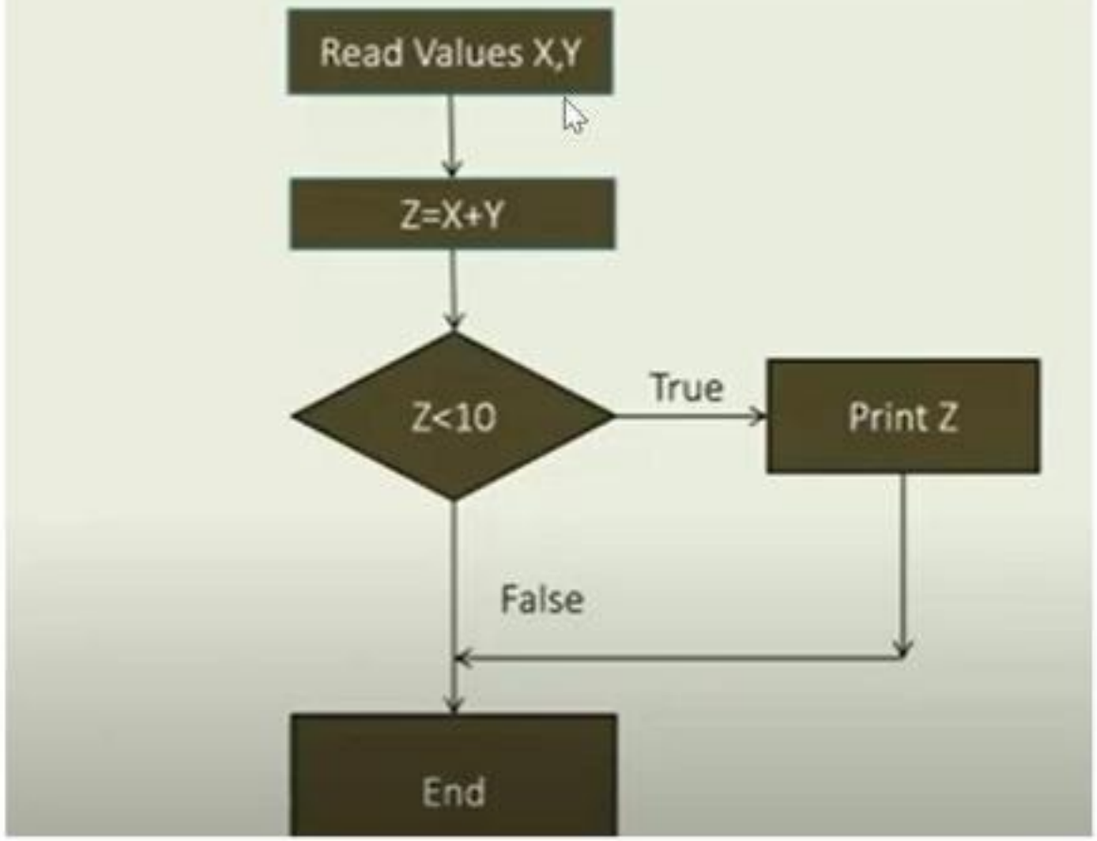
White Box testing uses internal software structure to design test cases.

It tests how the decisions and conditions are exercised, unit test cases are developed to exercise loops, decisions

It tests paths within a unit and also flow between units during integration of units.

Requires knowledge of internal code structure and good programm skills.

White-box Testing Technique



Experience-based testing technique

In experience based testing technique knowledge of testers, developers, users and other stakeholders is the used to design test conditions and test cases

Experiences of both technical people and business people is considered

Because of the prior experience people know where things can go wrong in the software.