

Survey on Test Driven Development for feasibility: An Industrial Approach

Pramod P. Jadhav¹ , Dr. Shashank D.Joshi²

¹ Research Scholar , Dept.of comp. Engg. Bharati vidyapeeth, Pune,Maharashtra, India

² Professor in Dept. of Computer Engg. Bharati Vidyapeeth, Pune, Maharashtra, India

Abstract: *Most of the researchers has a question in their mind whether Test driven development (TDD) concept is really feasible for development?. To find out the reality we conduct a survey on the basis of industrial people. This paper is based on survey and tries to provide nucleus conclusion from resulting analysis.*

Test driven development (TDD) is a Agile development practice. In this concern test driven development has five development stages. Test case, run the test, write a enough code, run test again, and Refactoring. Development of this Research is based on these different stages. But to given back ground/support to this work is, to provide good literature survey. In literature survey most of the paper is for based on the empirical study of industrial and academic point of view. The given survey tells that the test driven development is advance testing as well as development strategy in agile development.

Keywords: TDD, Testing, SRGM, Development

1. INTRODUCTION

To checking the feasibility of test driven development, first we should know the importance of Agile development process.[8] Agile method basically a very important method which give a technical base of development. This is general process of software development, but more focus on testing ,it will also show the better performance as compare to traditional testing.[4] Lot's of paper are prove that how this method are impressive and getting more accurate result in terms of requirement. Agile testing or agile process is not only support the software development but also give a better reliability, better effectiveness in overall process.[4] This is a concept which only give a theoretical support. to take mathematical support we are expected to get mathematical formulas like

a Cobb-Douglas production function, Goel-Okumoto (G-O model) function.[5][6]

Improve the software reliability is research goal in software engineering. To improve this, software reliability growth model (SRGM) is a major technical supporting model. SRGM proves the mathematical formulas which give a major support for reliability. In SRGM, reliability theory, mathematical base and its revised version of formulas is also given. Along with the G-O model, and Cobb-Douglas production function, it will focus on Yamada, Parreto, and more updated version of this formulas.[5][6] Most of the Research paper are taking the base of this formulas and derive their research problem statement.

Only SRGM model is not the sufficient to prove the reliability but along with that we must take the other evidences like tools, parameter, experimental setups, technical surveys, etc. and many more conceptual factors to prove this. Technical survey is also an important part because technical survey gives us a facts and figure to which we are very clear to our research idea. Conceptually it is very hard to convince the said research concept/problem.

Few papers are support on applying TDD on Automation system engineering process. And introduce "Test driven Automation" system. Automation system is important because it indicate that how test-first development is used in business software development. But most of the time the initial discussion is on challenges and research component in Automation system. For that researchers are take a help of different component. Few Paper [1] focus on Packaging automation, testing and diagnosis aspect in test-Driven Automation (TDA). To making testing concept more reliable, need to make Automation system more flexible.[1] So that it is easy to shift functionality from hardware to software solution. But at

the same time developer face new problem and try to solve complicated problem through TDA system. To tackle with such problem developer/tester are uses, construction, re-factoring, verification and validation concept.[1]

Another solution of such problem is Dual-Targeting concept. To use this concept developer has to know, target hardware bottleneck, and what was the problem in using Target bottleneck. To handle complicated problem, researchers was suggest to use Dual-Targeting concept. Dual target means any code will design to run on two platforms: first is final target hardware and second is won development system. Little paper [2] was giving the example of how to handle dual-Targeting system. Developer has to focus on, risks of dual-Targeting Testing, embedded TDD cycle, and its structure also. [2]

2. BACKGROUND STUDY

Most of the researchers discuss a mantra of TDD. For this discussion they will take a help of graph and there different tools. Framework of embedded unit testing is playing an important role in its functionality. For better performance of TDD, Mocking Hardware with UML is good option. Some Paper [3] lights on different methods of mock replacement policies. E.g. Interface based mock Replacement, Inheritance based, composition based, link-Time Based, Macro preprocessed mock, etc. TDD is useful for embedded software; to prove such concepts, developers take a help of supporting concept like, Mock driver code, Test on Host, test on Target, etc. To successful execution of embedded TDD patterns it will concentrate on 3-tire architecture. 3-tire TDD structure is based on 1) Hardware independent 2) Hardware aware and 3) Hardware specific concept. [3]

Our literature survey indicates that most of result or assumption is drawn on the basis of survey. Survey gives a background of any concept, some time survey provides a platform to specific study. To get a clear idea from different assumption, survey or empirical study/ finding, helps a lot. [9] Survey gives a concrete idea and conformity of present work. There are different type of survey are available. For this specific topic (TDD) survey are either academic or industry people base. The attribute of the survey are different, some attribute are based on technique of TDD, some are based on methods of TDD, some are show an effectiveness of TDD, and some are

show a behavior of TDD etc. Survey on TDD tries to prove the properties of TDD in positive way.

3. STRUCTURE OF SURVEY

We had made a survey, on combining above few points. In our survey we drew a 10 different question on TDD. These questions are proving different fact on TDD. For this survey we select a people/res-ponder which having an experience of more than five year. To make this survey successful we select a small as well as big scale industry. So that it will easy to analyze, whether TDD is really effective in small or domestic industry, as like a Multinational (MNC) industry. For this purpose we select Quick heal and hexagon as a domestic, and Cognizant, TCS (TATA Consultancy service) as a MNC. The responders of above Software Company are working as developer as well as Tester, because development and testing are co-related term in test driven development. In TDD, developer should work as Tester and vice a versa. The question of This survey is focus on different perspective of TDD. We are taking this survey online through www.surveymonkey.com. After this survey, collected result will analyze, and draw a graph on the basis of result.[4] After analyzing given survey few concluding remark are also drawn.

4. DATA ANALYSIS AND RESULT

In this survey 10 different question are framed. And as mention above, this question are provided through www.surveymonkey.com. Another side of this survey is, same set of question are provided through offline also. We took a hard copy of this survey and distribute among different software developer who were working in multinational software Company (MNC) and domestic company. Out of that few are like, persistence, Infosys, Practo, Cybage, Quick heal etc. We got same result as like a online. We analyze these question one by one.

4.1 QUESTIONS

a) Question 1:

1] Are you aware about Test driven development (TDD) a Testing technique? - This question gives an idea about knowledge of responder regarding TDD. Option are A) Little bit, B) Average, C) Good, D) Very good. Most of the responders are having good knowledge of TDD. Few are average knowledge of TDD.

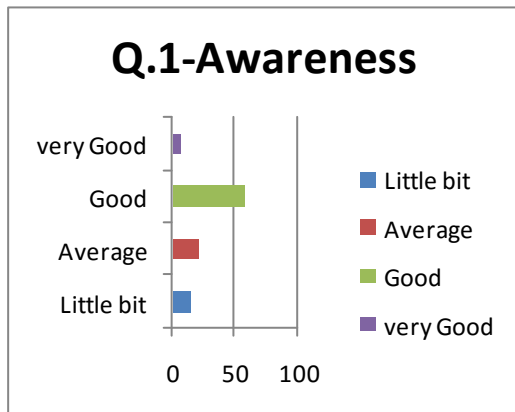


Chart.1- Analysis for Awareness

Above graph show 57.14% people are aware about the TDD concept. And rest of the people little bit and average aware about the TDD. It indicates that maximum responder is aware about this concept. So we move towards the further question, and let us see how these people are reacting with the TDD.

b) Question 2:

Next question is 2] **Do you think Test Driven Development is Effective technique for development?** This question shows an overall effectiveness of TDD in regular use. To analyze this question we are more focus on result of domestic company. And the result is surprise; because 90% responders in domestic company are saying 'yes'. Options are yes, somehow, and No. It indicates TDD is effective in different perspectives even in small scale industry.

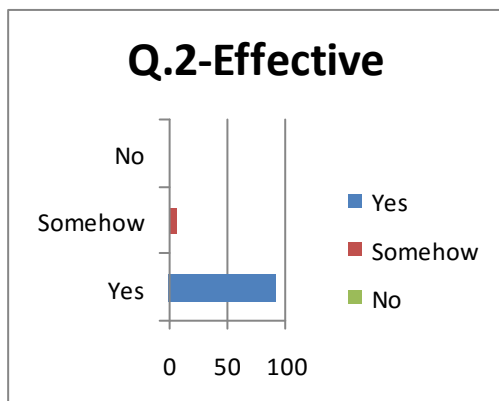


Chart.2- Analysis for Effectiveness

Above graph indicate that, TDD is an effective technique. If we look out the percentage, we get a drastic result from this graph. 92.86% people are saying yes it is an effective technique. How it is effective, let's focus on further question.

c) Question 3:

Next question is 3] **Is TDD is easy to use?** To getting proper result, we are again focus on domestic industry. Because if domestic people are saying it's easy, then tools and other supporting perspective of TDD is least bother. Because most of the survey says that, different tools like JUNIT is matter a lot, in performance of TDD. Provided option are like this - A) Very difficult, B) Difficult, C) Neutral, D) Easy, E) Very easy. Few people say easy, and few people say it is neutral to use.

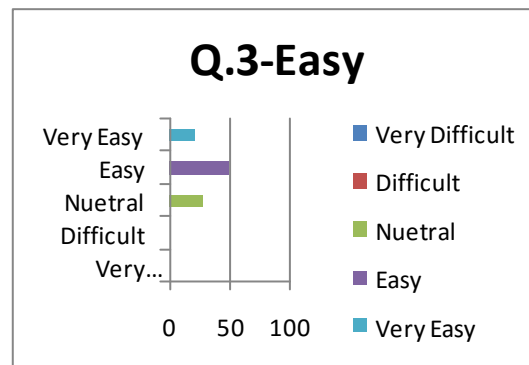


Chart.3- Analysis for Easiness

The result of this question is positive, because Domestic as well as MNC industry people are agree and say TDD is not difficult. 50% people say that it is easy. 28.57% people say it is neither difficult nor easy to use. And 21.43% developer says it is very easy.

d) Question 4:

Next question is 4] **What kind of benefits is achieved by using TDD?** This question is important because we come to know the actual benefit of TDD. Options says that A) Increase quality of software, B) Increase efficiency of software, C) Increase effectiveness of software, D) increase productivity of software, E) All of the above. We get a harsh result from this survey, because almost all responder responded to all of the above. If we focus on option, quality will increase when efficiency and effectiveness are get increase. So after improvement of all of the above term, productivity will automatically increase.

Hence we get most of the benefit from TDD. Industry people are known about this behavior of TDD. So industry people itself are the evidences for property of TDD. Above option are considering as the property of TDD.

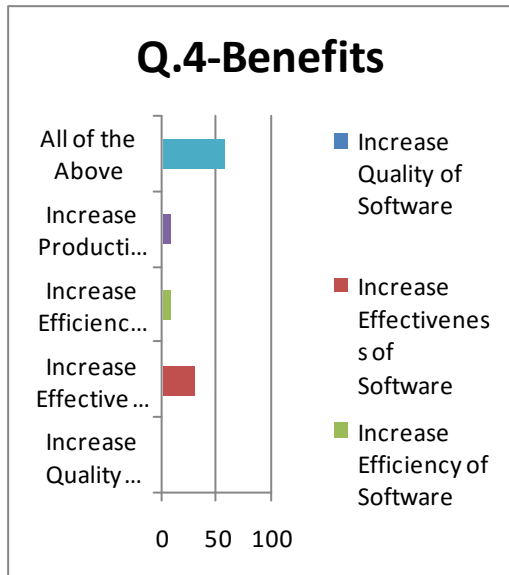


Chart.4- Analysis for Benefits

Regarding the advantages of TDD we refer a slandered paper which explains the different benefit of TDD. Our survey is also focus on different benefits of TDD like Quality, efficiency, productivity, effectiveness etc. 28.57% developer are says it increase the effectiveness of product. Few people are say it increase efficiency and productivity. And maximum people are agreeing to say all the four qualities are increase because of TDD. In this survey, this question is very important because the result of this question is proving that TDD is a concrete methodology for development.

e) Question 5:

Next question is **5] Is development speed is increase by adopting the TDD?** After meeting to developer he told that, the process of TDD is look lengthy, but actually it is not. And it's really increase the speed of development. But opinion of most of the people is in favor of option Good. We are provided the option like A) Little bit, B) Average, C) Good, D) Very good.

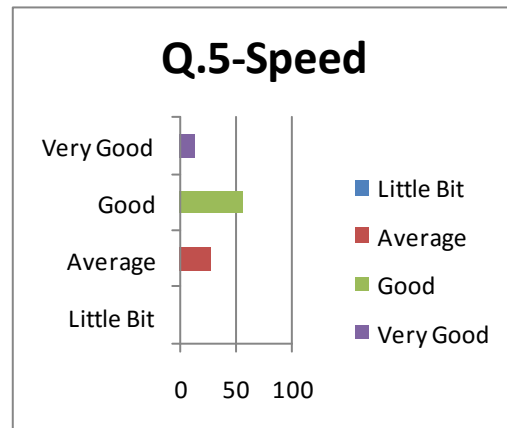


Chart.5- Analysis for Speed

If we look at the facts and figure of this result, 57.14% people says speed is increase good enough. Few are saying average speed is increase and few are saying maximum speed increases. So the TDD technique is better for software development. Due to lack of time, customer haven't give enough time to industry for making product, in this situation TDD like concept having a lot of weight-age for software development.

f) Question 6:

Another question is **6] which technique is most preferable, Test driven development or traditional?** Obviously after getting this much amount of advantages, opinion is in favor of TDD.

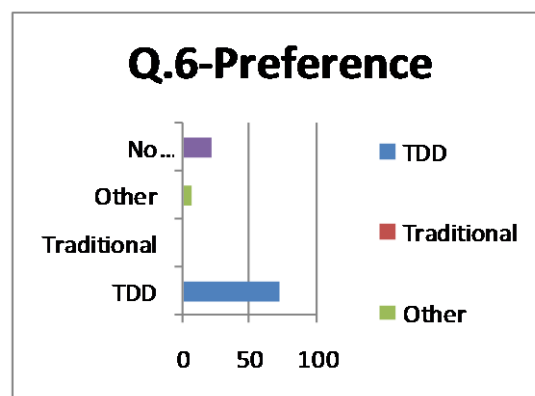


Chart.6- Analysis for Preferences

Obviously 71.43% people prefer a TDD technique, but 21.43% people are not ready to give their opinion. And still 7.14% people are using other technique. As compare to other technique TDD is having higher market.

g) Question 7:

Next question is 7] **How much Test Driven Development is affect on reliability of software, as compare to traditional technique.** To prove this, most of the IEEE and other standard paper are preferred useful production function, like Cobb-Douglas, Goal-Okumoto (G-O Model). Along with this other supporting fact and figure are indicate, TDD is more reliable than Traditional one. So our provided option are whether A) Decrease reliability in good form or B) Decrease reliability slightly, in another side whether it C) Increase in good form or D) Increase reliability slightly, and finally neutral or E) No Impact. As per the developer thinking it increase. Graph is also showing same result.

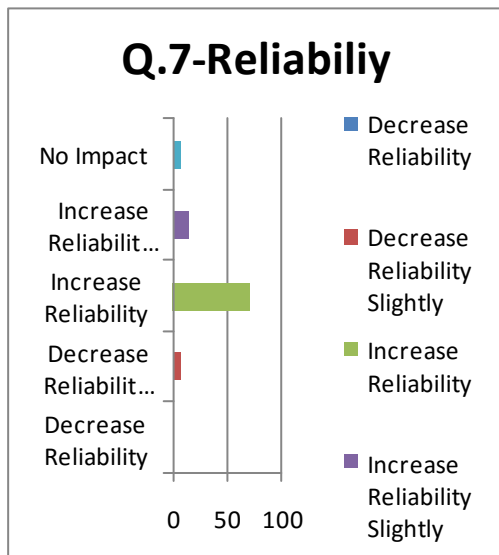


Chart.7- Analysis for Reliability

As we discuss above, there are lot of other method for increasing reliability. Different mathematical formulas are available as well as standard production function are also available. But all these formulas and function are with TDD, so it get a large affect on reliability. More than 70% developer is say it increase the reliability. 7.14% are saying slightly decrease reliability. And 7.14% are says there is no as such impact on reliability while using TDD.

h) Question 8:

Next question is 8] **which factor are essentially improve to ease use of TDD.** This question is quite different, because it indicate requirement to ease use of TDD. The option which is provided by us is A) knowledge of TDD, B)

Design technique, C) Development time, D) Experience people, E)All of the above . Few people are says knowledge of TDD is important. And most of the people says that along with knowledge other factor like, Design technique, time, etc. is also important.

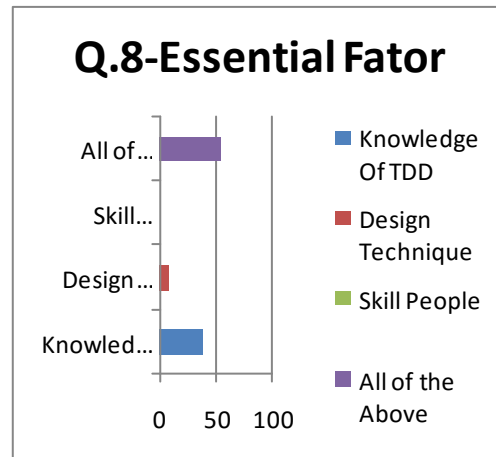


Chart.8- Analysis for Essential Factor

Yes TDD is wonderful, but 53.85% developer says that to ease use of TDD we required all of the above i.e. knowledge of TDD, design technique, and skill people. How to use TDD is not a big deal. Because the result of third question says it is easy to use. Simple steps should be following while using TDD. So there is no complicated process or algorithm for TDD.

i) Question 9:

Another question is 9] **what is affect on Defect Density while using Test Driven Development as compare to traditional technique.** This question is on defect density of program. Industry people are very much sure that defect density will decrease, because after using TDD concept occurrences of defect or error will automatically reduce, so defect density is also decrease.

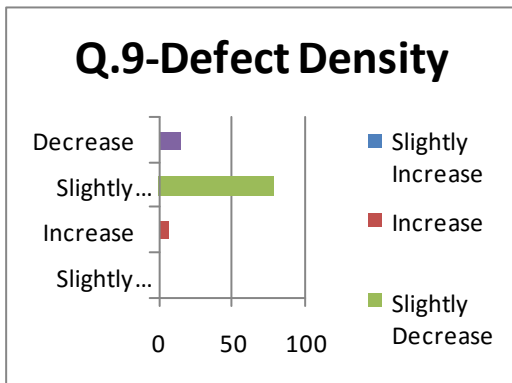


Chart.9- Analysis for Defect Density

By observing the TDD cycle, it shows simple steps. Because of test case domain principle, and refactoring concept, most of the problem or query gets resolve in the running process of TDD. That's why rate of detecting the defect is very low. At the end of project, or shipping time of project there is no notable or preferable defect in the project. So as per above result defect density is decrease, 78.57% people are agree for this conclusion.

j) Question 10:

And last question is 10] **what is affect on productivity while using Test Driven Development as compare to traditional technique.** As per the above discussion, and focusing on different properties of TDD it will largely affect on productivity in positive way. So productivity is increase while using TDD.

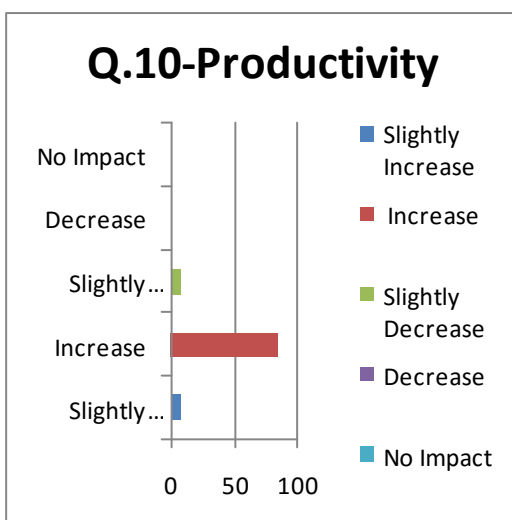


Chart.10- Analysis for Productivity

Same scenario is happen with this question. As per the principle of test driven development, it resolve maximum query when process is going on, and hence time is save per project. So it directly affects on productivity and finally it increase productivity. Above facts and figure are in favor of the said conclusion. 85.71% developer is saying that productivity is increase. 7.14% people are agreeing for slightly increasing productivity, and same percentage people disagree for this result.

5. THREAT TO VALIDITY:

As we focus on different survey on related topic, the validity of our result is subject to vary in several supporting factor of survey. While making a survey lot of other factor should be consider like, scale, nature of question, participant, experience and knowledge of participant, applying method, etc. Let us talk regarding the scale, if same survey is taken in a large scale then coming result will most accurate. Along with this, nature of question is matter a lot, because confidence of participant is may vary according to nature of question. Another side of survey is level and choice of participant. Survey based on academic participant may vary the result, because of actual working in industry and theoretical assumption is make a difference. Still we will see the different survey on academic, and few are made their conclusion based on both academia and industry. Another factor is experience and knowledge of participant. A more an experience people will get better result. Obviously knowledgeable people made our survey more powerful. And last but not the list, method. The way participant use or handle the concept, how they tackle with the subject of survey, and which methodology they will adopt, all these factor or threat give a significant validation in any conceptual survey.

6. CONCLUSION REMARKS

The principal conclusion drawn from our survey is that Test driven development is easy to adopt. And TDD concept gives a lot of benefit to developer, so it is a reliable technique. According to the literature review and studying different survey on TDD it is clear that TDD is effective and efficient methodology which increases speed of development, increase productivity, and quality of software product. [11] Maximum participant of our survey believed that TDD is most preferable technique in MNC as well as domestic industry.

During this survey we got drastic result from participant, which is in favor of Test Driven Development. Few result are really promising, but still it pointing out that TDD can be taken into account as an most reliable software development process.

TDD is a process which maintains a good relation with the normal process to agile process. This can be done with different conceptual aspect of testing. Test driven development is the part which gives a trust with agile software development practice. But while studying this concept, developer should know what are the limiting factor to adoption of TDD.[12] After analyzing these factor system will try to improve the process. TDD process provides different documentation aspect, as well as real testing experience to the developer.

Documentation point of view, agile process is a combination of different agile practices like, XP, Sprint, story, backlog, etc. To follow these practices will automatically maintain the data in a proper format, call documentation. [10] These documentation is useful while shipping the project to customer. We require all these documentation for a better transparency of software process. While adopting the TDD concept, developer having skill to write a test, develop a code, and use refactoring concept accordingly.

7. CONTEMPORARY ASPECT OF TDD

Few contemporary aspect is take into consideration, **first** is, test case strategy is a important part of TDD. While writing the test case it should be used in a uniform pattern. How developer writes the test case, where to write the test, in which pattern they write, all these factor is matter when test case is concern. **Second** aspect is following agile process, when developer/ tester apply TDD in his project, it should be follow a given pattern. Because agile process having lot of practical stages, as discuss above, all these having their won importance. **Third** aspect is requirement; this requirement is in case of customer, as well as programming point of view also. [7] Both the part is equally important, because customer requirement is first and important factor of development. Without any clear idea about customer requirement, projects will not success. To delivering up to date software, project manager/Team leader, having a deepest idea about customer requirement. This requirement may directly affect on software project, and start to end process. Another side is programming requirement,

because as the customer requirement changes programming concept is also changes. According to the customer requirement developer should use a advance technique of coding. Advance structure of coding is helpful for fulfillment of project need and hence software is complete with final conclusion.

8. REFERENCES

- [1] Dietmar Winkler Stefan Biffel Thomas Östreicher Vienna University of Technology, Favoritenstrasse 9/188, 1040 Vienna, Austria "Test-Driven Automation: Adopting Test-First Development to Improve Automation Systems Engineering Processes" EuroSPI 2009 – 1
- [2] James W. Grenning "Test-Driven Development for Embedded C" The Pragmatic Bookshelf Dallas, Texas • Raleigh, North Carolina 2010
- [3] Piet Cordemans, Sille Van Landschoot, Jeroen Boydens and Eric Steegmans "Test-Driven Development as a Reliable Embedded Software Engineering Practice "A Software Engineering Perspective, Studies in Computational Intelligence 520 © Springer-Verlag Berlin Heidelberg 2014
- [4] Shaweta Kumar, Sanjeev Bansal "Comparative Study of Test Driven Development with Traditional Techniques" international Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013
- [5] Hong-Wei Liu, Xiao-Zong Yang, Feng Qu, and Yan-Jun Shu "A General NHPP Software Reliability Growth Model with Fault Removal Efficiency" IRANIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING, VOL. 4, NO. 2, SUMMER-FALL 2005
- [6] Pankaj Nagar ,Blessy Thankachan Department of Statistics, University of Rajasthan, India "Application of Goel-Okumoto Model in Software Reliability Measurement" pecial Issue of International Journal of Computer Applications (0975 – 8887) on Issues and Challenges in Networking, Intelligence and Computing Technologies – ICNICT 2012, November 2012
- [7] Adnan Causevic, Daniel Sundmark, Sasikumar "An Industrial Survey on Contemporary Aspects of Software Testing" Punnekkat Mälardalen University, School of Innovation, Design and Engineering, 2010
- [8] Michael Hilton, Nicholas Nelson, Hugh McDonald, Sean McDonald, Ron Metoyer, Danny Dig "Using Software Changes to Understand the Test Driven Development

Process“ https://ir.library.oregonstate.edu/xmlui/bitstream/handle/1957/55027/TDDClassifier_ICP_C.pdf?sequence=1 2015

[9] Simo Mäkinen and Jürgen Münch **“Effects of Test-Driven Development: A Comparative Analysis of Empirical Studies”** Springer 2014. This is the author's version of the work. The definite version was published in Proceedings of the 6th International Conference Software Quality Days (SWQD 2014), Vienna, Austria, January 14-16, 2014

[10] Markus Hummel Goethe University **“State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development”** 2014 47th Hawaii International Conference on System Science 2014 IEEE DOI 10.1109/HICSS.2014.579 computer society 2014 .

[11] Yahya Rafique, Toronto Vojislav B. Mišić, **“The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis”** June 2013 (vol. 39 no. 6) pp. 835-856.

[12] Adnan Causevic Daniel Sundmark **“Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review “** Berlin, Germany March 21-March 25 ISBN: 978-0-7695-4342-0 2011