A Roadmap For Defect Elimination With Enhanced Software Development

Mr. M.Selvakumar

Assistant Professor, Department of Computer Applications, Ayya Nadar Janaki Ammal College (Autonomous), Sivakasi, Tamil Nadu. *Contact:* 8220206200, *Email :* mselvakumarmca@gmail.com

Abstract: Software testing is an essential part of every project or product development. Nowadays every software project or products are delivered with lot of defects. These defects are not possible to solve in the maintenance stage or deployment stage of the product or project. At the same time no one cannot add any requirement in the deployment stage also. So project or product must be tested every stage in the Software Development Life Cycle (SDLC). In the SDLC models, each stage contains lot of intermediate and sublevels. So, testing must be done in all the levels of the Software Development Life Cycle. It is not an easy process for testing in all the levels of SDLC. But, in this paper gives some better suggestion and solution for testing in the all the levels of SDLC. These solution or suggestion surely minimizes the number of defects to deliver better product or project development.

Keywords: Testing, defects, SDLC

1. BACKGROUND

Software Engineering

Software engineering is defined as the application of systematic disciplined and quantifiable approach to develop an effective and efficient software product or project.

Software testing

It is an essential and important internal part of software engineering. The verification and validation of a software project or product is called software testing. It is an execution of software project or product with actual test data. Testing is focused on quality issues, poor quality software not acceptable by the user. Software failure leads to catastrophic losses.

Software project/Product

Software project is an individual requirements and product is a universal requirements. Both are built by time, budget and quality constraints.

Software development

When an engineering approach is employed to the software development it implies,

- ✓ Development process understanding
- ✓ Planned projects
- ✓ Defined life cycle models
- ✓ Quality measurements standards
- ✓ Evaluation of the projects
- ✓ Components are reused
- ✓ Verification and validation
- \checkmark Engineer training, etc.

Software process

Generally software development process is started in terms of phases, procedures, steps, etc. It is a set of activities and associated results which produces a software product or project. These activities are carried out by software engineers. Fundamental process activities are,

- ✓ Software specification
- ✓ Software development
- ✓ Software validation
- ✓ Software evaluation

Defect

It is a variation between actual software requirement specification and final executed build (i.e. exe files). The product varies from software specification and customer expectation. Defects are categorized by,

Wrong - the specification have been implemented incorrectly.

Missing - A specified or wanted requirement is not in the built product.

Extra - A requirement incorporated into the product or project is not specified.

Defects can be carried from,

- ✓ Poor educational background of software engineer
- ✓ Poor communication of software engineer to group members
- ✓ Software engineer omit something
- ✓ Misdirection of activities, etc.

Test Case

To avoid maximum number of defects, test cases must be developed. It helps to find and analyze, software product or project is working properly or not. Test case creation, tester set hypothesis. All test cases based on hypothesis, it helps to

- ✓ Prove hypothesis
- ✓ Disprove the hypothesis

2. IMPLEMENTATION

a. Overview - Problem description

Testing is done in testing phase of Software Development Life Cycle (SDLC). Defects or misdirection of requirement occurs any time in the Software development Life Cycle (SDLC). So, testing must be done in all the

levels of the Software Development Life Cycle. It is not an easy process for testing in all the levels of SDLC. But, in this paper gives some better suggestion and solution for testing in the all the levels of SDLC.

b. Level 1 – Feasibility study

Technical feasibility

- \checkmark Check the technical skills of team members.
- \checkmark Check the capabilities of team members.
- \checkmark Technology is stable or established.

 \checkmark Number of users increases then, how to handle the problem.

 \checkmark Number of users increases then, how to improve the product.

Operational feasibility

- Check the priorities of the user requirements.
- ✓ Divide the acceptable and non-acceptable solution by the team members.
- ✓ Check user will adapt new technology or not.
- ✓ Organization is support new technology and alternative proposed solution or not.

Economic feasibility

- ✓ Investigate the software, hardware, development team and training cost.
- ✓ Monitor the long term and short term gains of organization.

Organizational feasibility

 \checkmark Maintain organization summary and environment.

 \checkmark Maintain legal and corporate structure of organization.

Financial feasibility

- \checkmark Estimate the total startup cost needed.
- ✓ Check financial performance of similar business.
- ✓ Check the financial attractiveness of the proposed venture.

c. Level 2 – Requirement gathering

- ✓ Avoid poor priorities of user requirements.
- ✓ Check the client technical and problem solving skills.
- \checkmark Establish the communication for the clients.
- ✓ Check the clarity of requirements.
- ✓ Check the mindset of the client.

d. Level 3 – Requirement analysis

- ✓ Describe the required interaction between the system and environment.
- ✓ Check the non-functionality requirement of the clients E.g. Usability, reliability, performance, maintenance, etc.
- Check the main functionality requirement of the clients E.g. Security, error handling, etc.

e. Level 4 – Software Definition Document

- ✓ Check the background and objective of project/product.
- \checkmark Check the description of the project/product.
- ✓ Check the detailed characteristics of the proposed system and environment.
- \checkmark Mention roles and responsibilities of the
- ✓ Maintain the purpose of document, Etc.



Fig 2.1 Initial level testing f. Level 5 – Software Design High Level Design

- ✓ Check the overall structure of the project/product.
- \checkmark Check the main modules and their connections.
- ✓ Address the main nonfunctional requirements.
- \checkmark Maintain the coupling and cohesion.

Low Level Design

- \checkmark Check the inner characteristics of the modules.
- \checkmark Check the looping and control structure.

✓ Understand the relevant programming language skills.



Fig 2.2 Middle level testing

g. Level 6 – Coding level testing

- \checkmark Check the statement coverage.
- \checkmark Check the path coverage.
- \checkmark Check the program technique coverage.
- \checkmark Check the program condition coverage.
- \checkmark Check the program loop coverage.



Fig 2.3 Final level testing

h. Level 7 – Testing

- \checkmark To ensure that is no error in the product/project.
- \checkmark Get the proper test conditions.
 - ✓ Collect proper test variables and inputs.
 - \checkmark Use relevant automation tools.
 - \checkmark Write proper test script with conditions.
 - ✓ Validate all the functional and nonfunctional testing.

i. Level 8 – Maintenance level testing

- ✓ To perform all the maintenance testing in customer site. E.g. Preventive, Corrective, performance, perfective, adaptive, etc.
- To deploy the enhanced, migrated and changes software product or project.

3. RESULTS AND DISCUSSION

In the SDLC models, each stage contains lot of intermediate and sublevels. So, testing must be done in all the levels of the Software Development Life Cycle. It is not an easy process for testing in all the levels of SDLC. In testing level test, test only the test data and test variables. But, in this paper gives some better suggestion and solution for testing in the all the levels of SDLC. These solution or suggestion surely minimizes the number of defects to deliver better product or project development. Because all the level in the software development, this paper gives some better ideas for software development.



Fig 3.1 SDLC vs. Defects

In the above diagram, whenever developer or designer misses the requirements, the defects will be doubled in the every SDLC. So, testing must be done in all the levels of the Software Development Life Cycle gives better defect removal. The above solution or suggestion surely minimizes the number of defects to deliver better product or project development.



Fig 3.2 Better defect removal

4. CONCLUSION

This paper gives some better suggestion and solution for testing in the all the levels of SDLC. These solution or suggestion surely minimizes the number of defects to deliver better product or project. Testing is the most critical part of the Software Development Lifecycle, as it is something upon which the final delivery of the product or project is dependent. It is time consuming and an error defective process, therefore, enhanced techniques and innovative methodologies are requisite. This paper can enhance the existing testing methods, both for defect removal effectiveness as well as for efficient and reliable final product or project which not only meets the specified requirements but also provides with maximum operational efficiency and minimal defects.

REFERENCES

- [1] P. Ron. Software testing. Vol. 2. ndianapolis: Sam's, 2001.
- [2] S. Amland, "Risk-based testing:" Journal of Systems and Software, vol. 53, no. 3, pp. 287–295, Sep. 2000.
- [3] Redmill and Felix, "Theory and Practice of Riskbased Testing", Software Testing, Verification and Reliability, Vol. 15, No. 1, March 2005.
- [4] B. Agarwal et al., "Software engineering and testing". Jones & Bartlett Learning, 2010.
- [5] K. Bogdan. "Automated software test data generation". Software Engineering, IEEE Transactions on 16.8 (1990): 870-879.
- [6] Jacobson et al. The unified software development process. Vol. 1. Reading: Addison-Wesley, 1999.
- [7] Everett et al., "Software testing: testing across the entire software development life cycle". John Wiley & Sons, 2007.
- [8] J.Irena. "Software Testing Methods and Techniques", 2008, pp. 30-35.
- [9] Guide to the Software Engineering Body of Knowledge, Swebok, A project of the IEEE Computer Society Professional Practices Committee, 2004.

- [10] E. F. Miller, "Introduction to Software Testing Technology", Software Testing & Validation Techniques, IEEE, 1981, pp. 4-16
 - [11] M. Shaw, "Prospects for an engineering discipline of software," IEEE Software, November 1990, pp.15-24
 - [12] D. Nicola et al. "A grey-box approach to the functional testing of complex automatic train protection systems." Dependable Computing-EDCC 5. Springer Berlin Heidelberg, 2005. 305-317.