SURVEY ON SOFTWARE TEST STRATEGY

[1]PADMARAJ NIDAGUNDI, [2]VIKTORIJA PONOMARENKO

[1,2]Riga Technical University, Latvia
E-mail: [1]padmaraj.nidagundi@gmail.com, [2]Viktorija.ponomarenko@gmail.com

Abstract— Software testing becomes very important task before software needs to be used by the intended end customers. Software testing reduces the software building cost as well as an increase the customer satisfaction with software products. In recent years companies are spending their 30-40% budget for the software verification and validation process. One side new software development frameworks, software testing tools and automation brought a lot of advantage and also increase the complexity with manage the overall process itself. To reduce the software defects and make software error free, test strategy play the key role. The various techniques are proposed by the number of researchers about software test strategy. But lean canvas design utilization is the new approach and it provided the test team to design an appropriate test strategy in the SDLC and help to focus on the software quality goal easily. This review paper focuses on the various software test strategy in different development methodology and techniques.

Index Terms— software testing, software testing strategy, test process management.

I. INTRODUCTION

In today's world the use of the software is increasing rapidly and are considered as important means most of the data is stored with software in clouds or dedicated servers. Moreover, the developed software needs to work according to end customer and business requirements. Software development companies develop and manage their projects in various development methodologies such as scrum, waterfall, lean, and kanban. etc. In all kind of methodologies software testing is a key attribute to make the sure software work according to stakeholder's requirements. From a long time with software testing process is still following the traditional approach with long documentations, lack of test strategy that fits with the scoping of software.

In this paper, we are first discussing the classification of the different most famous IT industry adopted software development methodologies and different approaches for the testing. And finally, there is a survey on different test strategy approach for the software testing.

II. DIFFERENT SOFTWARE DEVELOPMENT METHODOLOGIES

The software development is done in various methodologies (SDLC). These methodologies adopted according to the team size, team member skills, time and budget.

Waterfall Model: It's one of the traditional ways of the software development process in software engineering. It has 6 stages In waterfall all process happen in a linear flow with itemized steps to let the user get next level in a progressive way on completion of the previous stage. Pros: it's easy to understand, easy to handle all steps, save time on testing & analysis. Cons: Difficult to judge the possible outcome of the project, Developer not able change code in the testing phase. Not bring ROI on long time ongoing projects.

Agile methodology: Agile contain itself many subs approaches such as scrum, lean, kanban and, extreme programming [1]. In general agile focus on the short software development cycles and review them every two to three weeks. Pros: Easy to respond to changes & more adaptive. Improve the defect's detection with small parts of the application in the early stage. Cons: Lack of documentations and more focused on building the software. Project outcome is not clear.

Prototype methodology: It is the method where developers make the valid prototype first before creating the authentic final solution. The prototype needs to work as the customer wants and later only add more requested features. Pros: Less risk of failure with developed functionality [2]. The functional process of the software is very clear to document and better for the requirement gathering & analysis. Cons: Many changes in the later stage can be risky to develop.

Rapid application development: This approach is focused on the get quick results. It provides the excellent development approach the requirements [3]. Pros: Makes development effortless and clients can get the quick review. Cons: Everything depends on the team effort and team members need to have the excellent skills. It is not suitable for the small budget projects.

Spiral model: It is best to approach with development and risk analysis. It contains the four phases and each stage risk is evaluated [4]. Pros: Avoid the risk in each phase of development and good for the large size, complex projects. Cons: More expensive for the software development. If risk analysis is the failure then it may bring all project in the trap.

III. DIFFERENT SOFTWARE TESTING APPROACHES

Software testing can be done with the different approaches with the scope of the development methodology, project scope, budget and skills with the

team members. Testing will be mainly classified as a functional, non-functional and maintenance testing. In industry different type of testing in practice with the different level of the software development.

Table 1: Literature Survey

No.	Author	Title	Testing type	Development Methodology	Approach	Test strategy
1	Gay. G	Generating Effective Test Suites by Combining Coverage Criteria	Search-based Test Generation	Waterfall	Build and vary their testing strategy based on the needs	Top Ranked, Boost, Unique Faults Strategy
2	P. Maragathavalli, A.Saranya	Prioritization on Software Reliability Assessment using Adaptive Testing Strategy	White-box test	Not mentioned	Coverage-based ART	Adaptive Testing Strategy
3	Rojas J, Vivanti M, Arcuri A, Fraser G	A detailed investigation of the effectiveness of whole test suite generation	Automated generation of unit tests	Not mentioned	Whole test data suite generation	Search-based software testing
4	Chunduri A, Feldt R, Adenmark M	An Effective Verification Strategy for Testing	Distributed Automotive Embedded Software Functions: A Case Study	V Model	Functional testing	Gaps and redundancies reduce Software verification
5	Kawaguchi S	Trial of Organizing Software Test Strategy via Software Test Perspectives	Independent verification & validation	Not mentioned	Software test perspective	Organizing software test strategy
6	Samer I. Mohamed	DevOps shifting software engineering strategy Value based perspective	Functional testing	DevOps	Value based perspective	Value based perspective
7	James M. Clarke	Automated Test Generation from a Behavioral Model	White-box test & black box	Waterfall process	Behavioral Model	Automating test generation
8	D. Janzen, H. Saiedian	Test-driven development concepts, taxonomy, and future direction	White-box testing	Agile	Test-Driven Development	Lower defect density
9	Ghazi A, Garigapati R, Petersen K	Checklists to Support Test Charter Design in Exploratory Testing	Black box testing	Not mentioned	Exploratory testing	Test design checklist

Stages such as unit, integration, functional, system, acceptance, beta, performance, security and regression tests famous among them.

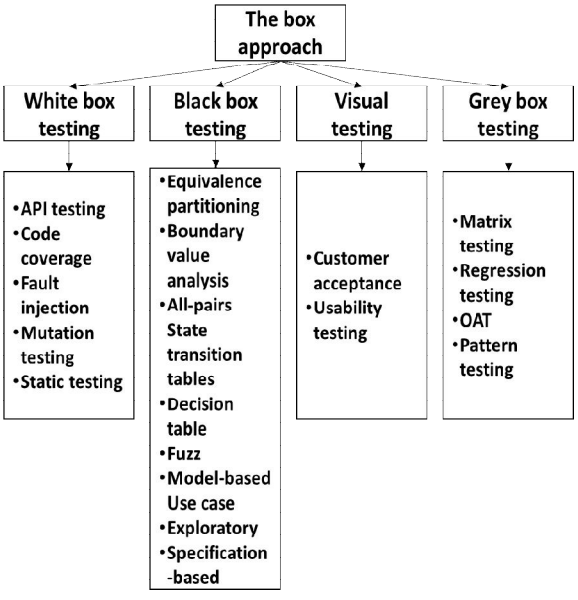

Fig. 1. The box approach for the software application testing.

Software testing types [5], [6], [8] are mainly categorized in the three subsections a) testing methods b) testing levels c) testing types. Testing methods illustrate is what is an appropriate method used to test the software and covers the static vs. dynamic testing, box approach, and gray-box testing (Fig.1). Testing level illustrates what type of testing need with software and it covers the unit testing, integration testing, component interface testing, system testing and operational acceptance testing. Testing types categorized in more sub parts with functional and non-functional tests.

IV. LITERATURE SURVEY

The literature survey above table 1 showcase the study about different test strategy in software development life cycle, it covers the importance of the test strategy and its adoption in the software development life cycle for the testing different software projects. When any project starts making test strategy is play a key role to make project successful within the time, budget, and resources. ISO/IEC/IEEE 29119-3: test documentation describes [7] the test management process documentation: this included the test plan & test strategy. There is no standard approach or methodology for creating the test strategy. All possible techniques approach and methods are discussed in this review paper.

CONCLUSION

To achieve the desired goal we need a high-level plan it can be done with forming the strategy. Test strategy determines an approach to different software

development life cycles. According to different authors, there are various types of software development and testing types with each individual projects not adopted the same type of test strategy based on the different requirements of the projects. In above survey, different authors testing strategy are not following any framework or structure. Having the clear test strategy help software testers to set the proper path for testing the software application and this will guide an entire team to focus and improve the quality of the software, it will also impact on the reducing the cost of software development.

ACKNOWLEDGEMENT

The present research has partly been funded by Latvian National Research Programme "Cyber-physical Systems, Ontologies and Biophotonics for Safe & Smart City and Society" (SOPHIS) Project No.2 "Ontology-based Knowledge Engineering Technologies Suitable for Web Environment".

REFERENCES

[1] Wortman K, Duncan B, Melin E. Agile methodology for spacecraft ground software development: A cultural shift, 2017, pp. 1-3.
[2] D. A. Fern, S. E. Donaldson. Tri-Cycle: a prototype methodology for advanced software development, 1989, pp. 377-386.
[3] Deepa V, Rexy R. Geetha, Rapid development of applications in data mining, 2013 pp. 1-4.
[4] Boehm. W, A spiral model of software development and enhancement, 1988, pp 61-72.
[5] Jain M, Gopalani D, Aspect Oriented Programming and Types of Software Testing, 2016, pp 64-69.
[6] ISO/IEC/IEEE International Draft Standard, Systems and Software Engineering, Requirements for Testers and Reviewers of User Documentation, 2017, pp 1-49.
[7] Software and systems engineering Software testing Part 3:Test documentation ISO/IEC/IEEE 29119-3, 2013, pp 1-138.
[8] Venkatesh R, Shrotri U, Zare A, Agrawal S, Cost-effective functional testing of reactive software, 2015, pp 67-77

★★★