International Journal of Ingenious Research, Invention and Development
ISSN (Online): 2583 648X

Volume 1 | Issue 2 | February 2023
Scientific Journal Impact Factor (SJIF 2023) = 3.647
DOI: 10.5281/zenodo.7821125

Debugging Error with The Help of Software Testing Methods

Dr Sachin Agrawal[1], Prabhat Chauhan[2]
[1]*Assistant Professor, CSE, College of Engineering and Technology, Akola, India*
[2]*Graduation Student (BE Final Year), CSE, College of Engineering and Technology, Akola, India*

Abstract: Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. Testing is a very important step in software development to ensure that the software does properly, what it is supposed to do. While the development of the software there arise certain problems that leads to error, defects, and complications in the software testing to make the software run smoothly. In this report, I tried to over whole software testing, why software testing is required, hoe test cased are made, types of testing, levels of testing, method of testing, why software testing, levels of testing, method of testing & further how important software testing is. In short, we can say that testing is the last step in the software development life cycle.

Keywords: Testing Methodologies, Software Testing Life Cycle, Testing Frameworks, Automation Testing, Test Driven Development, Test optimization, Quality Metrics, etc.

I. INTRODUCTION

Software is process carried out to find whether the actual results meet the expected results. Software Testing is the process to check whether the software is defect-free or not. Software testing is an activity to check whether the actual results match system is Defect free. Testing is the last step in the software life cycle. Time pressure is well known and increasing because too many defects are found late and must be required. This Seminar shows you how to cope with this situation, Early test planning and the use of reviews achieve a high degree of preventing and defect removal. Testing can never completely identify all the defects within software. Instead, it Furnished a criticism or comparison against oracle-principles or mechanisms by which someone might Recognize a problem. These oracles may include (but are not limited to) specification, contracts comparable product. Past versions of the same product, Inferences about intended or expected purpose, user or customer expectations, Relevant standards, applicable laws, or other criteria. Every software product has a target audience. For example, the audience for video Game software is completely different from banking software.

II. SOFTWARE DEVELOPMENT LIFE CYCLE

SDLC is a process used by software industry to design quality software that meets the expectations of the Real Users. It is cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure. There are seven basic steps for the software development life cycle/ i.e., Ideation/project initiation, Requirement gathering analysis, Design, Implementation/development, Testing, Development, Maintenance. All the steps/phases of the SDLC have their own specific work and outputs.

International Journal of Ingenious Research, Invention and Development
ISSN (Online): 2583 648X

Volume 1 | Issue 2 | February 2023
Scientific Journal Impact Factor (SJIF 2023) = 3.647
DOI: 10.5281/zenodo.7821125

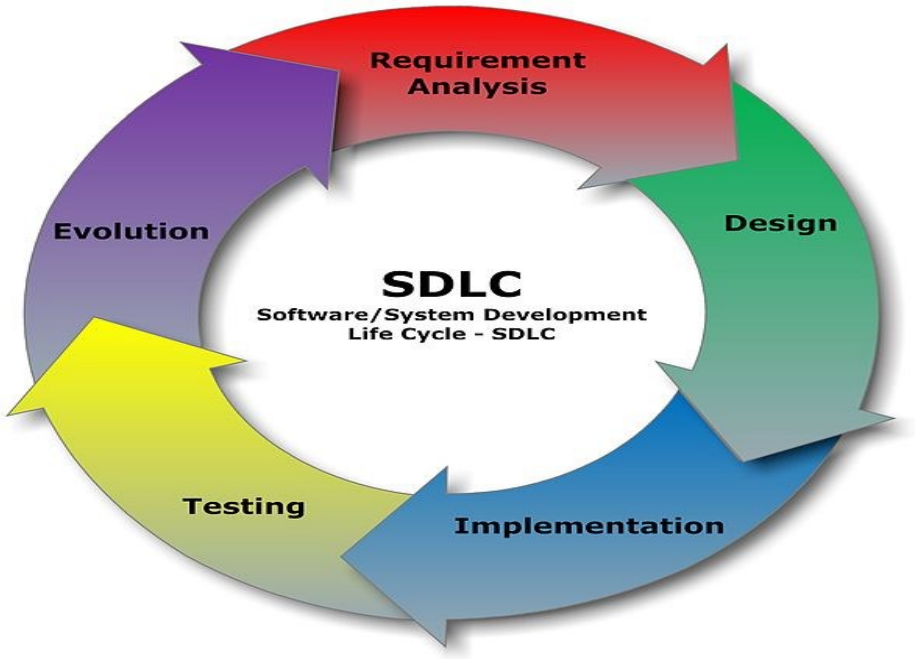

Figure 1: Software Development Life Cycle

2.1) Testing Methods

After development of software, there are two paths using that testing can be done: named as white box testing and black box testing

1. White-box Testing: It is also known as clear box testing, glass box testing, transparent box testing and structural testing. In white-box testing, an internal perspective of the system is used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. White-box testing can be applied at the unit, integration, and system levels of the software testing process. This is also involving in security testing.
2. Black Box Testing: Black-Box testing is a method of software testing that examines that functionality of an application without peering into its internal structured or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system, and acceptance. It is sometimes referred to as specification-based testing.

2.2) Levels of Testing

There are four levels in software testing: Unit Testing, Integration Testing, System Testing, and Acceptance Testing.

1. Unit Testing: In this type of testing, errors are detected individually from every component or unit by individually testing the components or units of software to ensure that if they are fit for use by the developers. It is the smallest testable part of the software
2. Integration Testing: In this testing, two or more modules which are unit tested are integrated to test i.e., technique interacting components and are then verified if these integrated modules work as per the expectation or not and interface errors are also detected.

International Journal of Ingenious Research, Invention and Development
ISSN (Online): 2583 648X

Volume 1 | Issue 2 | February 2023
Scientific Journal Impact Factor (SJIF 2023) = 3.647
DOI: 10.5281/zenodo.7821125

III. LITERATURE REVIEW

Software Testing is the most critical part of the Software Development Lifecycle, as it is something upon which the final delivery of the product is dependent. It is time consuming and an intensive process, therefore, enhanced techniques and innovative methodologies are requisite. This makes Automated Testing and other various Test Metrics implementation before and during the testing process.

The platform over which the software development and testing reside continues to evolve and remains exceedingly eminent. However, something so crucial and critical like Testing comes often quite late in the process of Software Development. There should be a maximum interaction between specification writers and Testers for better understanding and early review, which may fix ambiguity problems and consequently result in saving the cost of later fixing of the software.

Testers after being clear about the specifications and requirements should hand over developers a certain light weight test model, so they make sure the primary specification are met before handling the project for official testing. Use of simulation tools can immensely help the testers in creating the similar environment in which the products destined to run, certain exception testing and methods for the exception handling can be best determined. While testing the product in the similar testing environment for which the products meant for, and that can be easily done by integrating the simulation within the Testing process. Hence, the future work in relevance with the testing process will be much more technology dependent harnessing the simulation and automated testing model-based approach, not only expediting the testing life cycle but also providing optimum bug prevention and efficient quality assurance.

IV. CONCLUSION

Testing is a critically important verification method that takes up a very large portion of a project's resources, including schedule, budget, staffing, and facilities. Unlike the many constructive activities of systems engineering, testing is relatively unique because it is inherently destructive. Its primary purpose is to force the system or its components to fail so that the defects that caused the failure can be uncovered and then fixed. In addition to defect detection, testing is also performed to provide sufficient objective evidence to justify confidence in the system's quality, fitness foe purpose, and readiness for being accepted and placed into operation.

REFERENCES

[1] McConnell, Steve (2004). Code complete (2nd ed.). Microsoft Press.Pp.29. ISBN 0-7356-1967-0.
[2] Principle 2, Section 1.3, certified Tester Foundation Level Syllabus, International Software Testing Qualifications Board
[3] Trab, Eucheuma (1999). "Verification/validation/Certification". In Koopman. Topics In Dependable Embedded Systems. USA: Carnegie Mellon University.. Retrieved 2008-01-13.
[4] See D. Galperin and W.C. Hetzel
[5] Introduction, Code Coverage Analysis, Steve Cornett.
[6] Laycock, G. T. (1993) (PostScript). The Theory and Practice of Specification Based Software Testing Dept of Computer Science, Sheffield University, UK. Retrieved 2008-02-13.
[7] Patton, Ron, Software Testing.
[8] Binder, Rovert V. (1999). Testing Object-Oriented Systems: Objects, Patterns, and Tools. Addison-Wesley Professional, P. 45. ISBN 0-2-1-80938-9.
[9] Bezier, Boris (1999). Software Testing Techniques (Second ed). New York: Van Nostrand Reinhold Pp. 21,430. ISBN 0-442-20672-0.
[10] IEEE (1999). IEEE Standard Computer Dictionary: A Complication of IEEE Standard Computer Glossaries. New York: IEEE.ISBN 1559370793.