IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

A Review Study on Agility and Testing on Software Organisation

Jotten Ben John[1], Shamsiya Shanavas[2], Dr.Neetha Thomas[3]
Santhigiri College of Computer Sciences, Vazhithala, Kerala[1,2,3]

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, SCRUM, Empirical Study, Extreme Programming, etc.

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 onagility and testing processin software organizations. This was based on study took by Jussikasurinen.

II. RELATED RESEARCH

In this reviewwe 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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

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 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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

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.

Intheir study, they haveinterviewed 12 OUs, which represented different operating domains and polar types of software industry, their selection varied from small, national companies to large international software producers; from electronics producers to software houses. The number of agile practices in the participating organizations varied greatly. One organization applied a complete SCRUM-based approach in software projects, while another had a completely design-based approach in software development. However, in almost all organizations some agile practices were applied; for example, explorative testing, iterative development cycles and feature set-based approach were common.

The participants can be roughly divided to three groups based on their agility: low, medium, and high, based on the number of applied practices. In the high level, an organization applies agile development method in all activities. In medium, the main development process may be design-based, but additional projects like feature development or updates, are done with agile approach. Also, organizations that applied several agile practices were considered medium. Low is the level where only individual users or individual projects applied agile practices. From "low" organizations, three applied almost strictly design-based development processes. On the other hand,one organization was dedicated in introducing agile practices to their processes. A brief description of the OUs - and their main agile practices – participating in all of the data collection rounds, is available in Table 1.

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

The initial population was selected based on their prior research on with the combination of the polar type selection method, meaning that the organizations were selected to be as heterogeneous as possible and included as many different types of business activities in the software development business as possible. The participants were selected from our research partners, and supplemented with additional organizations to fulfill the polar type requirements. All of the selected organizations were professional software producers of a high technical level, producing software as their main activity.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

Our objective was to gain insight into and understanding of the software test process in selected organizations, and later analyse which concepts affected the test activities and how the test organizations worked. To approach this problem, we decided to interview several people from the organization during three interview rounds, in which we used semi-structured questions on themes such as the development method, testing tools, test automation, perceived quality aspects and such.

Table 2: Organisation Interview Round [10]

Round type	Number of interviews	Interviewee role	Description
1) Semi-structured	12 OU interviews	Designer or Programmer	The interviewee was responsible for or had influence in the software design.
2) Structured and Semi-structured	31 OUs, including 12 OUs from 1st and 3rd round	Development or Testing Manager	The interviewee was responsible for a sofware project or a testing phase for a software product.
3) Semi-structured	12 OU interviews	Tester or Programmer	The interviewee was a dedicated tester or was responsible for testing the the software product.

The data collection was completed in three rounds, in which theyinterviewed software designers, development or test managers and actual testers, one from each participating OU. Typically, the interviews lasted approximately one hour and were held face-to face at a location selected by the interviewee, typically at their office. All of the interviews were tape-recorded fortheir later analysis. Should the organization not have had separate designers or testers, theyinterviewed the person, usually a programmer, whose responsibilities matched the desired role description. On two occasions, the OU selected two persons for an interview, as they considered that they did not have only one individual with sufficient knowledge on the interview topics. The interviewed persons were also allowed to see the questions beforehand and prepare for the interview if they deemed it necessary.

On the second round theyalso conducted a structured survey on the participating OUs. This was decided to gather also quantitative information on the test resources and current state of the overall test process. The quantitative survey wasconducted in 31 OUs, all of the 12 interviewed organizations included. As for the grounded theory analysis, the answers to the qualitative questions in the second round from 19 additional OUs were discarded. The qualitative data was subsequently used as an additional source of information in the data analysis, in establishing relationships between different categories. A summary of the data collection rounds and participants is presented in Table 2. The interview questions, which included such topics as development method, test phases.

They started with designers in the first round for test their prior observation on the test process and to see their assumption were correct. On the second face they have interviewed Managers, the decision to survey managers was made because they tend to have better understanding of the project- and organization-level aspects such as standards, testing policies and customer influence on the test process. As for the other estimates, the manager-level personnel were considered to be in a better position to assess such aspects as resource availability or personnel competence.

The third group that was interviewed was the testers. During this round, the focus was on testing activities and everyday testing. During this round, the same group of OUs was used as during the first round.

C. Data Aoihg cxnalysis

The grounded theory method contains three data analysis steps, which include open coding, axial coding and selective coding. During the first step, open coding, the observations from the interviews were codified to represent different phenomena and aspects that are related to the analysed topic. The observations were connected to groups called categories, which lead to definitions of relationships and interconnection between categories. The daily sports and control of the procedure in Ous are inner however the sports are recommendedfrom upper degree control with motivator such which include, axial coding and selective coding as company guidelines or operational techniques.

Interviews were codified to represent exceptional phenomena andelements which can be associated with the analysed topic. The observationswere linked to organizations referred to as classes, which causedefinitions of relationships and interconnection among categories.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

The categories were developed from "seed categories" , which in their studyresources", "Process problems" and "Role of the upper management" were a few of the applied seed categories In practice, the open coding section supposed that theone of a kind observations have been given the form of "class : phenomenon", such as "Test process: Applicability of agile methods" or "Test resources: Division of available resources", so that their relations and concepts connected to the categories could be further elaborated. During the coding, some of the prior categories merged, new categories have been developed, and a few classes altogether rejected by way of evaluating unique observations +to locate styles, similarities or regularities with each other. After the open coding, our data set collected from 36 interviews resulted to 166 codes in 12 different categories.

In the axial coding, the focus was on the causal conditions and relations between the existing categories. In this phase, the classes themselves were tested greater closely to establish any styles of connections among them on the larger scale. At this stage, the observations and codifications were becoming rigid, allowing the focus to shift towards defining a connection between larger concepts such as categories.

The objective for selective coding is the definition of the core category. The core category may sometimes be one of the existing categories, but it can also be an amalgam of categoriesshould none of the final categories be broad or influential enough.In this study, the core category was identified to be such anamalgam, consisting of the test strategy, the test organization, and the customer. Also, the existing, known problems of the testprocess were considered important part of the study, as they pointed out possible difficulties in the process of applying agile methods and software testing. Basically, these observations meant that instead of one influential aspect, the core category in this study was identified to consist of several test process and software development issues, listed in Figure 1.

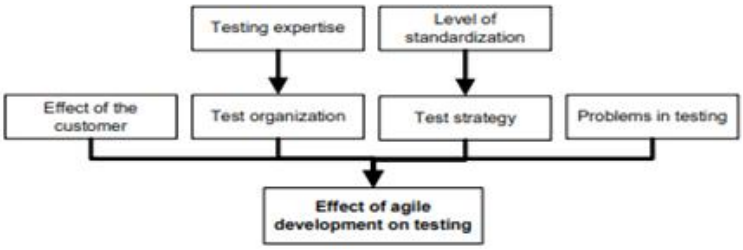

Figure 1: Central Aspects of Agile Devolepment That Affect The Testing Work

D. Hypothesis

We created the hypothesis based on the categories described earlier and observations from the organization units. The hypotheses were shaped to generalize or explain the observed similarities or differences, following the chain of evidence from study concepts to case observations and interviewee opinions. For example, the hypotheses 1 and 2 were derived from observations when comparing the different organizations, and the interviewee opinions on how the agile practices affect – or would affect – their testing work. Similarly, hypotheses 3 and 4 were summarized from two larger categories of observations; problems of test processes and customer participation. Fifth hypothesis was conceived from interviewee comments regarding benefits from applying agile practices. The observations, which were used as a basis for these hypotheses, are also available in summarized form in Table 4.

Hypothesis 1: Agile practices tend to allow more time for testing activities, while the total time for the project remains the same.Agile practices reserve more time for testing and speed up the project by dividing and delivering the developed product in pieces. This approach enables the implementation of the most critical parts of the software first, which allows testing to start early and focus on the critical parts first. Testing can be executed piece by piece during the development; hence the completion of the entire software is not required in order to start testing. In plan-driven methods, testing tasks may be discarded or downscaled if the implementation and specification phases require extra time. Especially if there are no separate testers, the developers are obviously needed primarily for building the required features.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

Hypothesis 2: Applying agile practices smoothens the load of test resources, but does not reduce the total amount required during the entire project.Delivering the product piece by piece enables the testing to start early. This approach may even be a more effective solution for executing tests, since the targets of the testing process are focused on certain parts of the software. the project, and used later on as a basis for additional tests. Testing is also done continuously during the project, which eases the required effort and need for test resources when compared to plan-driven methods. The need for test resources is not reduced because test resources are needed during the course of the project.

Case OU	Test organization	Testing expertise	Level of standardization	Effect of the customer	Test strategy	Problems in testing
A	Developers	Experienced developers	No standardized processes	Partly involved in development, does acceptance testing	New functions and changes, exploratory testing	No strategy or plans, resources (time, personnel)
B	Project manager or customer, customer service	Person specific, domain knowledge	Official, ISO 9001, not strictly followed	Participate in testing, relationship is important, close collaboration	New and core functionalities, explorative, resources are focused	Plan, strategy, resources (time, personnel)
C	Dedicated testers working in a Scrum team	Senior-level testers or test consultants, technical knowledge over domain knowledge	Company-specific practices that are followed variedly	Internal customer, close collaboration in development	Test everything or risk-based, rely on test automation, resources are focused, follows test plan	Documentation, senior-level testers required, resources (time, personnel)
D	Field testing	Users' know-how	Company-specific processes, no process for testing	Customer is not involved in development or testing	Core functionalities and critical targets, separate test projects	Plan, process, costs, resources
E	Developers and designers	Designers (testers) technical and domain knowledge	Company-specific processes, based on ISO 9001	Involved in testing and specification of requirements	Core functionalities, operation, coverage, plans and surveillance, test automation	Plan, surveillance is weak, resources (time, personnel)
F	Internal customer	Domain knowledge	Company-specific processes	Internal customer, arranges testing, close collaboration	New and core functionalities, explorative, resources are focused, test automation	Test automation, do not hire dedicated testers, resources (time, personnel)
G	Separate unit for testing	Testers' domain and technical knowledge and experience	Company-specific processes and policy, based on ISO 9000 – series, strictly followed	Internal customer, involved in development, surveys testing	New and core functionalities, operation, test policy, plan and surveillance, test automation	Maintenance of test automation, communication, inconsistent division of resources
H	Separate workgroup for testing	Domain knowledge	Company-specific processes, based on CMMI	Does acceptance testing, possibility to define tests	Functioning, resources are focused, test automation	Test coverage, plan, surveillance, resources (time, personnel)
I	Customer service, project manager	Testers domain knowledge, experience of project manager	Own internal practices and guidelines are followed, no process for testing	Feedback is important, no other involvement in development work	New functionalities, operation, coverage, project manager directs the testing, explorative.	Strategy, development of test automation, resources (time, personnel)
J	Separate unit for testing	Testers' domain and technical knowledge, and testers should know organizational practices	Official, ISO 9001, strictly followed	Accepts specifications, participates in acceptance testing, surveys the progress of testing	New functionalities, operation, coverage, follows standard, plan, surveillance	Strategy when testing new, resources (time, personnel)
K	Separate unit for testing	Domain knowledge	Company-specific processes, based on CMMI, are strictly followed	External customer accepts test plan, internal customer surveys development and testing	Operation, test policy, plan, some explorative.	Resources (not enough test environments), testing changes
L	Testing director	Testing directors' experience, domain knowledge of staff	CMMI, company-specific process for testing is obeyed	Internal customer, close collaboration, is involved in specification, surveys testing, does acceptance testing	Operation, coverage, stress, plan of the testing director, surveillance, test automation	Plan, strategy, no professional tester, need for tacit knowledge, documentation practices not followed

Table 4: Overview of the observations from Case OUs

Hypothesis 3: In order for the development and subsequently testing to be effective, the stakeholders have to understand and conform to the practices in agile methods. In order to use agile methods, the vendor organization is required to increase the amount of collaboration and communication with the

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

customer. Every stakeholder should understand the practices applied in agile methods. An agile development process requires a different approach to development from the customer point of view, as well. Requirements and specifications are not strictly defined at the beginning of a project; they are allowed to change and develop gradually during the development process. Customer participation is important throughout the project, since requirements are clarified during the course of development. In testing, the close collaboration shows in quick feedback and maybe in more effective acceptance testing, since testing can be performed gradually, as well. Another aspect is that old testing methods may turn out to be insufficient when agile practices are observed. This may lead to the OU being forced to think of the testing practices and activities that should be utilized in agile process.

Hypothesis 4: Internal customer supports the deployment of agile methods. In the principles of agile methods, the significance of customer collaboration is emphasized. In the studied OUs, the collaboration was considerably closer if the OU had an internal customer. For an internal customer, it is easier to participate in the development work continuously. It may be challenging to tie an external customer to the project, since agile methods require close collaboration throughout the project. In addition, they require reliance between the external customer and vendor in order to use agile methods without fear of serious disagreements.

Hypothesis 5: Applying agile methods allows faster reaction times for change and feature management in testing. In case I, the developer wanted to make changes to the product at short intervals, but still maintain the product in a deliverable state. In agile methods, the length of the development iteration can be set to fit the organization and the change. As testing is part of the iterations, and not just a phase at the latter stages of the project, the application of agile practices allows testing to be more flexible in case changes are needed.

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)

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

[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.