IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 2, Issue 5, May 2022
Impact Factor: 5.731

Software Organisation Agility and Testing

N. Sharmiya Sankari
Sri Venkatesa Perumal College of Engineering and Technology, Puttur, AP, India

Abstract: *In this paper was a review on study agility and testing processes in software organization we studied the differences in testing activities between software organizations which apply agile development methods and organizations which take the traditional plan-driven approach. Our focus was on the concepts which allow the software organization to successfully apply agile development methods or plan-driven methods. The agile software development movement brings both new energy and new hype to the software methods discussion. For this we have taken 12 organization units for observing a which organization unit uses best method this was done by several methods. Result and observation are recorded in this report.*

Keywords: Agility, Ground Theory

I. INTRODUCTION

Several different approaches can be applied to software development. The software development process is based on traditional plan-driven methods such as prototyping or waterfall or agile development methods such as SCRUM or Extreme Programming (XP). In theory, the main incentive for applying agile development in industry is to gain cost savings and faster development cycles, as the development is focused on communication and reacting to the process objectives instead of excessive design. Both plan-driven and agile development methods have their advantages and disadvantages. Hence it is seen that they have their own home grounds.

As testing is one of the costliest parts of the software process [1] – a common estimation is at least 50 percent of the development costs - testing activities in agile development are an interesting field of research. Applying agile development methods has effects on the software test process. Several agile methods promote early testing activities as a part of the program, meaning that the test phase is involved in the development.

In that way testing, is not left as the final phase of development, as in plan-driven methods, and there may be more time to perform testing activities and correct defects found in testing. In this study, we observed the effect of agile development methods from the viewpoint of the test process and test activities. Our goal was to determine how agile methods affect the execution of software testing in practice. The focus was on the organization of testing and on the test strategy. The study

was conducted as a qualitative one using the grounded theory research method. The research material consisted of interview data collected from three different interviewees from each of the twelve organization units participating in the study. This is a review about a study on agility and testing process in software organizations. This was based on study took by Jussikasurinen.

II. RELATED RESEARCH

In this review we have consider the researches related to the topic Software program testing with agile strategies were ordinarily case research. as an instance, numerous research has made experimental observations in a single business enterprise while applying agile techniques. The authors look at the subject by using concentrating on their reports of checking out practices in an agile development environment. but, a much broader angle is taken in a study by way of Itkonen et al. [4]. similarly, Ambler [2] takes software program checking out into consideration in his study that considers scalability of agile software program development. Ambler introduces a crew of unbiased testers within the software of agile methods at scale.

In the examiner byway of Talby et al. [5], checking out performed by developers is emphasised because excellent assurance is each crew member's duty. The trade off in checking out by means of builders is that the time required for trying out is taken from the development of recent functionalities. Talby et al. observed that the conventional view of applying an impartial tester is not sufficient in an agile development venture. Testers working in isolation from developers led to a nearly worthless method, and control noticed t at developers may want to correctly check the software themselves.

Testers working in close interplay with builders become seen as a greater appropriate technique for agile projects. however, Talby et al. did no longer provide any concrete results of this method.

Test automation was seen as a key factor in agile testing in Puleio's [6] study. With test automation, it was possible to keep testing and development in synchronization. In Puleio's project, they applied Scrum and XP together to experiment withagile techniques. Their results state that one of the main challenges they faced was software testing; insufficient understanding of software testing raised questions inside the development team, while communication between stakeholders was the key to solving this problem. Finding a common language was helped the team to communicate and understand the effort testing requires. Next issue was the estimation of the testing activities. Testers could not divide the testing tasks that are given to testers into appropriate pieces of work that would

be possible to complete during the iterations. However, estimations improved during the development process, even though one of the testers did not approve breaking down the testing into smaller tasks.

Based on these studies, it seems that test automation is a crucial part of testing in an agile environment. Another aspect is that testing practices used in plan-driven methods may not be compatible with agile processes. In addition, it seems that the role of testers is not as clearly defined in agile development as it is in plan-driven development. As for our study, these findings were used as a basis for qualitative research. Our objective was to observe how applying agile development, or agile practices, to the software process affects the testing process, and which the most prominent factors are that cause it.

III. RESEARCH METHODS

The method they have used for the research is specified here they collected data and analyse it result of the data is also specified. According to them Software testing at the organizational level has several aspects, which can be considered to affect the effectiveness of the test process, and ultimately, the quality of the end-product. These seems to vary between different types of organizations and software projects, as even in the seemingly similar organizations the approachto the software and test process may differ significantly. Some of this can be attributed to the human factors in the process. The testing work involves many actors, such as developers or testers, and communication between these actors, meaning that these factors should be addressed in observing the organizations.

A. Grounded Theory

Acknowledging these conditions, theyselected the grounded theory approach [3, 7] to conduct an empirical analysis of the software testing practices. The grounded theory approach was considered suitable, as observing and describing organizations and phenomena are its strong areas. For example, Seamandiscusses grounded theory research as a way to identify new theories and concepts, making it a valid choice for research in software process and testing practices, and therefore suitable for theirneeds. Grounded theory was first introduced by Glaser and Strauss [10] as a method for collecting data and creating theories in social sciences. The modern methodology has two main approaches: Glaser, as introduced inand Strauss and Corbin, as introduced in [7]. For

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 2, Issue 5, May 2022
Impact Factor: 5.731

this study, they haveapplied the Strauss and Corbin approach, which places a greater emphasis on the systematic categorization of observations and abductive reasoning [32]. In theory building, theyfollowed guidelines by Eisenhardt, with additional input and principles for interpreting field study results derived.

B. Data Collection

As per their studies, their main focus was set on the level of organizational units (OUs), as described in the standard ISO/IEC 15504-1. The organizational unit is a part of an organization, deploying one process or more within a coherent process context, operating within set policies and objectives. Typically, an OU is one part of a larger organization, but especially in micro- and small-sized businesses, as defined in European Union SME definition, an OU consist of the entire corporation. In other definitions such as TMMi, the definition of an OU is further elaborated. The day-by-day activities and management of the procedures in OUs are inner, but the sports are prompt from higher degree control with motivators together with company regulation and operational strategies. However, the OUs usually have some ability, albeit a limited one, to affect these steering motivators, with activities such as enhancement proposals or feedback. Overall, the OU was selected as an assessment unit because the use of an OU normalizes the company size and makes direct comparisons between different types of companies possible.

Table 1: Description of interviewed ou'S [10]

OU	Business	Company size[2] / Operation	Amount and types of agile practices in the organization
Case A	MES[1] producer and electronics manufacturer	Small / National	Low; explorative testing, focus on communication between stakeholders
Case B	ICT consultant	Small / National	Low; develops software as feature sets
Case C	Logistics software developer	Large / National	High; applies SCRUM [24]
Case D	Internet service developer and consultant	Small / National	Low; testing activities interweaved to development
Case E	Safety and logistics systems developer	Medium / National	Low to none; module reuse in non-critical aspects
Case F	Maritime software systems developer	Medium / International	Medium; applies iterative development cycles in most projects
Case G	Financial software developer	Large / National	Low to none; unit testing done by developers
Case H	ICT developer and consultant	Large / International	Low to none; unit testing done by developers
Case I	Financial software developer	Large / International	Low; piloting agile practices in development
Case J	SME[2] business and agriculture ICT service provider	Small / National	Medium; process able to adapt to the product, feature sets based on customer requests
Case K	MES[1] producer and logistics systems provider	Medium / International	Medium; daily builds, some SCRUM [24] activities, test automation
Case L	Modeling software developer	Large / International	Low; explorative testing

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 2, Issue 5, May 2022
Impact Factor: 5.731

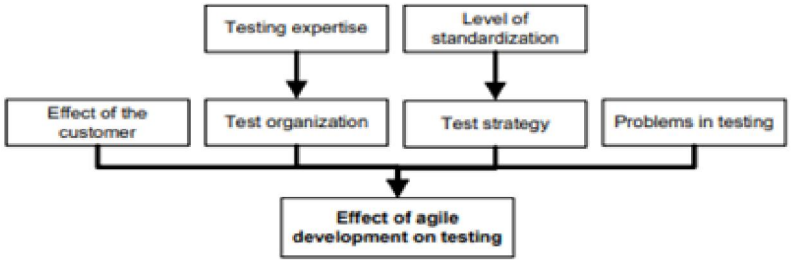

Figure 1: Central Aspects of Agile Devolepment That Affect The Testing Work

IV. CONCLUSION

In this paper, we presented our research on test processes in software organizations. Our study observed software development processes which apply agile methods or agile practices and examined how these activities affect the test process when compared to test processes based on the traditional plan-driven approach. from the very beginning to help predict the required testing time. Based on the observations made in our study, it seems that the software organizations which apply agile methods are in a position to achieve the goals Sumrell has suggested.

It seems that organizations which apply agile methods are generally more flexible in terms of changes and testing in the software process. However, testing in parallel with development work is difficult to execute and requires a reasonable amount of consideration from the organization. It could be stated that to successfully incorporate testing into agile methods, the development organization is forced to think and revise their testing processes in detail. On the basis of our review, we conclude that case c is the best case use of agility development. Which use higher agility method than case k because case k use middle than case c.

REFERENCES

[1] Bertolino, A. 2007. Software Testing Research: Achievements, Challenges, Dreams. Future of Software Engineering, 2007 FOSE '07, 85-103

[2] Ambler, S. 2008. Agile Software Development at Scale. Balancing Agility and Formalism in Software Engineering, 5082, 1-12, Springer, Berlin. http://dx.doi.org/10.1007/978- 3-540-85279-7_

[3] Glaser, B. and Strauss, A.L., The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine, 1967.

[4] Itkonen, J., Rautiainen, K. and Lassenius, C. 2005. Towards Understanding Quality Assurance in Agile Software Development. Proceedings of the International Conference on Agility (ICAM 2005), 201-207.

[5] Talby, D., Keren, A., Hazzan, O., and Dubinsky, Y. 2006. Agile Software Testing in a Large-Scale Project. Software

[6] Puleio, M. 2006. How Not to Do Agile Testing. Proceedings of AGILE 2006 Conference (AGILE'06)

[7] Strauss, A. and Corbin, J. 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques, SAGE Publications, Newbury Park, CA.

[8] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.

[9] ISO/IEC, ISO/IEC 15504-1, Information Technology - Process Assessment - Part 1: Concepts and Vocabulary, 2002

[10] Kari-Smolander/publication/220854534_A_study_on_agility_and_testing_processes_in_software_organizations

[11] Strauss, A. and Corbin, J. 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques.

[12] TMMi Foundation, Test Maturity Model integration (TMMi) reference model, Version 2.0, 2009.