

A Review Paper on Software Testing

Amit M. Kale¹, Vivek V. Bandal², Komal Chaudhari³

^{1,2}Bachelor Student, Dept. of Electrical Engineering

³Professor, Dept. of Computer Engineering

Abstract – Software Development Life Cycle is an important concept used in Software Engineering to describe a procedure of software development i.e., Requirement Gathering Analysis, Design, Coding, Testing, Deployment and Maintenance phase. These phases are actual implement as per customer requirements. The aim of Software Development Life Cycle is produce to high quality of software that meets the customer expectations, reaches completion within times.

Software Testing is a process used to help identify the correctness, completeness and quality of developed computer software. Testing is very important activity in Software Development Life Cycle.

The aim of this paper is discuss the Software Development Life Cycle and Software Testing Life Cycle process for better quality assurance of product.

Key Words: Software Development Life Cycle (SDLC)¹, Software Testing Life Cycle (STLC)², Methods of Testing³, Levels of Tetsing⁴, Requirement Gathering Phase (RGA)⁵ etc.

1. INTRODUCTION

SDLC and STLC is one of the wide topics in Software Engineering. SDLC is a method by which the software can be developed in a systematic manner which wills increases the probability of completing the project before deadline and maintaining the quality of standard. The ISO/IEC 12207 standard represents the process that establishes a life cycle for a development of software.

Testing is a process to find out the difference between required condition and existing condition. We find out the defects at the time of testing and fix these defects the quality of software system is increases. STLC refers to a testing process which has specific steps to be executed. Different organization have different phases in STLC; however basics remains are same i.e., Test Planning and Control, Test Analysis and Design, Test Implementation and Execution, Evaluate Exit Criteria and Reporting and last phase is Test Closure Activity. Each phase has different goals and different output.

2. Software Development Life Cycle

SDLC is a process used by software industry to design, develop and test high quality of software. SDLC follows the

standard ISO/IEC 12207 for the proper development of product. The SDLC aims to produce a high quality of software that meets customer expectations and customer satisfaction.

2.1. Phases of Software Development Life Cycle

The work product of SDLC is Build. The work product means output of that phase. In SDLC involves different phases i.e., RGA, Design, Coding, Testing, Deployment, Maintenance.

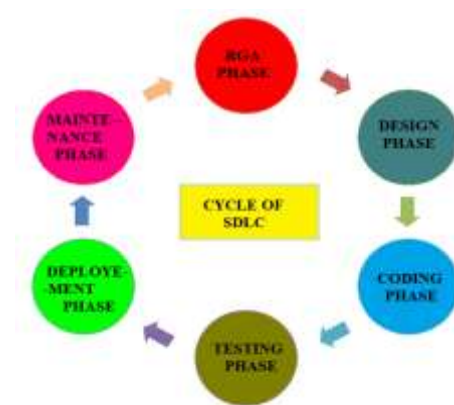


Fig -1: Cycle of SDLC

2.1.1. Requirement Gathering Analysis Phase

RGA is the first and basic phase of SDLC. This phase is most essential for developing the product. We discuss in details of RGA phase is given below.

Requirement Phase: - Requirement means actually need of people or “What” customers want? The characteristics of requirements are complete and correct, unambiguous, verifiable, traceable, modifiable, atomic and understandable. The requirements are classified in two parts i.e., Functional Requirement and Non-Functional Requirement

Functional Requirement: - Functional requirement means what my software actual do? Or what actual work of my software

EX: - Pen (Functional Requirement of Pen is writing on paper).

Functional Requirement are two types which is Explicit Requirement and another is Implicit Requirement. In Explicit Requirement, requirements are demanded by customer or requirement mentioned by customer. In Implicit

Requirement, requirements are not demanded by customer but should be in product.

Non-Functional Requirement: - Non-Functional Requirement means How, Look and Feel of software. Means how work, how perform and design of such application.

EX: - Pen (Non-Functional Requirement of Pen is Smoothly writing on paper or design of pen).

Gathering Requirement Phase: - Gathering or collect the information related to product. This information is gathered by Business Analyst i.e., BA. BA collects the information related to need of customer. Then he create the document known as Business Requirement Specification i.e., BRS which is non-technical or user language.

Analysis Phase: - After requirement gathering, requirements are analyzed by Senior BA and Technical Team for the verification of such requirement and possibilities of developing requirement. In this phase involves both Functional and Non-Functional Technical Team. The Senior BA and Technical Team create a document known as Software Requirement Specification i.e., SRS. The output of RGA phase is BRS and SRS.

2.1.2. Design Phase

SRS Document is the input for the designing of product. Design helps in specifying hardware and also helps in defining overall product structures. We can distinguish two types of design phases i.e., HLD and LLD.

HLD: - HLD stands for High Level Design. It is also known as Architectural Design or System Level Design. The architectural design provides an overview of an entire system to identifying the main components that would be develops for the product and their interface.

LLD: - LLD stands for Low Level Design. It is also known as Structural Design, Program Level Design or Details of Design. This LLD Design process can be used for designing data structures and source code.

The work product of Design phase is HLD and LLD.

2.1.3. Coding Phase

Once the Design is complete then coding is the one of the important phase in SDLC. The input for coding phase is LLD. The goal of coding phase is convert the design of system into a programming language. The actual development is start in this phase. The coding is done by Developer. The work product of Coding phase is Source Code which is in Document File and Build which is in Execution File.

2.1.4. Testing Phase

In testing phase required as Build and SRS document. The basic function of this phase is to find bugs, defects and errors in developed product. The goal of this phase is releases the

quality product or customers satisfied by her requirements. In testing phase testers execute the test case by using SRS file. The work product of Testing phase is Test Plan, Test Case, Test Scenario, Test Script, Requirement Traceability Matrices and Defect Report.

2.1.5 Deployment Phase

Deployment is done after the system has been tested and accepted by the customer. In this phase involved experienced Installation Team.

The work product of this Deployment phase is User Guide and User Manual.

2.1.6 Maintenance Phase

In this phase care is taken developed product. Once when the customers starts using the developed product then the actual problems comes starts and needs to be solved from time to time. If any bugs are occurred then bugs are fixed in this phase.

Updating any feature, Fixing bugs or Restrictions on any features are done in this phase.

2.2. Software Development Life Cycle Models

There are various models defined and designed which are followed during development process. Those models are also known as "Software Development Process Models". Waterfall Model, Multi-Waterfall Model, Spiral Model, Agile Model, Verification and Validation Model are the most important and popular SDLC Model followed in industry.

2.2.1. Waterfall Model

The Waterfall Model is the Classical model in Software Engineering. This model is one of the oldest models. The Waterfall Model has six phases i.e., RGA, Design, Coding, Testing, Deployment and Maintenance phase. In Waterfall model, typically the outcome of one phase acts as the input of the second phase sequentially. This model does not work smoothly if there are some issues left at the previous phase. The next phase is started only after defined set of goals are achieved in previous phase. In this model future adjustment in project are not possible.

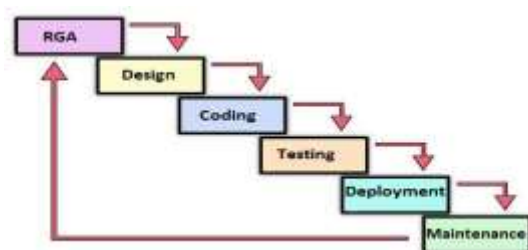


Fig -1: Waterfall Model

2.2.2. Multi-Waterfall Model

This model generally used in large project. Multi-Waterfall Model is known as Incremental Model which leads the software development process in iterations. After each iteration, the management process team can do work on risk management and prepare for the next iterations. In this model parallel development are done. This models required good planning and design.

2.2.3. Spiral Model

The Spiral model has four phases i.e., RGA, Risk Analysis, Engineering (Developer and Tester), Evaluation Customers (Customers Feedback). The Spiral Model begins with designs of goals and ends with customers' satisfaction. Software is produced in early phase. It is very costly method and for development of models needs expertise person.

RGA Phase: - In this phase requirements are gathered. Requirements like BRS and SRS

Risk Analysis Phase: - If any risk is found during the risk analysis phase the alternate solutions are suggested and implemented.

Engineering Phase: - In this phase development and Testing is done.

Evaluation Customers: - In this phase customer evaluate the output of project and give their feedback.

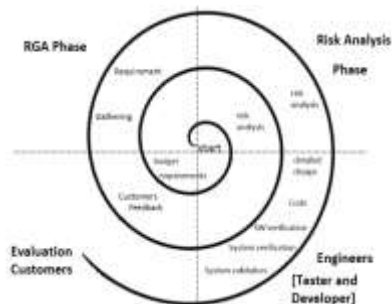


Fig -3: Spiral Model

2.2.4. Agile Model

Agile Model is known as Incremental Iterative Model. Agile Model is used for changes requirements as per clients. Software is developed in incremental or rapid cycles. The phases in spiral models are RGA, Risk Analysis, Design, Coding and Testing. For starting next iteration required 2 to 3 months. Regular adaption to changing circumstances is done.

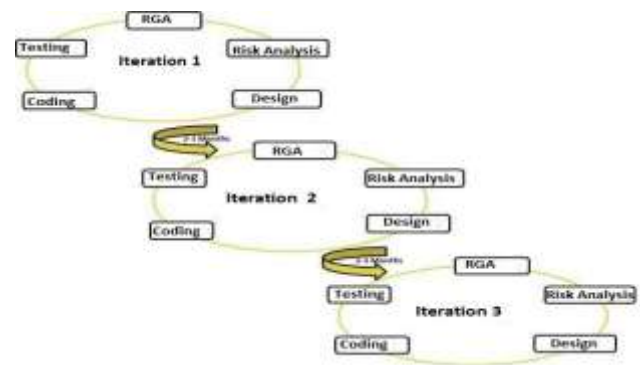


Fig -4: Agile Model

2.2.5. Verification and Validation Model

Verification and Validation Model is known as V-V Model or W Model. The V-V Model is a SDLC Model where execution of process happens in a sequential manner in V shape. This is a highly disciplined model and next phase starts only after completion of the previous phase. It explains the relationship between each phase and the associated testing phase. This model does not work smoothly if there are some issues left at the previous phase. The next phase is started only after defined set of goals are achieved in previous phase. In this model future adjustment in project are not possible.

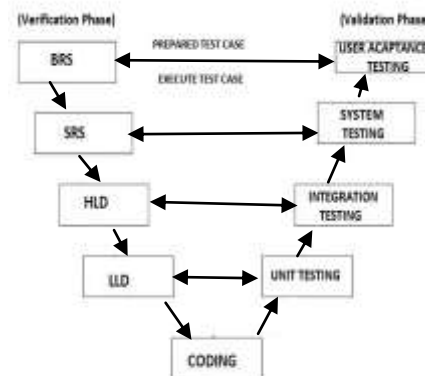


Fig -5: Verification and Validation Model

3. Software Testing

Software Testing is necessary because we all make mistakes. Some mistakes are unimportant, but some of them are inexpensive or dangerous. We need to check everything and anything because sometime we produce may be a wrong in software. That's reason software testing is necessary for improving all bugs or mistakes or to improve the quality of software.

3.1. Who does Testing?

It depends on the process and testing can be done by everyone. In most cases, the following professionals are involved in test a system within their respective capacities:

- Software Tester
- Test Leader
- Test Manager
- End User
- Developer

3.2. When Start Testing?

Testing can be started from the requirement gathering phase and continued till to the deployment of software.

3.3. When Stop Testing?

It is difficult to determine when to stop testing. Testing is a never ending process. When the testing deadline are reach, completion of test case execution or by management decision stop the Testing.

3.4. Manual Testing

The part of software testing that requires human input, analysis or evaluation.

3.5. Bug

A bug is an error or fault in computer program cause it to produce an incorrect result.

3.6. Error

Error is the mistakes or gives the unexpected result or working is not properly of the system.

3.7. Fault

Fault is an error caused by ignorance or inattention by the human.

3.8. Failure

When suddenly stops the working of project or system is known as Failure.

3.9. Defect

Difference between expected result and actual result is known as Defect.

3.10. Software Testing Life Cycle

STLC refers to a testing process which has specific steps to be executed. The different phases of STLC are Test Planning and Control, Test Analysis and Design, Test Implementation and Execution, Evaluate Exit Criteria and Reporting and Test Closure Activity.

3.10.1. Test Planning and Control

Test Plan: - It consist complete planning of testing process. Test leader or test Manager is involved in the planning. In this planning we schedule the test analysis. It includes test tasks, test items.

Test Control: - In Test Control, controlling all tasks or activity of test plan. Test control measures and analyse the result of testing. It control on the document progress and exit criteria.

3.10.2. Test Analysis and Design

Test Analysis: - In this test analysis, analyze the requirements are testable or not. In this test analysis SRS are involved (i.e., Functional and Non-Functional Requirement)
Test Design: - The purpose of test design technique is to identify test condition and test scenarios through which effective and efficient test cases can be written

3.10.3. Test Implementation and Execution

Test Implementation: - In this test implementation phase test case are implement for testing the product verifying that the test environment has been set up correctly.

Test Execute: - In Test Execution Phase test cases are executed or build the product. In this execution process we compare actual result and expected result.

3.10.4. Evaluate Exit Criteria and Reporting

Evaluate Exit Criteria: - Evaluating exit criteria define the stop or end of the testing process. Exit criteria are done when all the test cases are implementing and execute successfully. Evaluate exit criteria depends on parameter like 100% , 80% Passed - 20% Fail Criteria (depends on Organization) and deadline reach.

Reporting: - Test reporting activity is one of most imp activity in the organization. In this activity give report to the senior person or Team Leader on daily, Weekly or Monthly basis.

3.10.5. Test Closure Activity

When all tests are done successfully or meet the expectations or requirements then testing can be closed. It is a Knowledge Transfer Process. It is used like a reference book used for another test project.

3.11. Methods of Testing

There are two types of methods of testing i.e., Black Box Testing and White Box Testing

3.11.1. Black Box Testing

Black Box Testing is known as Closed Box Testing, Functional Box Testing. Black Box Testing is done after completion of project. This test is done by Tester. A Technique of Black Box Testing is Equivalence Class Partitioning Technique, Boundary Value Analysis Technique and Error Guessing Technique.

3.11.2. White Box Testing

White Box Testing is known as Closed Box Testing, Structural Testing or Glass Testing. Test is used for check the internal structure i.e., coding/programming. This test is done by developer.

3.12. Techniques of Black Box Testing

There are three types of Black Box Testing Technique are Equivalence Class Partitioning Technique, Boundary Value Analysis Method and Error Guessing Method.

3.12.1. Equivalence Class Partitioning Technique

Equivalent Class Portioning is known as ECP. This techniques is very common and mostly testers are used this method. At the name suggest Equivalent Class Portioning is divide or partition of a set of test conditions. In these techniques we take only one Valid Value and two Invalid Values.

3.12.2. Boundary Value Analysis Technique

Boundary Value Analysis known as BVA. In Boundary Value Analysis, we test the boundaries between equivalence partitions.

3.12.3. Error Guessing Method

Error Guessing is an Experienced based techniques. This test is done by Senior Tester, Experienced Person of Team Leader. For the Error Guessing Techniques required Skilled and Experienced Person.

3.13. Levels of Testing

In Levels of Testing we checks the end product after it is created. It helps in finding the presence of defects. Levels of testing also known as Techniques of Validation. The different levels of testing are Unit Testing, Integration Testing, System Testing and User Acceptance Testing.

3.13.1. Unit Testing

Unit Testing are also known as Component Testing or Module Testing. It is the testing process of Individual Module

of a system. The purpose is to validate that each unit of the software performs as designed. This test is done by Developer. Approaches for Unit Testing are Black Box Testing and White Box Testing.

3.13.2. Integrated Testing

In Integration Testing test interfaces between components and interaction with different parts of a system. This test is done by Developer. There are three types of approaches of Integration Testing are Top Down Approach, Bottom Up Approach, Big Bang Approach

3.13.2.1. Top Down Approach

In this Top Down Testing, testing is conducted from main module to sub module. Main module is known as calling module and sub module is known as called module. We create sub module as a Dummy Module or Stub.

3.13.2.2. Bottom up Approach

Bottom Up Approach Testing starts from the bottom or lower unit of application and gradually moves on. We create main module as a dummy module or Driver.

3.13.2.3 Big Bang Approach

In this Big Bang Module we have to integrate all modules at the same time. But this is not possible in real that's reason this Big Bang Approach is not used in Industry.

3.13.3. System Testing

In this System Testing whole application or software should be check. This test is done by Tester. System Testing includes Functional Testing and Non-Functional Testing. System Testing mainly focus on external interfaces, security, performance and usability.

3.13.4. User Acceptance Testing

This test is done by Customer. There are two types of User Acceptance Testing i.e., Alpha Testing and Beta Testing.

3.13.4.1. Alpha Testing

Alpha Testing is known as Internal Testing. Alpha testing is final testing before software or application is released. In this testing involves Developer, Tester and Customer. At the Development Site Customer are come for the testing product.

3.13.4.2. Beta Testing

Beta Testing is known as External Testing. In these testing customers uses their own data or live data to test the

software or application. Customers test the software or application at customer's site.

4. CONCLUSION

This paper is focused on SDLC Cycles and STLC Cycles. In this paper we studied on different SDLC cycle, SDLC Model and STLC Cycle, Methods of Testing, Levels of Testing. Software Engineer is an Engineer Branch associated with development of software produced using well defined methods and procedure. After review I would like to conclude that software testing is a basic activity of SDLC. Testing only shows the presence of errors and bug. Debugging helps us to improve the quality of software.

REFERENCES

- [1] Involve testing throughout the SDLC available at <http://www.silverpath.com>
- [2] Sanjana Taya, Shaveta Gupta, "Comparative Analysis of Software Development Life Cycle Models"
- [3] A. M. Davis, H. Bersoff, E. R. Comer, "A Strategy for Comparing Alternative Software Development Life Cycle Models", Journal IEEE Transactions on Software Engineering, Vol. 14, Issue 10, 1988
- [4] Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan, "Software Testing Techniques and Strategies", International Journal of Engineering Research and Applications, Vol. 2 Issue 3, May-Jun 2012, pp.980-986
- [5] SoftwareTestingOverview, available: http://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm
- [6] E. F. Miller, "Introduction to Software Testing Technology", Software Testing & Validation Techniques, IEEE, 1981, pp. 4-16