International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 07 | July - 2021 ISSN: 2582-3930

SOFTWARE TESTING LIFE CYCLE

Pragati

ABSTRACT

Testing is a major activity in software development process to find the defect in the software. In software development we have software development life cycle(SDLC) model is a process which includes various phases that are followed to develop the software application and in process of testing we have software testing life cycle(STLC) is a process to evaluating the software application. The testing phases of SDLC are expanded further in software testing life cycle. STLC defines the series of activities which should be carried out systematically to help the evaluation of software application. This paper presents the different phases, activities and deliverables of STLC[2].

Keywords: Software testing, Software testing strategies, Testing tools, Test Cases (TC), Software Testing Life Cycle.

I. INTRODUCTION

Today`s era is computer era, everything is fast going. Easy, accurate and fast retrieval of information is necessary because time is most precious thing at present so we need of testing before & after launching the software application in the industry. Testing is the critical element of software quality assurance and represents the ultimate review of specification, design, and coding. Software can be tested either manually or by automation tools. Process of software testing is the main activity of evaluating and executing software to find out the errors. It is the process where the system requirements are analyzed and evaluated manually and test implementation by using automation tools. This paper emphasizes the main activities, which are requirement analysis, Test Planning, Test Design, Test Implementation, Test Execution and Test Closure[3].

II.SOFTWARE TESTING TECHNIQUES

1.Black box Testing- Functional and system testing,Stress testing,Performance testing,Usability testing,Acceptance testing,Beta testing,Ad hoc testing,Regression testing, Inter system testing, Volume testing, Parallal testing, Boundary value.

2.White box Testing-Unit testing, Error handling testing, Desk checking, Code walk through, Code reviews and inspection , Code coverage testing, Statement/ Path/ Function/ Condition testing, complexity testing/Cyclomatic complexity, Mutation testing .

3.Grey box Testing-Integration Testing, Regression Testing[2].

III. SOFTWARE TESTING LIFE CYCLE

Software testing life cycle defines the series of activities which should be carried out systematically to help the evaluation of software application.

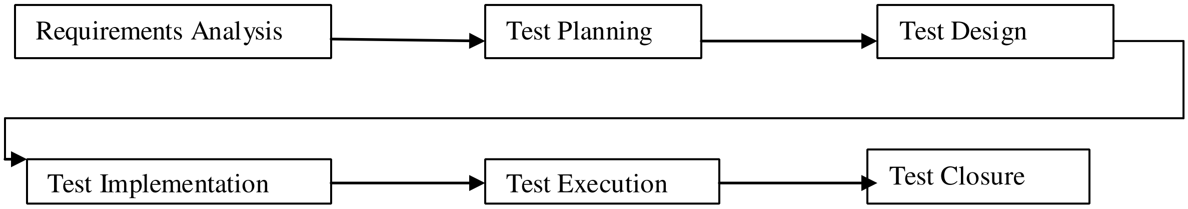

Fig. 1: Block diagram of STLC

3.1 Requirement Analysis

During this phase, testing team gathers application knowledge & details about priorities and focus area. Testing team studies the requirements from a testing point of view to identify the testable requirements. The QA team may interact with various stakeholders (Client, Business Analyst, Technical Leads, System Architects etc) to understand the requirements in detail. Requirements could be either Functional (defining what the software must do) or Non Functional (defining system performance /security availability). Automation feasibility for the given testing project is also done in this stage. Activities of this phase is to identify types of tests to be performed, gather details about testing priorities and focus, prepare Requirement Traceability Matrix (RTM) ,identify test environment details where testing is supposed to be carried out, Automation feasibility analysis .Deliverables of this phase are list of Testable requirements[1].

3.2 Test Planning

Test plan derives the complete process of the Software Application. Activities of this phase are preparation of test plan/strategy document for various types of testing, test tool selection, test effort estimation, resource planning and determining roles and responsibilities, training requirement.

Deliverables of this phase are test plan /strategy document, effort estimation document.

3.2.1 Test Plan contains following details

Test plan identifiers it gives unique identifier with version number. Can also be called as document number.

Introduction it gives overview of software application and purpose of this document.

Test scope it defines the extent of testing and defines the clear boundaries for testing process.

Test objective it defines Testing Goal and provides guidelines for Testing Process.

Assumption and Dependencies It provides documents prerequisites, which if not met could have a negative impact on schedule, Quality of Budget of testing project .Examples Readiness of Test Environment setup etc. **Risk Analysis** It provides document possible risks during testing activities and their estimated impact on the test effort and schedule. Examples of risks Builds not available on time for testing, test environment related issues, defect not getting on time.

Strategy

It provides what methods of testing to be used Black box or white box or combination.

It provides levels of testing in scope unit, integration etc

It provides number of test execution cycles to be planned

It provides non-functional tests to be conducted performance, Security.

It provides process for Test Environment setup, Test Data creation.

Features to be tested

This provides listing of functionality mentioned in Software requirement specifications which are to be tested & features not be tested and features which are not in current scope of testing.

Role and Responsibilities

It provides testing team structure and their responsibilities.

Test Schedule

It provides major test activities, their sequence and their inter dependencies, time estimation for each activity and test resource plan i.e. planning of people, tools, infrastructure and facilities.

Entry and Exit criteria

It describes when to start testing and when to stop testing .For example testing can be started when build ,test environment and test data is ready and for exit criteria when all test cases are executed and all the defects are closed.

Suspension criteria: It denotes when to pause or suspend Testing activity.

For example when major functional defects are found blocking the further testing process

Resumption Criteria

When all the suspension issues are resolved continue with testing process.

Test environment

It provides hardware /software configuration, Environment requirements for each stage and type of testing, Procedures for Configuration management and release / version control, location for individual test artifacts, defect tracking mechanism.

Communication plan

It provides what to communicate in terms of reports, status, documents for information etc It provides whom to communicate in terms of stakeholders, when, how in terms of emails, formal and informal meetings, working sessions.

It provides which communication tools to be used such as electronic bulletin boards, emails, intranet sites etc.

It provides Escalation procedures viz communication protocol, hierarchy to be followed.

It provides miscellaneous items such as project contact lists, meeting audiences.

Testing tools to be used

Test management tool, configuration management tool, defect tracking tool, special tests testing tools, test script development tool, test utilities i.e record playback or test data creation etc.

Document Control

It includes reviewer of test plan, approval of test plan and test plan distribution list

3.3 Test Design

This phase of STLC covers the process of getting inputs to start test design , creation of test scenarios , constructing test cases, creating test data , using techniques like equivalence class partitioning, boundary value analysis and requirement traceability matrix. This phase aims at documenting „how" to do testing. This phase involves creation, verification and rework of test cases & test scripts. Test data , is identified/created and is reviewed and then reworked as well. Activities of this phase include to create test cases, create test data, automation scripts, review and baseline test cases and scripts

Deliverables of this phase are Test cases/scripts and Test data[5].

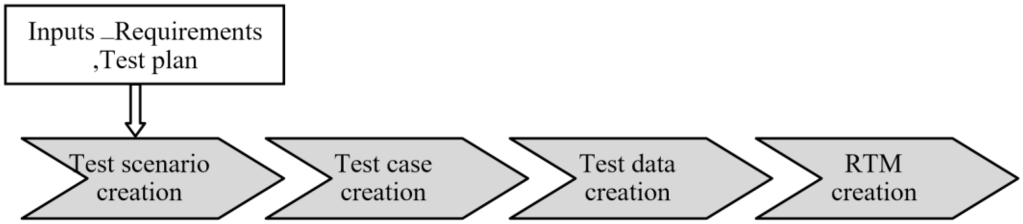

Fig. 2 Test design

Once test planning is completed, test plan is available to the testing team members. It gives clarity about the testing scope and entire the testing process that needs to be followed .The first key activity of the testing team is to design a test case . Test case design consists of creation of test scenarios, test cases, test data and requirement traceability matrix.

Inputs for test design activity

Stage of application	Inputs for test design or artifacts to study
Application under development	Requirement documents-SRS, use cases, wireframes, prototype
Application in use • Under enhancement (phase wise releases) • Under maintenance (handling change requests)	• Application knowledge through walkthrough, demo • Previous testing artifacts like test scenarios, test cases, defects for app • Change request documents

3.3.1 Test Scenario

Test Scenario can be defined as a top view of functionalities under test or what is needed to test at high level. Test scenario identification ensures coverage of all features of the application in testing. If test scenarios are not identified , it may result in a particular functionality not getting tested in detail or not getting tested at all. Test scenario further help in developing End to End or combination scenarios which actually simulate the complete user interaction with application. Though test scenarios are important to create, they need not be formal deliverable always.

Identification of Test Scenarios

From use case- Use case describes how user (Actor) will interact with (or Use) the application functionalities (processes). Here each process is the Test Scenario. For example, in user registration use case, registration can be considered as one process.

By functionality breakdown – Functionality breakdown includes identifying various features or services provided by the application based on the requirement specification documents. Number of test scenarios will be equal to the number of broader level functionalities of the application. Using state transition diagram technique- In transition based applications, the status of an entity can change based on the user transactions. The process that changes the state of an entity can be identified as one test scenario. This is called state transition technique. For example, cancel order process can change state of order from in process to cancelled. Hence cancel order is identified as one test scenario[6].

Example on Test scenario

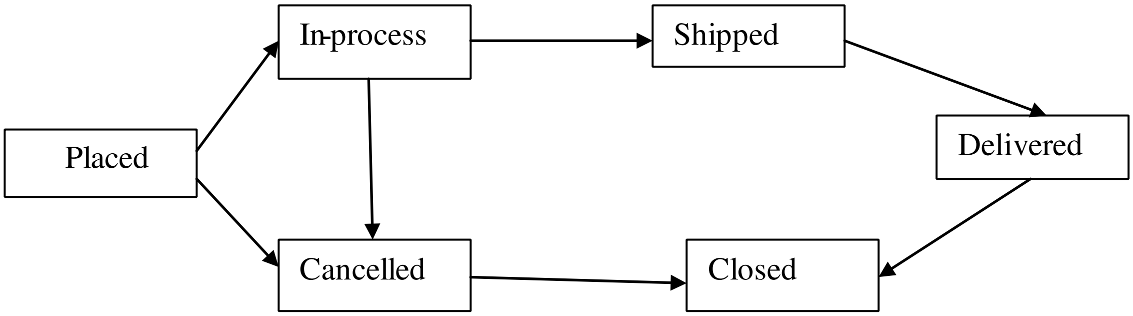

Fig. 3 Test scenario

This figure 3 depicts the possible state transitions for an order.

→ Placed → →in-process shipped delivered → closed

→ → Placed cancelled closed

→ Placed → in-process cancelled → closed

3.3.2 Test case

Test case provides a detailed procedure that helps to test a particular aspect or feature of an application in detail .It specifies how to test the particular functionality and describes steps to be performed with input data and output expectations based on the user requirements. Test cases are used while test execution to check the actual behavior of the application **. Process of Test case Creation**

Test case creation includes identification of test conditions and then documenting the detailed process to test this test condition .Test conditions can be identified in the following two ways: From Test Scenario breakdown- For every scenario ,identify various paths – a normal or happy path ,alternate path and error flow . Each path will lead to unique test condition.

From Use Case or functionality- Check normal flow, alternative flow and error flow documented for use case or functionality .Each of them leads to unique test condition[5].

Test Case Review

Once test cases are created by the tester they need to undergo a review before they are actually used in validation or dynamic testing. Reviewers can be peer i.e. test team members, Superior i.e. Test lead or application expert. Sometimes, customers can also review the test cases.

Why review Test Cases?

Test cases are the most important deliverable of Test design phase and if needed , are shared with customers.

To check correctness and completeness of documented Test cases.

Test Case Storage

Test cases should be storage in Test Management Tool . Quality center , Testlink are Test Management Tool.

3.3.3. Test Data

Test case includes test data to be used during test case execution. One test case can be executed multiple times with different sets of test data to check the application behavior under different conditions.

3.3.4. Requirements Traceability Matrix (RTM)

Requirement Traceability Matrix is a tool to ensure that the project`s scope , requirements , and deliverables are in line. **Use of RTM**

- To check that all requirements are implemented
- Helps to do impact analysis when any particular requirement changes –Tester can find out the affected Test scenarios and test cases using RTM.

3.4 Test Implementation

This phase implements the test cases by using Automation tools by writing test scripts.

Deliverables of this is Test scripts

3.5 Test Execution

On completion of Test design phase , test cases which document „how to do testing are ready. The next logical phase is "Test Execution". In this phase test cases are executed and application`s behaviour is observed to evaluate whether it is as expected .Test execution is defined as processing of a test case by the software under test to produce an outcome. During this phase testing team will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed. Activities are

- Document test results, and log defects for failed cases
- Map defects to test cases in RTM
- Retest the defect fixes
- Track the defects to closure Deliverables

- Completed RTM with execution status
- Test cases updated with results
- Defect reports

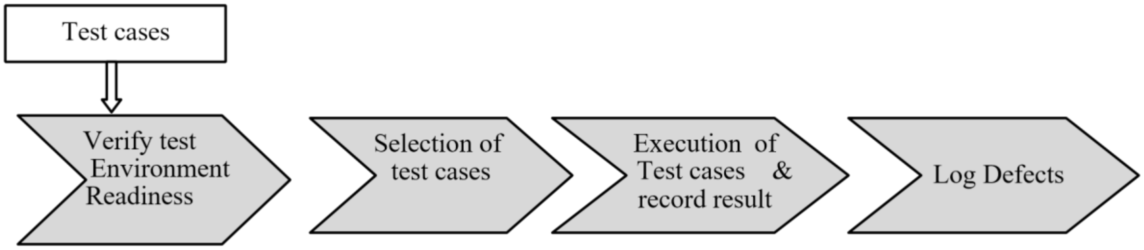
Test cases
Verify test Environment Readiness
Selection of test cases
Execution of Test cases & record result
Log Defects

Fig. 4 Test execution

When to start Test Execution?

Test execution entry criterion defines when the test execution should start. Test Environment readiness, availability of test cases and test data, completion of prior levels of testing and availability of build for testing are the minimal requirements for test execution[5].

Test Execution Cycle

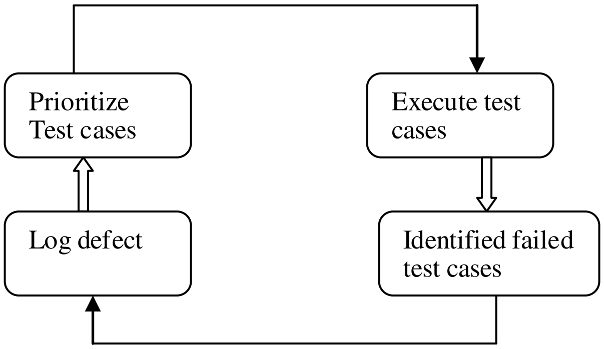

Fig. 5 Test execution cycle

Test Execution Steps

1. Prioritize test cases: Test case prioritization is needed when a tester has insufficient time in hand. Test cases which test core functionality of the application or critical test cases are executed on high priority and rest of the test cases follow them based on the time availability.
2. Execute identified test cases.
3. Perform the steps mentioned in the test case and observe the application`s response .
4. Compare Actual result and Expected Result and accordingly update the status column as

"pass" if both match or "fail ". Other status value can "on hold" if the test case is dependent on completion of failed test case or" not executed" if the functionality is not in scope of current cycle.

5. Log defects for failed test cases.

When to Stop Testing?

Testing should be stopped when it meets the exit criteria put in the test plan. Test manager , client and respective stakeholders will consider all above and similar other factors to decide when to stop.

3.6 Test Closure

Test closure is last phase of STLC. In this data is collected from completed test activities to consolidate experience, leanings, facts and figures . Test closure activities include the following major tasks:

- Check which planned deliverables have been delivered Create Test Summary Report.
- Handover to test ware to the maintenance team.

Test summary Report provides details of the testing outcome and advice on the release readiness of the application. The purpose is to summarize the results of the designated testing activities and to provide evaluations based on these result. Testing team will meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle. The idea is to remove the process bottlenecks for future test cycles and share best practices for any similar projects in future[8].

3.7 Mapping of Software Test Process ans Testing Types

There are four major phases in software test life cycle which are earlier explained in this paper. Each phase require some sort of testing to be performed. In test analysis and test preparation phase only verification of requirement documents and other test documents is done. Verification involves all types of reviews, inspection and walkthroughs. It is mainly done before validation. Once Test Preparation is done and all artifacts are reviewed and base lined, then Test execution begins, where in actual validation is performed. In this phase all types of testing is performed which are shown in Figure 6 below at a high level[9].

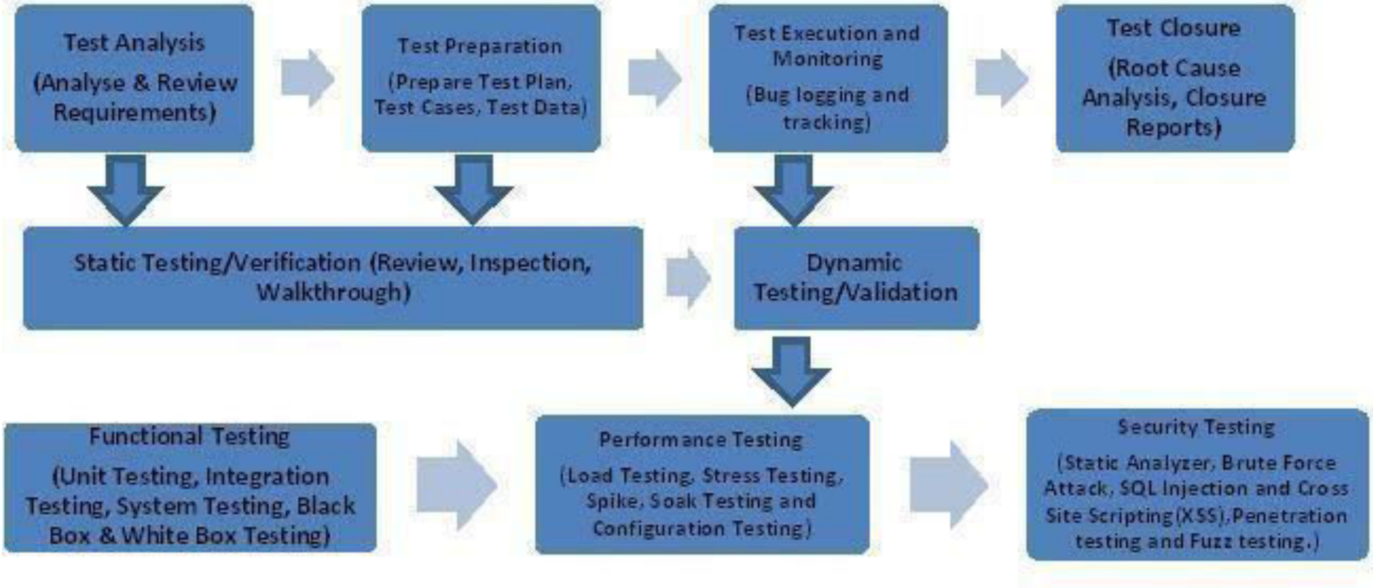
Test Analysis
(Analyse & Review Requirements)
Test Preparation
(Prepare Test Plan, Test Cases, Test Data)
Test Execution and Monitoring
(Bug logging and tracking)
Test Closure
(Root Cause Analysis, Closure Reports)
Static Testing/Verification (Review, Inspection, Walkthrough)
Dynamic Testing/Validation
Functional Testing
(Unit Testing, Integration Testing, System Testing, Black Box & White Box Testing)
Performance Testing
(Load Testing, Stress Testing, Spike, Soak Testing and Configuration Testing)
Security Testing
(Static Analyzer, Brute Force Attack, SQL Injection and Cross Site Scripting(XSS),Penetration testing and Fuzz testing.)

Fig. 6 Mapping of Software Test Phases and Testing Types

3.7 CONCLUSION

This paper was to study on different phases of Software Testing Life Cycle. After studied various phases of software life cycle it is found that there are main 4 phases in testing life cycle that could be categorized as Analysis, Planning and Preparation, Execution and Closure. In this paper we studied the inputs , activities & deliverables of different phases and also seen the overview of mapping of software test phases with different Testing types[8].

REFERENCES

[1] Pressman, R.S. 1997. Software Engineering: A practitioner Approach.4th Edition. Tata McGraw Hill.

[2] Sommerville, I. 1998. Software Engineering. 5th edition. Addison-Wesley.

[3] Myers, G.J. The Art of Software Testing. New York: John Wiley and Sons.

[4] McGraw.Chess,B. Seven pernicious kingdom.2005.A Taxonomy of Software Security Errors.WISST*Workshop on Software Security Assurance Tools, Techniques and metrices.*

[5] G.McGraw.2004.Software SecurityTesting .IEEE Security and Privacy.2,2(Sept/Oct 2004),(80-83).

[6] D,Verndon. G.McGraw.2004.Risk Analysis in software Design. IEEE Security and Privacy.2,4.32-37 (July/August 2004).

[7] Sarma, M. D,Kundu.Mall, R. 2007. Automatic Test Case Generation from UML Sequence Diagram. International Conference on Advance Computing and Communication.Doi:10.1109/ADCOM:2007.68.

[8] Bertolino, A. Basanieri, F. 2000. A Practical approach to UML-based derivation of integration tests. In Proceeding of the Fourth International Software Quality Week Europe and International Internet Quality Week Europe(QWE), Brussels, Belgium.

[9] Boghdady, P,N.Badr, ,L.Hashem, M.Tolba, M,.F. 2011.

A Proposed Test Case Generation Technique based on Activity Diagrams.IJET-IJENS: 11.No:3.(37-57)

[10] Treharne, H. Draper, J. Schneider, S. 2006. Test Case