

Testing in the Software Development Life Cycle and its Importance

Noortaj

Department of Computing and Technology
Abasyn University Peshawar, Pakistan
Email: damonkhan99@gmail.com

Abstract : *Software development life cycle is a structure followed by the development team within the software organization imposed on the development of software product. The important role of testing in the software development life cycle is to improve the performance, reliability, and quality of the software. I believe testing should be thoroughly planned and conducted throughout the Software development life cycle for best results. Hence, software testing is facing different challenges. In this article, i have explained different phases of the software development life cycle, different testing techniques and its importance.*

Keywords: Software Development Life Cycle Testing, Testing in the Software Development Process, Software Testing, Importance of testing.

1. INTRODUCTION

Testing plays an important role in the software development life-cycle to recognize the defects in the development process and resolve those defects. But before the explanation of the Software development life cycle, I would like to explain the phases of the software development process.

Phase 1 Requirement collection and Analysis: For the collection of requirements a discussion is conducted to analyze the requirements and to state how the problem should be solved. The main work of the requirement collection is to determine the anticipated issues and it helps companies to finalize the necessary timeline in order to complete the work of that system in time.

Phase 2 Requirement Specification: Requirement specification describes the behavior of the required system and also helps to avoid inconsistency and redundancy. Single UNIX Specification and multiboot specification are examples of the functional specification.

Phase 3 Feasibility study: The process of the feasibility study is conducted with the help of the 'SRS' document means 'Software Requirement specification' document. This document includes everything which should be designed and developed during the software development life cycle.

Phase 4 Software Construction: In the software construction the development of user's documentation for the system is done through the combination of coding, integration, verification and debugging. Software construction main fundamentals are:

- Minimizing complexity
- Anticipating change
- Construction for verification

Phase 5 Software Designing: The designing phase normally involves problem-solving and planning up solutions for the software. Two types of design documents developed in this phase, High-level designing and low-level designing.

Some main concepts of design are:

- Abstraction
- Data structure
- Information hiding
- Structural partitioning
- Modularity

Phase no 6 Software Testing and Evaluation: It includes software verification, testing, validation of the software just developed. The most important purpose of testing are to find software defects so that it can be corrected. Software testing is done to verify that the product works according to the customer requirement. In Software evaluation, we evaluate Resource Management, Technical Requirements and Product Quality, including Reliability.

Phase 7 Debugging: Debugging is the process of locating and resolving the defects in software and hardware through the debugger. A debugger is a debugging tool that helps in locating and identifying programming errors at different phases of software development.

Phase 8 Software Deployments and Maintenance:

Deployment starts after no errors or bugs left.

Maintenance includes error corrections, optimization and enhancement of capabilities.

Main activities of maintenance are:-

- Corrective Maintenance
- Perfective Maintenance
- Preventive Maintenance
- Adaptive Maintenance
- Risk-based Maintenance
- Condition-based

2. TESTING IN THE SOFTWARE DEVELOPMENT LIFE CYCLE

Software testing means that there should be no disparity in the software development process. Each software development has passed through a set of different phases. So early starting of testing means we can find and remove small bugs before they become bigger problems in the end and also provides the chance to review requirements for the attributes of the quality. There are three testing phases in SDLC:

- **Test Analysis:** In test analysis, the tester tries to understand the project.
- **Test Design:** The tester design the test cases based on user requirements.
- **Test Execution:** The tester executes the test cases and finds out defects if any.

Different types of testing involve throughout SDLC are:

1. **Acceptance Testing:** Formal testing to determine if the requirements of the user are met and business processes conducted to determine the acceptability of the system.
2. **Ad-Hoc Testing:** Ad-hoc testing is performed without documentation or planning and its main objective is to find errors that are not discovered by other types of testing.
3. **Alpha and Beta Testing:** Alpha testing is done by testers at development sites after the acceptance testing. Beta testing is carried out by potential users in the real environment.
4. **White Box Testing:** White box testing tests the internal structures and workings of applications, as opposed to its functionality.
5. **Black Box Testing:** Black box testing is the testing technique where the functionalities of the software are being tested.
6. **Integration Testing:** In integration testing the individual tested modules are combined as one and the interface between them is tested. It is done after the unit testing.
7. **Automated Testing:** Automated testing is done by using automation tools to write and execute test cases.
8. **Grey Box Testing:** It is a testing technique that uses a combination of white box testing and black box testing. The aim of this

testing is to find out the defects if there are any due to improper structure or improper usage of the application.

9. **Regression Testing:** Regression testing is a testing activity rerunning the functional and non-functional tests from existing test cases to know that software changes have no unintended side-effects.
10. **Stress Testing:** A software testing activity that determines the stability and robustness of software by testing beyond normal operational capacity limits.
11. **Grey Box Testing:** It is a testing technique that uses a combination of white box testing and black box testing. The aim of this testing is to find out the defects if there are any due to improper structure or improper usage of the application.
12. **Regression Testing:** Regression testing is a testing activity rerunning the functional and non-functional tests from existing test cases to know that software changes have no unintended side-effects.
13. **Stress Testing:** A software testing activity that determines the stability and robustness of software by testing beyond normal operational capacity limits.
14. **Security Testing:** Security testing tests the ability of the software to reveal flaws and to prevent unauthorized access to the resources of the software.
15. **UAT (User Acceptance testing):** User acceptance testing is performed by the end users of the product. It is done in the last phase of testing.
16. **Performance Testing:** The purpose of software performance testing is to assess a user's experience in real scenarios on our developed application. In the software field, it is performed to determine how a system performs in terms of stability, responsiveness, and speed in every type of condition.

3. The importance of testing in the SDLC

Some major importance of testing in the SDLC are the following.

3.1 Identification of Defects/Bugs:

The identification of bugs and defects is done in the testing phase. These defects can be at any level (unit level/system level). With the testing of all phases, you get to keep away from almost all kinds of defects that can affect your application.

3.2 Product Quality Improvement:

Testing helps to know what is the actual result and the expected result. Surely, it helps in the improvement of your product quality. After proper testing, your software is free of errors and develop a desirable software for the users.

3.3 Reputation of company and satisfaction of stakeholders:

Testing helps you to know the product's quality. The stakeholders also get good information through a testing phase about the quality of your company products and will be satisfied.

3.4 To defeat your Competitors:

Best testing tools and techniques help you and your product to grow up in the market and to defeat your competitors. After testing in all phases, the software you develop will be more reliable, safe and secure.

3.5 Technical Importance:

In software development life cycle testing is important for the technical aspects, as you know that it has to come out with the best technically correct application.

3.6 Verification and validation:

One of the main purposes of testing in SDLC is verification and validation. Testing will help you that your software meets with the required specifications through verification and validation.

3.7 Improved Quality:

Properly tested software you will be more confident that your software will be of great quality, secure, reliable and safe for all the users.

3.8 Prove Operability and Usability:

Another important purpose of testing is to prove that the application is both operable and usable. Usability testing is a testing technique to observe a user's interaction with the product on all aspects, and where the users are facing problems those problems are recorded and analyzed.

3.9 Keep away any danger:

It is important that you go under all phases of testing, in order to keep everyone away from any type of danger.

3.10 Reliability Estimation:

Testing determines your software's reliability. If you have tested your software through all levels like unit testing, regression testing, and all other types, then surely it is the most reliable software.

3.11 Prevent Defect Recurring:

If the defects are detected early, then they can be prevented from recurring to the next development phase. Early detection and removing errors lead to saving a lot of development costs.

3.12 Economic importance:

Well-tested software will have a pretty good impact on your revenue. And that is because anyone would want to go with a reliable, secure, good quality and trusted applications in the market.

Software Quality Factors

Product operation (Functionality)	Product Revision	Product transition (Adaptability)
Reliability	Testability	Reusability
Correctness	Integrity	Flexibility
Usability	Documentation	Maintainability
Efficiency	Structure	portability

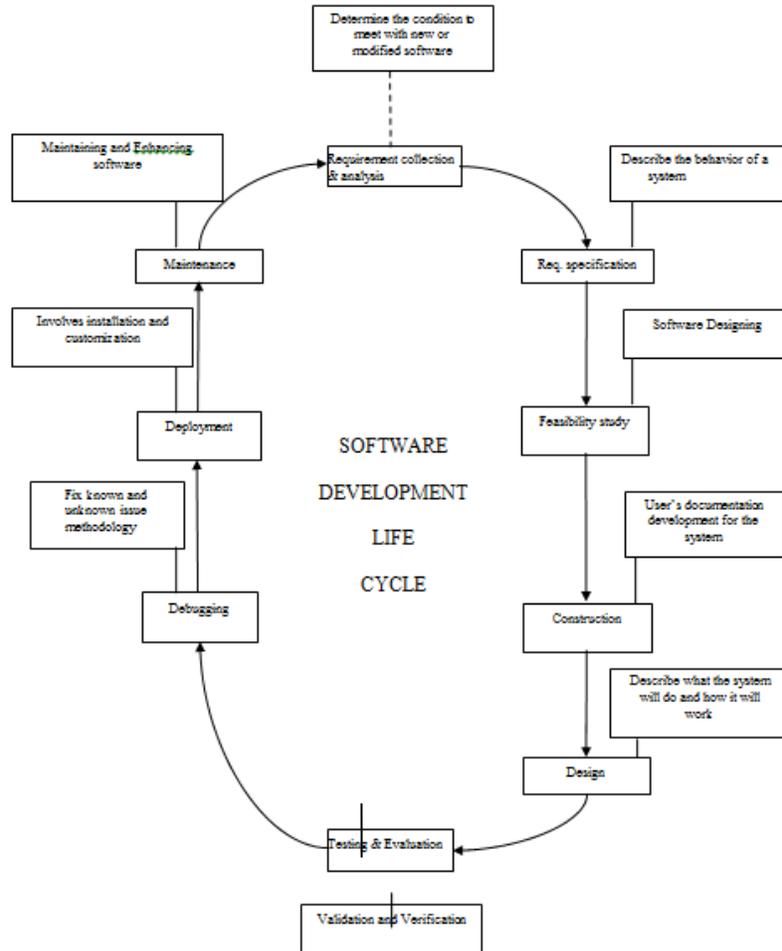


Fig. 1 Represent all phases of the software development life cycle

4. CONCLUSION

From this paper, it is cleared that testing is very important for delivering a quality product to the users and to the success of the overall effort. From this paper, it is concluded that testing should be thoroughly planned and conducted throughout the SDLC for improving reliability, performance, and quality. In this paper, I have explained all phases of the software development life cycle and the importance of testing in every phase.

REFERENCES:

- [1] T.Rajani Devi, "Importance of testing in SDLC", International Journal of Scientific & Engineering Research Volume 3, Issue 5, May-2012.
- [2] Rajendra Kumar, "Software Testing techniques and Strategies," International Journal of Engineering Applied Sciences and Technology, 2016 Vol. 2, Issue 2, ISSN No. 2455-2143, Pages 113-117.
- [3] J. A. Whittaker, "What is Software Testing? And Why Is It So Hard?" IEEE Software, January 2000, pp. 70-79.
- [4] Prof. Swati Dubey, Prof. Shubhangi Takwane, Prof. Dipti Dighe, "Study on various phases of Software testing," 7th International conference on recent trends in engineering, Science and Management, 1st-2nd April 2017, Pune.
- [5] Mohd. Ehmer Khan, Farneena Khan, "Importance of testing in SDLC," IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 2, No 2, March 2014.
- [6] Langer, Arthur M. "Guide to Software Development-Designing and Managing the Life Cycle." January 2012.
- [7] Paul Jorgensen, "software testing: A craftsman approach" 4th edition (1995).

- [8] Tanu Jindal, "Importance of testing in SDLC," International Journal of Engineering and Applied Computer Science (IJEACS), Volume: 01, Issue: 02, December 2016.
- [9] A. Abran, "Software Metrics and Software Metrology," Wiley & IEEE Computer Society Press, Hoboken, 2010.
- [10] M. Thillaikarasi, K. Seetharaman, "Regression testing in developer environment for absence of code coverage," Journal of software engineering and applications, 2014, 7, 617-625.
- [11] Kamala Ramasubramani Jayakumar, Alain Abran, "A survey of software test estimation Techniques," Journal of software engineering and applications, 2013, 6, 47-52.
- [12] Thakur Monika, Sanjay, "Review on structural software testing coverage approaches," International journal of advance research, ideas and innovations in technologies, Volume 3, Issue 3.