UPPSALA
UNIVERSITET

Agile Testing in Scrum

A Study of Major Factors for Successful Agile Testing Implementations in the View of Agile Testers

Samira Azarkerdar

Subject: Information Systems
Corresponds to: 30 Credits
Presented: VT 2018
Supervisor: Dr.Ruth Lochan
Examiner: Prof. D. M. Haftor
Department of Informatics and Media

Abstract

Agile testing, one of the most pivotal aspects of Agile software development, was first introduced in 2003. Since then, companies have been moving towards the 'Agile way of working' which is quite challenging indeed, especially for large companies used to working with traditional methodologies where testing is conducted only prior to final product release. In contrast, Agile methodology requires that testing be done after each iteration to detect bugs as quickly as possible. Therefore, traditional companies find it hard to adapt to the more recent, different methodologies. This case study aimed to illuminate the main factors which are of paramount importance in the Agile testers' perspective for the successful Agile testing implementation in an organization using Agile testing methodology with Scrum framework. These factors will be particularly useful for companies which intend to change their current methods and employ Agile ones to detect bugs as quickly as possible, accelerate product delivery, and elevate product quality. In this study, Cybercom as a company that aspires to interact effectively with customers and readily adapt itself with their feedback was selected. Data were collected through semi-structured interviews with Agile testers of Cybercom Group in Sweden, who presented their experiences about implementing Agile tests and made it possible for the researcher to collect empirical data and to evaluate methods of Agile testing in real Scrum teams. The interviews were transcribed, and thematically investigated. The findings indicated that (1) the importance of Agile testers' skills, (2) the importance of automated testing usage and its 'Dos and Don'ts' and (3) types of Scrum-based Agile tests and their implementation are main factors in Agile testers' perspective in successful Agile testing implementation. Recommendations on how these factors may be used to increase the overall efficiency are also presented. The researcher hopes to motivate companies that use Agile testing to focus even further on constructive interaction with customers and adapt more easily with customer feedback. Based on the findings of this study, some suggestions regarding Agile testing for the future projects were presented.

Keywords: *Agile Testing, Agile Software Development, Agile Testers, Scrum framework, Automated Testing.*

Table of Contents

List of Figures:

List of Tables:

1. Introduction

Agile software development emerged about twenty years ago and was immediately recognized as a significant software development model. Agile methodology has become known for being often enabling of adaptive planning, rapid product delivery, timely customer service, and for its capacity for constant improvement (Martin, 2003). The most important advantage of Agile methodology, however, is arguably its ability to immediately respond to change (Qasymphony, 2018). There are several varying types of Agile frameworks such as Scrum, Kanban, Extreme programming, Lean software development, and more. In 2001, the Agile manifesto was published by a group of independent thinkers called the Agile Alliance (Highsmith, 2001). The manifesto consists of 4 principles including:

(1) Individuals and interactions over processes and tools:

The first principle emphasizes the importance of communication, humanity and problem-solving. When individuals' work is characterized by flexibility, processes go forward more smoothly. Processes and tools are necessary elements for success in any project. Agile testing, therefore, not only does not overlook the significance of processes and tools but also helps processes go forward simply and efficiently. Testers use automated testing tools in order to increase efficiency and effectiveness. In actuality, the environment of Agile testing is highly customer-oriented and flexible enough to adopt new methods and ideas (Software Testing Fundamentals, 2018).

(2) Working software over comprehensive documentation:

The idea behind this value is that working with functional software invariably leads to progress and development. Agile testing does not promote the elimination of comprehensive documentation altogether; however, it demands that documentation should not hinder the testing process. For instance, Unit testing and Integration testing compose the software documentations of the system (Software Testing Fundamentals, 2018). In addition, a decisive factor here is that during the testing process, the progress of documentation with regards to achieving the objective is not evaluated.

(3) Customer collaboration over contract negotiation:

The third value refers to effective collaboration between developers, testers and customers, which is commenced by having extensive conversations. The idea is that a face-to-face conversation with mutual trust is more effective in helping the testers have a better grasp of the customers' demands. Agile testers can help customers and developers arrive at a common language (Software Testing Fundamentals, 2018).

(4) Responding to change over following a plan:

The last value encourages acceptance of change and adapting to it for the simple reason that changes can and will occur at any stage of the project life cycle. No one is able to stop changes

from happening; therefore, Agile testing embraces change and aims to turn it to the customers' advantage. Change in basic requirements is commonly a tester's nightmare, however; Agile testers are instructed and expected to welcome changes. Moreover, it must be noted that following the pre-determined plan is essential for any project to succeed, and projects that do not follow a clear plan are likelier to fail. It is very important for Agile testers to know what happened before and what is going to happen through guesswork. Agile testing does not eliminate the project plan, but it requires that no fixed test plan be in place. Agile testers need to be flexible and should respond to customers' needs and requested changes; there must be proper conditions for new features to be added seamlessly or for existing features to be removed at any given time to achieve customer satisfaction (Software Testing Fundamentals, 2018). But what is Agile testing?

1.1. Testing in Agile Methodology

Agile testing was created by Brian Marick in 2003. It is a customer-facing approach to testing, meaning that products are tested more often and there is more focus on the final product in order for Agile testers to be able to assess and improve it in simpler fashion (Crispin et al. 2009). Developers use *Test-Driven Development* (TDD) in order to write quality production codes. With TDD, developers write the test first, and in case they detect failures, they then write additional codes to allay the issues and proceed to the next functionality. Developers also write *integration tests* in order to make sure that the various sections of the code work together smoothly (Crispin et al. 2009).

Testing in traditional software development was only conducted when work on the product was finished and it was slated to be released. In traditional project coding, more time was needed and development teams spent longer periods in code-and-fix cycles; this led to situations where testing got *squished* (Crispin et al. 2009). Contrastingly, as mentioned before, testing in Agile software development is conducted for each iteration. In this way, developers hardly ever get ahead of testers because the work on no feature is considered done until it has been completely tested.

1.2. Who Is an Agile Tester?

Agile teams work dynamically, focus on doing their best, and continually seek improvements in order to deliver the best final product. An Agile tester is a member of an Agile team whose main duty is to carry out tests and assess the quality of the work in progress. Agile testers need to respond properly to change, have effective cooperation with colleagues in the technical and business departments, and clearly, understand testing concepts completely. Crucially, Agile testers should have adequate technical skills, know how to do efficient team work, be absolutely willing to learn, and feel obligated to deliver the products to customers in a timely manner (Crispin and Gregory 2009). Some suggest that having the right attitude is more important than technical skills. Crispin and Gregory assert that, "*Without the attitude, the skill is nothing*" (Crispin and Gregory 2009). Testers are customer-focused; this means that, in addition to considering the proverbial big picture, they should look at the application through a customer's point of view. Agile testers tend to help and support developers and customers alike; that is, they work cross-functionally, therefore they should not limit themselves only to solving test-related

problems. In brief, testers' main function is to supervise what is and/or is not working in the product (Crispin and Gregory 2009).

Testers in Agile teams have a crucial role in approving the code before it goes into production. Some of the key characteristics in all Agile testers are creativity, openness to ideas, willingness to do anything it takes to achieve higher quality, and focus on the customers' requirements. Good testers tend to understand precisely where codes fail and are aware of the available solutions to overcome failures (Crispin et al. 2009).

There are 10 principles of Agile testers including (1) providing continuous feedback,(2) delivering value to the customer,(3) enabling face-to-face communication, (4) Having courage to admit the mistakes, (5)Keeping it simple or applying a simple approach, (6)Practicing continuous improvement, (7) responding to changes, (8)self-organizing,(9) focusing on people, and (10) enjoying what they do.

1.3. Scrum Framework

The Agile framework Scrum was created by Ken Schwaber and Jeff Sutherland in the early 1990s. The term Scrum in software development refers to methods that emphasize sets of values, practices and empirical processes. Scrum is an iterative framework, with every iteration starting with sprint planning in order to determine what the team needs to do through the iteration. Team members work cross-functionally and, at the end of each iteration, the Scrum team should deliver "an increment of potentially shippable product functionality" (Schwaber 2004, p 12).

One of the main differences between Scrum and Waterfall (one of the traditional methodologies) is that in the former, requirements are well-defined at the beginning of each sprint, several tests are carried out within an acceptable timeframe, and the detected bugs are fixed during each sprint to simplify matters. Contrastingly, in waterfall, testing and debugging are performed before the final release, while the heavy documentation in this framework often means that making any subsequent changes is more complicated (Schwaber,2004).

There are several elements that distinguish Scrum from other Agile frameworks including (1) development and confirmation of codes by the Agile team, (2) testing codes in test environments and deploying the product in the production environment and (3) acceptance criteria are based on communications between the Agile team and stakeholders (Schwaber, 2004). There are also several Agile testing methods in Scrum encompassing (1) behavior driven development (BDD), (2) acceptance test-driven development (ATDD), (3) exploratory testing, and (4) session-based testing. Researcher would explain more about them in chapter 2 sections 2.2.

1.4. Research Problem

In traditional methods, tests are conducted right before product release, which means that testers would have to deal with a high amount of documentation during testing, would need much more time to detect possible bugs and would develop unwanted stress in testers (Crispin et al. 2009). Because of those weaknesses, researchers try to find ways to tackle such difficulties. The advent of Agile testing in 2003 opened a new horizon for testers and developers to address these weaknesses. Therefore, these days, most companies are inclined to employ an Agile testing

method, one which allows them to perform multiple tests during each sprint, thereby detecting any existing bugs much earlier than in older methods. This process leads to increased product quality (Crispin et al. 2009). This project is an attempt to investigate the main factors which are of paramount importance in the Agile testers' perspective for the successful Agile testing implementation in an organization using Agile testing methodology with Scrum framework, which should be particularly useful for companies which intend to exchange their current methods for Agile ones in order to quicken the process of finding bugs and product delivery, and create conditions in which constant improvement is highly facilitated. Although the Agile testers are inseparable part of Agile testing projects, little was known about Agile testing implementation and important factors involved from their standpoints.

1.5. Purpose of the study

Continuing with the problem introduced in the previous section and because the idea of Agile testers in Agile testing implementation was overlooked by former researchers, the aim of this thesis is investigating the important factors for successful Agile testing implementation in the view of Agile testers. To fulfill the purpose of the study, the IT Company- Cybercom Group- as an Agile company was recruited in order to clarify what factors are required to plan and realize Agile testing in the view of testers in an Agile company that uses Scrum frame work. Furthermore, in this circumstance, it is also relevant to provide a list of suggestions regarding how these factors may be used to increase the overall efficiency in the future projects.

The specific research question that was formulated to achieve this research goal is:

- What factors are of paramount importance in the view of Agile testers for successful Agile testing implementation in Scrum framework?

1.5.1. Why Cybercom Group?

Cybercom Group is the company with over 20 years of experience in information and communication technology. The international headquarters of Cybercom are located in Poland and India where it provides business services in the Nordic region such as Sweden, Finland and Denmark. Cybercom has 1,300 active employees across five countries. The main focus of Cybercom is on development of telecom tools and Agile software development projects. Cybercom prides itself on solving problems through technical expertise and sustainable solutions (www.cybercom.com, 2018). There are several reasons for choosing this company. First and foremost, this company works in the field of Agile testing and due to the simplicity of communication with the company. Second, this choice could provide the chance to cooperate with this company within the given timeframe to write this thesis.

1.6. Limitations

To address the research question, the researcher was searching a lot to find a company that is implementing Agile methodology. Since many Agile companies did not tend to participate in university research project, the process of finding a proper company was really time-consuming. Eventually, the company whose management finally agreed to granted the author access to its data

and work on it as case study was Cybercom (an IT consulting company). Unfortunately this company had no more than four testers among its personnel. It is crystal clear that a higher number of testers could have resulted in a much simpler and more accurate data analysis in the research.

1.7. Disposition

Chapter 2 includes a background of Agile testing application.
In chapter 3, the related literature is reviewed.
Chapter 4 elaborates the concept of explorative research, the data collection methodology and method for data analysis utilized in this study.
In chapter 5, the Themes of the research on implementation of Agile tests by the Agile testers are presented.
In chapter 6, the analysis of research findings is presented.
In chapter 7, the author discussed the findings extracted from interviews with the Cybercom's testers.
Chapter 8 contains the final answer to the research question of this thesis, while ideas for further research are suggested at the end.

2. Background: Agile testing application

In this section, a brief introduction to the Agile testing methodology is provided.

2.1. Agile Testing Quadrants

A summary of all the essential tests with regards to Agile software development as devised by Crispin and Gregory (2009) determine Agile testing quadrants (figure 1). They conceived there are four Agile testing matrixes and called them quadrants. The quadrant numbers follow no specific order but only refer to whether a quadrant is Technology facing, Supporting the team etc. It is worth noting that these quadrants do not work in Waterfall. Some projects start with Q2 because this quadrant contains specifications and tests that drive coding. Q4 puts the emphasis on the structure of features. When customers are not sure about the requirements, it is better to start with Q3 so as to use exploratory testing. These two quadrants require extensive coding. Companies that use Agile methodology, use the quadrants on a regular basis in order to work faster and to release products of higher quality. The quadrants also help Agile teams have better adaptive planning for their testing in the subsequent sprints (ibid).

Here, the author intends to briefly explain the four quadrants of Agile testing which comprise team-supporting tests, product-critiquing tests, business-facing tests, and technology-facing tests.

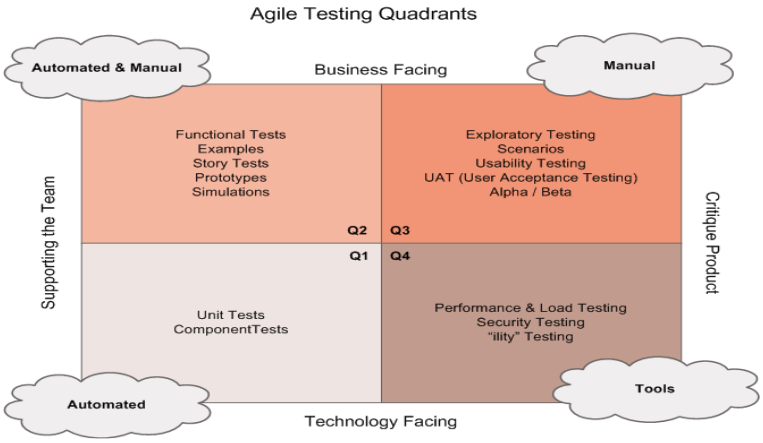

Figure 1: Agile Testing Quadrants

Image from *https://lisacrispin.com/2011/11/08/using-the-Agile-testing-quadrants/*

2.1.1. Team-Supporting Tests

Tests that support the team relate to product development. Tests in Q1 and Q2 are done more often. The aim of Q1 tests is to implement test-driven development in order to help programmers write the codes well. Unit tests confirm the functionality of the system and Component tests confirm the behavior of the system. Both are automated and are based on the same programming language as the application. The internal quality of a product is understandable by developers but, in most cases, not by business experts (Crispin and Gregory, 2009).

The purpose of testing in Q2 is to support the work of the development team at a high level. This means that tests are defined in detail for the customer team, with Business facing test to implement them at operational level. However, tests are written in such a way as to be

comprehensible by business experts using the business language familiar to them. Moreover, business facing tests need to be automated. To sum up, quadrants 1 and 2 provide quick access the information and enable the team to perform equally quick troubleshooting (Crispin and Gregory, 2009).

2.1.2. Product-Critiquing Tests

As Crispin and Gregory (2009) mention, it can be hard for a programmer to understand what a customer exactly wants. The term 'critique' refers to suggestions for revision and improvement. Q3 and Q4 are allocated to critiquing the delivered products. In Q3, the business facing test is categorized based on the experience of working with the software. The team does not have any competition. During the implementation phase of business facing tests on products, the author imagines how the user works with the application. In this quadrant, humans do manual testing i.e. testers.

They also add, in quadrant 4, technology facing refers to the types of Agile tests that have a critical role in various aspects of Agile software development such as performance, robustness, and security. Before starting the coding process, it is important that programmer become familiar with the features that are affected most by performance and security. This helps the programmer have an easier job of designing the code based on the customers' need.

In Agile software development using Automation tools is necessary for certain types of tests such as load testing, performance testing, regression testing and integration testing. In fact, it is Automated testing that creates the test cases and test scenarios required in manual testing.

2.2. Agile Testing Methods in Scrum

This section is largely inspired by a related article on Qasymphony.com (the complete address is mentioned under all image captions) and introduces all Agile testing methods in detail so as to facilitate the reader's understanding of this study.

2.2.1. Behavior Driven Development (BDD)

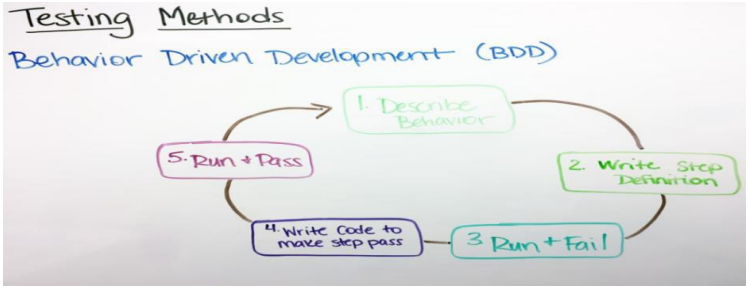

Figure 2: Behavior Driven Development (BDD)
Image from: *https://www.qasymphony.com/blog/Agile-methodology-guide-Agile-testing/*

It is a type of Agile testing that gets its inspiration from Test Driven Development (TDD). It is suitable for teams that make use of feature-focused software. Behavior driven development (BDD) is applied for high-level tests in business environments. In BDD, a set of initial

requirements are written based on the user's behavior which is determined through *human-readable* tests. These requirements serve as a guideline for testers whose specialty is in developing testing methods (QASymphony 2018).

BDD is different from traditional testing in that it requires test cases to be written as early as possible and calls for them to be executed toward the development cycle. However, BDD in an Agile environment means that tests are not based on requirements, while testing happens at the same time as development of the features (QASymphony 2018). Moreover, with Waterfall methodology, testers are tasked with writing the test cases. Contrastingly, a BDD approach reduces communication (or miscommunication) between the business analysts, developers and testers (QASymphony 2018).

BDD testers are expected to carry out the following:

1. Optimization of documents in order to keep track of the processes
2. Using the *Three Amigos* model when team members work together
3. Using a testing framework in order to define the criteria
4. Using an automated testing in order to reuse tests easily (QASymphony 2018).

2.2.2. Acceptance Test-Driven Development (ATDD)

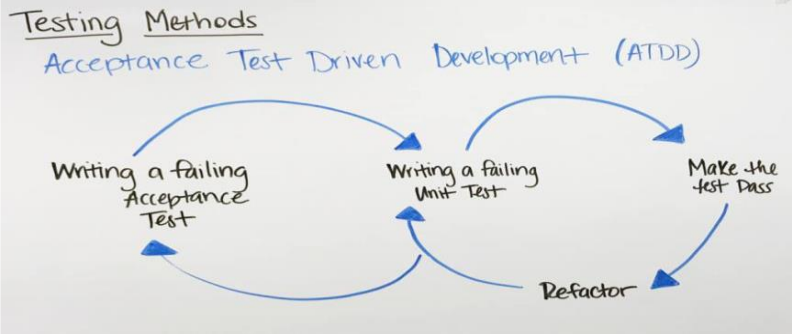

Figure 3: Acceptance Test-Driven Development (ATDD)
Image from: *https://www.qasymphony.com/blog/Agile-methodology-guide-Agile-testing/*

The second type of Agile testing is *Acceptance Test Driven Development* (ATDD). It is similar to BDD in that it needs to have tests written first, and in case the tests fail, codes will be written until the test is passed. ATDD tests are technical-facing unit tests, but ATDD tests are customer-facing acceptance tests. In ATDD, the user's perfect understanding of the product is as essential as its functionality, therefore it can help increase adaptiveness (QASymphony 2018).

ATDD is different from traditional methodology because it is not based on requirements of the testing process. In Agile methodology, ATDD is a *test-first* methodology. It is based on user experience with a high degree of adaptiveness. Some of the ATDD practices for Agile testers include: (1) Close interaction with customers in order to have a better understanding of their expectations, (2) Improving acceptance criteria based on customers' demands (QASymphony 2018).

2.2.3. Exploratory Testing

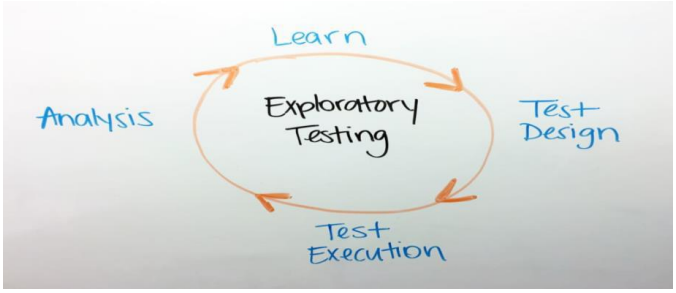

Figure 4: Exploratory Testing
Image from: *https://www.qasymphony.com/blog/Agile-methodology-guide-Agile-testing/*

Exploratory testing is a functional testing method and is relatively important in Agile testing environments (QASymphony 2018). It has been said that, "*Exploratory testing is a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project*" (Kaner.com 2018).

Exploratory testing has applications in both methodologies, but the main difference between them is that the interaction between testers and developers in Agile methodologies can be easier as regards the process. Exploratory testing is suitable for teams under time constraint; it is known to be able not only to reduce the required time but also to increase the code's coverage. The team member tasked with implementing the exploratory testing method is the tester (QASymphony 2018).

Exploratory testing is best-suited to testers who (1) Organize the functionality of the program, (2) Track the error repetitions, (3) Use automated tools and documenting results included in the tools in order to establish what has been tested by a given point in time and highlight the skill of testers (QASymphony 2018).

2.2.4. Session-based Testing

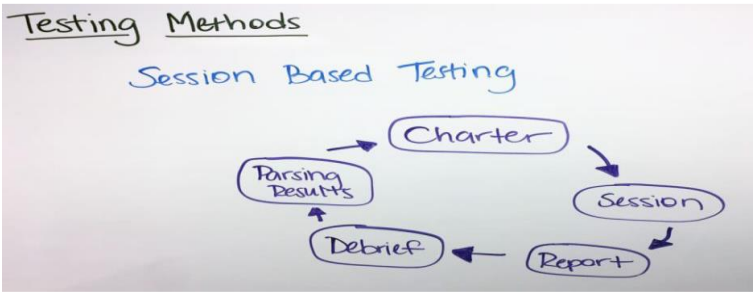

Figure 5: Session-based Testing
Image from: *https://www.qasymphony.com/blog/Agile-methodology-guide-Agile-testing/*

The last type of Agile testing is Session-based Testing (SBT), which is inspired by Exploratory Testing, although it is claimed to be more structured. In SBT, tests are conducted

during the available timeframe. Testers should report the outcome of the tests that are carried out in each session (QASymphony 2018).

Similar to Exploratory Testing, SBT is implemented in both methodologies. The biggest obstacle for testers is accepting the more structured nature of this test in each session. Session-based tests are performed by the team in order that they be involved throughout the process and remember everything afterwards. SBT also helps reduce the time it takes tests to finish and increases code coverage. Some of the considerations that session-based testers are expected to take into account are as follows:

1. Paying attention to session reports and documenting activities
2. Ensuring that tests are run without any interruption
3. Being clear about the software platform (QASymphony 2018).

In Agile software development – including Agile testing –various Agile methodologies and Agile tests are implemented. Among them, using Behavior driven development (BDD) in Agile testing provides the opportunity for stakeholders to be informed as to the projects' progress. In light of the fact that Agile testing reduces the implementation time of tests, the objective of some former research was to investigate how the testers can minimize the time it takes to achieve the target in each sprint and thus deliver the product to the customer more rapidly (Carrera et al., 2013). Moreover, the Acceptance test-driven development (ATDD) helps the Agile team through all steps of the development phase to have a better understanding of how the system should work based on customers' feedback (Pugh, 2011).

In what follows the relevant literature will be reviewed so as to propose the research question that the present study tends to answer.

3. Literature Review

This chapter provides a brief overview of a number of relevant studies and attempts to achieve a better perspective on the research question "What factors are of paramount importance in the view of Agile testers for successful Agile testing implementation in Scrum framework?"

This author chose the online database *Web of Science* as it provides a list of the most relevant articles based on the input given to the website's study narrower and helps form an initial idea about factors that are of paramount importance in the Agile testers' perspective for the successful Agile testing implementation in Scrum framework. The primary source of this research consists of articles found in the *web of science* database (at http://webofknowledge.com), while taking into consideration the literature on Information System (IS) and Human- Computer Interaction (HCI) resources. The specific phrases used by this author as keyword to search in the database were requirements of Agile testing in Scrum, testers' opinion in Agile testing implementation, successful factors in Agile testing, and Agile testing in Scrum.

The search was done between 1900 until 2018. The 205 records returned by the database were exported by the export tool on web of science and going through their abstracts helped this author pick the most relevant papers to be skimmed and scanned for more information. (The full list of the 205 articles is in Appendix B.)

Database	Key phrase	Refined by	Search results
Web of Science	Requirements of Agile testing in Scrum, testers' opinion in Agile testing implementation, successful factors in Agile testing, Agile testing in Scrum.	DOCUMENT TYPES: (ARTICLE & Review) AND TOPIC: (requirements of Agile testing in testers' opinion in Agile testing implementation successful factors in Agile testing, Agile testing inScrum) Time: 1900-2018 (All year)	205 docs

Table 1: Literature review search criteria

The results from the search in web of science revolved around Information Systems (IS) and Human- Computer Interaction (HCI), resulted in a total of 205 articles, but no more than 16 articles were identified as being relevant enough to factors of successful Agile testing, the history of Agile test, transition from the traditional testing to the Agile one, Automation in Agile testing, skills of testers, different types of Agile tests, requirements of Agile testing in Scrum , testers' opinion in Agile testing implementation, and Agile testing in Scrum. Others are mainly about the Agile project management, the quality of assurance, software testing, and product backlog.

Eliane Collins, Arilo Dias-Neto, and Vicente F. de Lucena Jr. (2012) and Vishvadeep Tripathi and Arivnd Kumar Goyal (2014) examine factors for having successful Agile testing. Eliane Collins, Arilo Dias-Neto, and Vicente F. de Lucena Jr. (2012) explore some testing Automation strategies that increase testing effectiveness. In their view, team collaboration, task distribution, testing tools, knowledge managements are of importance. Similarly, Vishvadeep Tripathi and Arivnd Kumar Goyal (2014) highlight that for providing successful Agile testing which factors should be considered. In their study, they overviewed the literature and extracted

main changes that Agile testers' practitioners mainly faced. Although reviewing literature so as to classifying the data is valuable it is not interactive at all. In other words, it may heavily rely on the researchers' interpretations of what have been mentioned in literature. In this study, the researchers concluded that the testers' responsibility is of importance. Team collaboration and Team management, Adopting Agile testing Mind set, Automated Unit, functional and regression Testing:, Emphasis on the feedback mechanism, and Crate foundation of core practice are main factors that lead to successful Agile testing implementation. The previous studies support factors determined by different researchers that lead to successful Agile testing implementation. Although those factors may vary, for instance, in the view of testers, developers, or other group of expertise, the former studies did not clarify such diversity and even controversy.

Lisa Crispin (2006), Sumanth Yenduri and L.A. Perkins (2006) and Van den Broek et al. (2014) investigate factors to improve Agile testing. The term 'Agile Testing' is to a large extent inspired by Lisa Crispin (2006). She indicates that using Session-based Testing (SBT) as a testing solution can help improve the quality of testing process. Crispin pays attention to the discrepancy between the customer's expectation and the delivered product and suggested that implementing SBT during each sprint in addition to having a clear picture of the customer's demand can help bridge the aforementioned gap. In this study, Cripsin (2006) only focuses on the session-based testing and overlooks other factors that may really involve.

Considering the importance of Agile programming, Sumanth Yenduri and L.A. Perkins (2006) in a case study investigate how using test-driven development (TDD) approach enhances the quality of developed software and its productivity. Furthermore, the costs in maintaining it at a later time will be dramatically reduced. In this case study although the researchers indicate how TDD approach can be beneficial for the company, their findings can be just applicable in small-sized software projects. In the similar vein, Van den Broek et al. (2014) shed light on how integration test is applied by Scrum. They present a visual model that integrates testing activities in Scrum. The paper concludes that Integration testing has a challenging face in complex projects but it can improve the quality of testing and in Agile methodology.

Jakobsen and Johnson (2008), Elallaoui et al. (2016), Berlowski et al. (2016) , Butgereit (2018) and Manu GV, Namratha M, and Pradeep (2015) study automated testing. Jakobsen and Johnson (2008) denote that Continues Integration is essential in order to check the code in the repository. This means that developers should check in all codes in the main repository before performing Automation test. As the authors clearly indicate their findings are just applicable in large Agile projects and cannot be practically used in small and medium-size projects. Moreover, Elallaoui et al. (2016) propose ideas that correspond with the principles of Scrum. One of the findings of Elallaoui et al. is that when all processes are to be carried out in automated fashion, and when system complexity increases, the test expertise is mainly needed to reduce test duration. In spite of investigating the idea that how important the test expertise is, in this probe, the researchers did not clarify which expertise they were looking for. Furthermore, in this study, all the findings are presented in theoretical level but they did not investigate the issues from the point of view of those who are practically involved.

Berlowski et al. (2016) put emphasis on Scrum-based automated testing. The paper concludes that, in complex projects, the testing process becomes a challenging affair, therefore, they suggested a number of effective approaches, based on principles of Agility, such as Continuous Integration, and automated testing in order to improve the quality of Agile testing. Their findings indicate that using automated testing helps increase the efficiency of the testing environment and improves customer satisfaction. Similarly, Butgereit (2018) asserts that automated testing has a positive effect on regression testing. Regression testing is implemented manually. Prioritizing feature development based on regression testing leads to "test debt." Each implementation is done with a certain amount of testing that may causes problems. Finally, though it is concluded that automated testing does solve many problems, but a lot of time is needed to develop Agile testing practices.

Manu GV1, Namratha M2, and Pradeep3 (2015) assert that from one hand Automation testing is essential and on the other hand for Automation testing implementation the company may face numerous challenges. In their view, because of various Automation tools, selecting a proper tool is one challenge. Additionally, executing Automation Project within an organization is really hard to manage. Since the right set of resources is a must when you are doing Automation, the resources should be skilled enough to design and code robust scripting, so that it requires minimum time for debugging during maintenance.

In short, Jakobsen and Johnson (2008), Elallaoui et al. (2016), Berlowski et al. (2016), Butgereit (2018) support the idea that Automation testing has constructive effects on Agile testing implementation. Additionally, GV1, Namratha M2, and Pradeep3 (2015) put more emphasis on challenges that test tools may bring about in test Automation. However, authors agree on how these tools may minimize the time for debugging. The present study, based on the former findings, attempts to once more take a closer look at the importance of automated testing in the Agile Company and represents the view of Agile testers in-details, those who practice Agile methodology in their testing tasks.

Searching to find the former research aimed at Agile testers, Puleio (2006) and Nidagundi and Novickis (2017) look at the skills of Agile testers. Puleio (2006) asserts that expert Agile testers in Scrum teams increase the cross-functionality of the Agile team. The author declares that deep knowledge of the methodology is required to do work estimation and task breakdown; therefore, calling on the services of a professional Agile tester with sufficient knowledge of their work almost invariably leads to improvements in team activities. Estimation is especially important and somewhat underestimated and neglected, according to Puleio, because developers (a) have a reputation for not being the best work estimators, (b) have regularly been accused of paying little attention to the set timeframe for an assignment, and (c) do not seem to care much for testing and, thereby, do not allocate sufficient time and effort to it.

Furthermore, Nidagundi and Novickis (2017) contend that Agile testers have a key role in Scrum teams when it comes to increasing the overall cross-functionality. Generally, testers are bound to encounter changes in the tests that should be performed in each sprint; therefore, they should have a wide knowledge of their work and adequate testing skills in order to increase the

effectiveness of Agile testing. With just the right amount of efficiency as well, adept testers would be able to increase project productivity meaningfully.

Theodore D. Hellmann, Abhishek Sharma, Jennifer Ferreira, and Frank Maurer (2012) , K.V. Jeeva Padmini, P.S. Kankanamge, H.M.N. Dilum Bandara, and G.I.U.S Perera (2018), N.Ganesh and S. Thangasamy (2012) and Ilia Bider and Oscar Söderberg (2017) shed more light on the significance of Agile testing, and study about its history and the transition from traditional testing to Agile testing. To be more precise, since testing is the pillar in Agile software development, Theodore D. Hellmann, Abhishek Sharma, Jennifer Ferreira, and Frank Maurer (2012) overview the history of Agile testing research and shed more light on some key factors that mainly influenced Agile testing studies. Although their investigation can draw a clear path for those who are willing to investigate in various aspects of Agile testing because they attempted to cover all relevant research papers, as they themselves assert they just selected the articles based on the papers' abstracts. In fact, selecting the papers merely based on the abstracts may lead to overlooking some that their abstracts may not completely cover the research domain.

Emphasizing the importance of Agile testing in increasing the productivity and reducing the time needed for changing or removing the sprint, K.V. Jeeva Padmini, P.S. Kankanamge, H.M.N. Dilum Bandara, and G.I.U.S Perera (2018) investigate that along with the mentioned advantages implementing Agile testing reduces the unwanted stress that testers may face during a project. They also scrutinize 15 challenges that testers usually face and their decisions to overcome the challenges.

N.Ganesh and S. Thangasamy (2012) compare the traditional testing with Agile testing. In their study, they mention that thanks to various advantages of Agile testing over traditional testing, more companies are inclining towards Agile testing. Furthermore, some do's and don'ts of Agile testing are presented that can be regarded as a useful guideline for those who are interested in Agile testing implementation. Moreover, Ilia Bider and Oscar Söderberg (2017) laid emphasis on the transition from traditional method of testing to Agile testing and called it non-disruptive transition. They also determined Agile Software Development mindset and its important features. In their view, the use of a process modeling technique is significant. What is worth mentioning in this study is that the researchers determined factors that mainly hinder an intended transition from traditional testing to Agile one but they did not suggest any solutions to tackle such difficulties.

Bearing in mind the importance of such transition, it is worth mentioning that investigating the factors involved in successful transition is of importance. It is clear that like any other paradigm shift, testers, developers, and other software practitioners need a guideline so as to find their ways and achieve their goals.

The review of related literature indicates that although there is existing research on the current trends in requirements of Agile testing and Scrum team in particular, the scholar that considers important factors for successful Agile testing implementation in the view of Agile testers is a relatively understudied area. In fact, considering the idea of Agile testing implementation and factors that dramatically lead to its success encourages the researcher to scrutinize this issue mainly through the lenses of Agile testers in a company which mainly tends to implement Agile testing

and from the perspective of highly motivated Agile testers. To bridge the existing gap the present case-study attempts to investigate what factors are of paramount importance in the Agile testers' perspective for the successful Agile testing implementation. The main aim of this study is to investigate important factors for successful Agile testing implementation in the view of Agile testers in Scrum framework. Furthermore, based on the research findings, the author provides a list of suggestions to increase the overall efficiency in the future projects. The intended audience of the study's findings is new Agile companies as well as those traditional companies which may decide to begin adopting the Agile methodology to quicken the process of finding bugs and product delivery, and create conditions in which constant improvement is highly facilitated.

4. Method

This section of the thesis elaborates on the concept of Explorative research and the data collection methodology utilized in this study.

4.1. Explorative research

This section states the underlying research paradigm used in this study. For this study, the author chose an explorative research methodology as the primary intention was to focus on one organization using Agile methodology in order to conduct an in-depth study on Agile testing, and provide a guideline for companies which tend to experience the transition from traditional testing to the Agile one.

Explorative research is rooted in interpretivism, therefore, it is *"concerned with understanding the social context of an information system: the social processes by which it is developed and construed by people and through which it influences, and is influenced by, its social setting" (Oates 2006).* This type of research leads to detailed analysis of a specific issue in order to achieve further insight as to the considered subject. Researchers often begin with a general idea and then use scientific research as a tool to expound the subject. The point of explorative research is not to arrive at a definitive answer (Saunders et al. 2003) but to use certain methods in order to gain more information. These methods include review and analysis of the literature related to the research question, interviews with the testers, etc.

4.2. Data Collection Methodology

This section explains the empirical data collection methodology used for this thesis. Explorative researchers tend to use primary data collection methodologies in order to collect the qualitative data (Oates 2006); this means that the data are collected personally but with the help of various testers via semi-structured interviews. The process of gathering the data requires more time. The possibility to use different data collection methodologies – such as observation, questionnaires, focus groups, interviews, etc. – was indeed there, but the author chose to use semi-structured interviews as it provided the chance to have (both face-to-face and on the phone) conversations, the testers' demand that they only be asked plain questions within their area of expertise (Scrum testing), and also because the interviewed testers did not have sufficient time for more elaborate types of interaction.

4.3. Interview Method

This thesis, Cybercom Group, is a company that prefers to use a modified version of the Agile methodology which it has developed according to its own particular needs. This author chose Cybercom first and foremost because this company works in the field of Agile testing and due to the simplicity of communication with the company. Second, such choice could provide the chance to cooperate with this company within the given timeframe to write this thesis. Cybercom has two separate teams, each consisting of seven developers and two testers. The testers have been selected in purposive fashion by the company. The data were gathered through semi-structured interviews in which the interviewees were not forced to comply with the ideas and classifications provided

by the author and also provided a situation which made it possible for the researcher gathering data to ask some follow up question from interviewees in order to gain more information. I had to conduct the interviews twice because while the determined research question was quite clear, the more answers and explanations one gets, the more comprehensive and clear the answers to the research question will be. The first series of interviews was done face-to-face at Cybercom's office in Stockholm, Sweden, where the interviewees answered the author's first question. The second series, involving the follow up question, was conducted by telephone.

Given that the research question of this study centers around important factors for successful Agile testing implementation in the view of Agile testers in Scrum framework, it follows that the interview questions should be related to the same topic and help answer it. To this end, the first question is related to the way the Agile framework is used in Cybercom. This question was deemed appropriate because the research question also related to a company that uses the Agile methodology. Moreover, Puleio (2006), Crispin (2006), and Nidagundi and Novickis (2017) refer to the significant role of the Agile methodology in improving cross functionality and increasing customer satisfaction. The second question, inspired by Schwaber (2004), is about the methods of choice within Agile teams themselves, because the way in which the Agile methodology is implemented in practice could have a significant impact on the quality and accuracy of Agile testing. Considerin that Cybercom Group, the case study for this research, uses Scrum, it was only logical that the focus be put on this framework. The third question comes from Crispin and Gregory (2009) and Puleio (2006) and is related to the process of performing Agile testing. This question, the author hoped, would help to quickly find a large portion of the answer to one of the research question. The fourth question, inspired by Nidagundi and Novickis (2017), and Puleio (2006), concerns the management of bugs via professional Agile testers. Since it is inevitable that during any testing process, testers should stumble upon several bugs and other issues, it seemed essential to include this question whose answer would help testers be more prepared and perceptive in handling these problems adequately. The fifth question deals with complications regarding product validation and was conceived by the answers to follow up questions during an interview. The question, furthermore, was also intended to provide another part of the answer to the research question, as verification and validation are two crucial stages in testing and no test may be considered complete without them. The sixth question is inspired by Jakobsen and Johnson (2008) and concerns the tools companies use for testing and influence the effective implementation thereof. The seventh and eighth questions concern the advantages and disadvantages of testing tools and resulted from the follow up questions during an interview. These questions shed light on how effective tools could be in performing the tests. The last question deals with acceptance testing because it is done at the end, immediately prior to delivering the product to the customers, and in companies that use the Agile methodology, it can be a part of the implementation testing.

During the interviews, the author did not eliminate any questions, yet asking some of the questions was not pre-planned and only took place due to certain answers to the original questions

which demanded follow up questions to become completely clear. A summary of the answers is included in the result section. The interview questions are displayed in Table 2.

Topic	
General Information	Why do you use the Agile methodology? What is your Agile method of choice? Why? How do you perform Agile testing? How do you manage situations where bugs are found? What are the problems you face regarding product validation?
Testing using different tools	What tools do you usually use for testing? What are the advantages of your preferred testing tool? What are the disadvantages of your preferred testing tool?
Release section	How do you perform acceptance testing?

Table 2: Questions of semi-structured interviews with Agile testers in Cybercom.

The face to face interviews were conducted in the meeting room of Cybercom Group headquarters, while the rest were via telephone. All interviews lasted nearly one hour. The interviews began with open-ended questions and carefully progressed towards more specific ones in order to produce more particular information. The researcher would first record the interviews in full, then proceed to produce transcripts of the audio recordings. At the end, thematic analysis was conducted on the resulted texts.

4.4. Thematic Analysis

One of the common forms of analysis in qualitative research is called Thematic Analysis. Braun & Clarke (2006) suggest that thematic analysis emphasizes the pinpointing, examining, and recording of patterns (or themes) within data. Themes have an important role in description of various phenomena (Daly et al., 1997). The aim of thematic analysis is, in short, to find the most considerable meanings in data sets. It is suitable for low levels of interpretation and it is flexible and useful research tool in order to provide rich, detailed, and qualitative data (Braun & Clarke, 2006).

Braun and Clarke (2006) state that there are three main factors that distinguish the thematic analysis from other methods. First of all, thematic analysis is appropriate for researchers who do not intend to devise a new theory, and because this thesis does not do so either, there was no better choice available. Second, it is appropriate for inexperienced researchers who are still struggling to settle on a definitive method. Finally, thematic analysis is well known and makes it possible for researchers to pay attention to reality. This thesis was thus decided to use the method of thematic analysis in order to be fully clear and transparent. Braun and Clarke's (2006) rationale is in accordance with what I had in mind and this led to my choosing this research method.

In thematic analysis, it is customary that in-depth analysis be utilized in order to generate the required themes and processes. “Deductive coding is a form of qualitative thematic analysis which involves recording or identifying passages of text or images that are linked by a common theme or idea allowing you to index the text into categories and therefore establish a framework of thematic ideas about it” (Gibbs, 2007). According to Braun and Clarke (2006) there is no specific way to discern what valid themes are needed in data sets. Therefore, the accuracy of the experts’ initial analyses has a decisive impact on the valid themes chosen in the end. To this end, the author selected a number of meaningful final themes through manual data coding for use in subsequent analysis. During this process, it is expected that the researcher should find meaningful connections between the said themes. Finally, the author evaluates the research results with regards to research question.

4.5. Method of data analysis

In this section, the collected data on Agile testing, extracted from interviews with Cybercom testers are analyzed. Using a deductive (bottom up) approach can be helpful in finding themes through the data. This type of analysis assesses the data based on the pre-determined research question. Braun and Clarke (2006) describe six major phases in coding: familiarization with data, generating initial codes, searching for themes among codes, reviewing themes, defining and naming themes, and producing the final report.

1. Familiarizing with data

The aim of this step is to generally know more about the collected data and is very important with regards to getting familiar with the gist of data before commencing the coding process Braun and Clarke (2006). This step in the present study includes transcribing, reading and rereading the data to note down the initial ideas. All the interviews were transcribed and are available in appendix.

2. Searching for themes

The initial themes are derived from the interview questions which, in turn, were conceived on the basis of the research question. The interview questions meanwhile were generated with the help of Ilene Burnstein’s seminal book, Practical Software Testing: A Process-Oriented Approach (2002) in addition to Lisa Crispin and Janet Gregory’s Agile Testing: A Practical Guide for Testers and Agile Teams (2008). Some of the questions, meanwhile, were produced spontaneously as follow-up questions in response to the first answer provided by interviewees during the aforementioned semi-structured interviews. The initial themes were, in the end, were selected as follows: Method of Choice, Performing Agile Tests, Manage the Bugs, Advantages of Testing Tools, Technical debt, Roles of Testers, and so on.

3. Generating initial codes

This step involves collating the codes into potential themes and gathering all the data relevant to each potential theme, coding the interesting features of the data systematically across the entire data set, and collating the data relevant to each code Braun and Clarke (2006). The author here collected codes and categorized key ideas regarding the interview’s topics i.e. themes.

4. Reviewing for themes

In this step, as Braun and Clarke (2006) indicate, in order to ensure that the data extracts have the desired relation with each other and form a coherent whole, the author re-read the data several times. Next, themes that seemed the most relevant with the data were examined. It is essential that all the previous three steps up to this one be considered again so as to produce a suitable thematic map. Only after reviewing the said thematic map could the author move on to step 5

5. Defining and naming themes

At this point, the ongoing analysis to define the specifics of each theme continues and the overall impression that the analysis produces is considered in order to arrive at clear titles and definitions for each theme. Then, by combing the themes which are related to each other and giving the outputs specific names (which pertain to the research subject or research question), conducting one final round of analyses, and finding possible connections with the relevant literature, the final themes are achieved (ibid).

6. Producing the report

As Braun and Clarke (2006) highlight, this step constitutes the last part of the process, deals with the final themes and is concluded by writing the report. The author in this probe, attempted to write a concise yet thorough report that encompassed all the data as part of the task of providing useful evidence for each theme. In this thesis, the final themes extracted from the study of this research were Skills of Agile Testers, Uses of Automated Testing, and types of Scrum-based Agile tests were used in the final report.

It must be noted that these steps are not implemented in the order they are written here. The author decided to choose a deductive approach because it is appropriate for examining the Agile methodology and, thereby, for this thesis.

The first theme refers to the importance of Agile tester' skills. It shows that the ability of a tester affects the accuracy of tests and pinpoints details regarding the importance of Agile tester' skills. For instance, the purposes of Testers' tendency and their Reasons in using Agile methodology and Testers' qualifications in accomplishing their determined roles. The second theme focuses on Various testing tools and the features of automated testing which are specific to each company. Some types of tests need to be performed repeatedly, for which automated testing is considered more suitable. Cybercom is a company where Automated testing is the dominant type and, hence, certain details as to their various applications of Automated testing are provided. The last theme deals with the details of a number of Scrum-based Agile testing methods and their implementation, such as the aim of the test, the type of the test, etc. In terms of the proportion of the answer this section provides to the research question, this theme arguably constitutes the main part.

The initial themes were derived from the interview questions. The extracted themes are displayed in the table below:

Final Themes	Potential Themes	Description
The importance of Agile tester' skills	Testers' qualifications in accomplishing their determined roles	Customer preferences, making notes of customer's feedback
	Testers' tendency and their Reasons in using Agile methodology	Trying to solve issues, Focusing on teamwork
The importance of automated testing usage and its 'Dos and Don'ts'	Various testing tools	What are the advantages and the disadvantages of this kind of testing tool?
	Automated testing tools features	What testing tools are available for using?
Types of Scrum-based Agile tests and their implementation	Various Scrum tests selection	What type of test is implemented in the Agile company?
	The significance of Scrum method	Improving skills, mastering new tools and practices
	Agile tests in an actual performing	How to perform tests in an Agile company?

Table 3: Final themes

4.6. Reliability and Validity

In this section the author explains how the quality of the research is validated. One of the frequent problems in this sort of research is that the researcher's command of data collection and interview techniques inevitably affects the accuracy of the gathered data. Certain problems are likely to occur while conducting interviews. In such situations, the interviewer's experience is crucial in allaying the situation as it not only improves the quality of the questions and interview but also positively affects the interviewees and accuracy of their answers. Another problem may be that testers or developers may not be given enough time or the chance to express their opinion about a given situation they find problematic. In actuality, however, the aforementioned problems did not negatively affect the validity of this thesis.

5. Themes

In this chapter, the themes of the research on important factors for successful Agile testing implementation in the view of Agile testers are presented. The author demonstrates the themes here in order of the sequence in which the topics were discussed in each interview. Therefore, a potential theme comes first, followed by the four professional testers' answers to it, and so on. The interviews were conducted (both face-to-face and on the phone) on March 2018, and the answers below were given by the interviewees: Appendix includes the unedited manuscript of the interviewees' answers.

Some of the initial themes were derived from the answers to interview questions, including the testers' tendency and their reasons in using Agile methodology, the significance of Scrum method, Agile tests in an actual performing, management of bugs, product Validation, Various testing tools, Advantages of Testing Tools, Disadvantages of Testing Tools, and Uses of Acceptance testing. The other initial themes were derived from the follow up questions that only came up during the interviews. These include the Roles of Testers, Uses of Capacity Testing/ Functionality Testing, Uses of TDD/BDD, Technical Debt, and Test Cases. Both interview topics and new ones are elaborated upon in the following section.

5.1. The importance of Agile tester' skills

The theme of *the importance of Agile tester' skills* emerged from the two data-related themes, namely *Testers' qualifications in accomplishing their determined roles* and *Testers' tendency and their Reasons in using Agile methodology* (see table 3). The author found patterns within the data in order to show what the qualifications of a tester are or how Agile methodology can affect the roles of testers.

5.1.1. Testers' qualifications in accomplishing their determined roles

Tester qualification has a vital role in organizations. Two potential themes were roles of testers and management of bugs.

- ***Potential theme: Roles of testers***

This table indicates the roles of testers from the points of view of Cybercom's testers.

Tester1	Stated that testers have various roles: they define the scope and a number of test cases for every node and perform the test cases. Although almost all features are written by the developers, the testers also need to understand the features in order to make better decisions as to what type of test is needed with each of them. Tester 1 made it clear that testers' experience invariably leads to smoother testing processes and quicker detection of bugs and other failure. Agile testers tend to support and help each other and developers. If the testers are experienced enough, finding bugs should not be difficult.
Tester2	Explained that testers try their best to free the products from all bugs and, to achieve this, make use of different types of testing. Testers with enough experience proved able to find bugs more quickly.

Tester3	Stated that testers have an important role in Agile teams. It goes without saying that testers should possess a deep knowledge not only about their own work, but also about the developers' in order to carry out their assignments impeccably. Without understanding scope, test case and features, testers would have a difficult time doing their jobs properly.
Tester4	Said that in Cybercom, developers and testers work closely with each other and when testers find bugs and write failure reports, they discuss the issue with developers and help find ways of solving it. Testers, thus, support the developers even while they are busy performing tests on different projects at the same time. Testers might indeed be professionals with enormous experience and past success, but they still need to perfectly understand every new feature.

Table 4: Roles of testers

- ***Potential theme: Management of bugs***

This table explains the way bugs are managed according to Cybercom's testers.

Tester1	Explained that in case of running into crashes or bugs, testers are to write a minute failure report and send proper feedback to the design team. Testers repeat testing until they are certain the codes are flawless and no bugs have remained. They then deliver the codes to the test manager who, in turn, has a duty to ensure that all the tests have been run properly and no bugs exist anymore.
Tester2	Stated that Scrum is the best tool also to find bugs and to eradicate them. When a bug is detected, members of the Agile team engage in pair programming to solve the problem together as well as to get acquainted with one another's activities, share their knowledge and experiences, and to utilize different types of the Scrum testing.
Tester3	Added that if a problem is found after delivering the product, the customer writes a trouble report and sends it to the Agile team. If the problem relates to testers, they will try to fix it as soon as possible.
Tester4	Revealed that testers have a requirement sheet as well as a report sheet. Therefore, if any problems occur while running the test cases, the testers will inform the development team, and the two teams work together to fix the bugs. It is also paramount that testers should work in peace and always comply with the principles of testers.

Table 5: Management of bugs

5.1.2. Testers' tendency and their Reasons in using Agile methodology

This table contains the opinions of Cybercom's testers as to why Agile methodology is their method of choice.

Tester 1	Stated that Cybercom's choice of Agile software development is due to the fact that it makes communication with other testers, developers, and customers simpler. For instance, the development team can respond to customer feedback more quickly. In an Agile team, everyone must be ready to do anything. As an example, if the design team were busy with a complex task, the test team should be able to join in and help them deal with an urgent extra task. In brief, one of the principles of Agile methodology is that testers and developers should be able to support one another at any given situation.

Tester 2	Reasoned that, given the customers' tendency to change their minds about certain things, an Agile tester should be ready to take note of customer feedback and try go ahead with the new requests. As Agile testers collaborate with colleagues, Agile development may be said to urge teamwork and improve the conditions of group dynamics.
Tester 3	Explained that Cybercom use Agile development because Agile teams are known as able to handle both the development and testing phases efficiently, therefore, there is remarkable cross functionality in Agile teams, which is a huge plus.
Tester 4	Believed that the choice of using Agile methodology is rational as it is very popular and always associated with flexibility and functionality.

Table 6: Testers' tendency and their Reasons in using Agile methodology

5.2. The importance of automated testing usage and its 'Dos and Don'ts'

This theme assessed *the importance of automated testing usage and its ' Dos and Don'ts'* in Cybercom and predetermined two other themes. The themes were *Various testing tools* and *Automated Testing Tool features* (see table 3). The themes were examined in order to determine their impact on the way testing is done.

5.2.1. Automated testing tools features

This theme describes two potential themes such as advantages and disadvantages of testing tools.

- ***Potential theme: Advantages of testing tools***

This table indicates the advantages of Cybercom's preferred testing tools according to the company's testers.

Tester 1	Asserted that one of the main factors that convinced them about choosing JCAT (Java Codon Automation Tool) was that it is very easy to use for write test cases, is flexible and fast, has a higher test coverage, and can usually be used in Java Script.
Tester 2	Added that the main strengths of JCAT are that it is not too centralized and works very fast. This means that testers would not need a lot of time to write test cases.
Tester 3	Said that using automated testing can improve testing speed and test coverage, as well as reduce time and cost.
Tester 4	Explained that what makes JCAT superior to other similar tools is that it can operate on different platforms, has a unique structure, and is suitable for Java script. This software collects all the logs and compares them easily. It also provides better and fast performance in terms of test execution. It run repeatable.

Table 7: Advantages of testing tools

- ***Potential theme: Disadvantages of testing tools***

This table indicates the disadvantages of Cybercom's preferred testing tools according to the company's testers.

Tester 1	Said JCAT takes longer to finish the process and, at times, it is harder to detect the bugs in it.
Tester 2	Said that the chief defect in JCAT is that it is a bit slower than similar tools.
Tester 3	Explained that node testing is the smallest type of test in the system. The tricky part involves configuration which is not easy to understand, especially when all parameters come from TGF. It is also a complicated affair, having to configure the nodes before starting the testing process.
Tester 4	Complained that JCAT is a little complicated, needs to be given many details to function, and is not widely usable.

Table 8: Disadvantages of testing tools

5.2.2. Various testing tools

This table explains how and when various testing tools are used in Cybercom according to the company's testers.

Tester 1	Stated that they use Automated testing and Manual testing based on the requirements of each situation. For automated testing, tester 1 said, they use JCAT (Java Codon Automation Tool) in order to test the software repeatedly. In fact, automated testing increases efficiency and leads to continuous integration in order to ensure all codes are compatible and functional. For best results, Tester 1 believed, manual testing and automated testing must be implemented simultaneously. This could also motivate the testers to use automated testing to perform different types of tests.
Tester 2	Related that they perform tests on the machinery using JCAT in order to prevent the failures and improve the quality of the product. After JCAT tests are done, JCAT displays a flag to inform the user whether the test was successful or not.
Tester 3	Revealed that they use TGF (Transforming Growth Factor) as the Automation tool of choice to carry out Integration and Regression tests. Furthermore, they use JCAT to execute the test, after which the outcome is returned to TGF. Tester engage perform different types of test as automatically.
Tester 4	Explained that They use TGF and JCAT for automated testing. Specifically, TGF is used for Performance test and JCAT for operation test. Also, automated testing is suitable for parallel projects.

Table 9: Various testing tools

5.3. Types of Scrum-based Agile tests and their implementation

The theme *Types of Scrum-based Agile tests and their implementation* consists of *The significance of Scrum method*, *Various Scrum tests selection* and *Agile tests in an actual performing* in general (see table 3). These three components were analyzed in order to determine what scrum tests are implemented and how they are then performed using the Scrum method.

5.3.1. The significance of Scrum method

This theme emphasizes the significance of Scrum method in Agile software development. Scrum is a useful framework whose features and functions have a positive effect on performing various tasks. Two potential themes are extracted in relation with Scrum whose specifications are also evaluated. The potential themes were Method of choice and Technical debt.

- ***Potential theme: Method of choice***

This table explains the methods of choice in Cybercom.

Tester 1	Explained that, Scrum method is suitable for Cybercom's projects because Scrum has short iterations with specific goals. Moreover, before the start of each sprint, there is comprehensive sprint planning in order to estimate the requirements of the procedure, while after the commencement of a sprint, there are daily stand up meetings in order to assess the work done so far and to know how every team member is doing or if there is any need for improvements. The team defines the goal each sprint is meant to achieve. The third step involves the sprint review meeting, which happens at the end of each sprint in order to deliver the product. Finally, the last step i.e. sprint retrospective meeting is held to analyze the finished sprint comprehensively before the next sprint starts.
Tester 2	Scrum is a process which is suitable for parallel projects because many features in projects are interconnected with each other and Scrum is particularly effective in handling such situations. In addition, with this method, there are pair programming efforts which means that developers and testers sit beside each other in order to get acquainted with one another's activities and share their knowledge and experiences.
Tester 3	Was of the opinion that the method of choice depends on the nature of the project at hand, adding that, for instance, they use Kanban when the team needs to correct a failure report and when time is not an issue. On the other hand, when developing new features and predicting what duration of time the feature would need for complete development and/or testing, Scrum is the more suitable method. In brief, Scrum is more suitable when planning is prioritized.
Tester 4	Said they use Scrum because it is effective in identifying the details of each task and features as well as in determining the goal of each sprint, while it helps everyone clearly understand the situation with each assignment and exactly which part of the product they are to deliver at the end of each sprint.

Table 10: Method of choice

- ***Potential theme: Technical debt***

This table explains the task of reaming in each sprint.

Tester 1	If the team misses the deadline in a sprint and the tasks remain incomplete, they inform the PO (product owner) to plan the next sprint in a way that enables the team to cover the unfinished features as well. In other words, incomplete tasks are always moved to the next sprint.
Tester 2	If the team needs to improve the code and perhaps lack sufficient time to do so, they should speak to the PO in order to continue work on the current code in the next sprint.
Tester 3	If the team spend time to fix a faulty code and do not solve the problem in time, work on fixing it should continue during the next sprint.
Tester 4	We use this when the team lacks time or needs to improve some aspects of the product.

Table 11: Technical debt

5.3.2. Various Scrum tests selection

The various Scrum tests selection theme consists of Use of Test-Driven Development (TDD), Use of Behavior Driven Development (BDD), Use of Acceptance Testing, Use of Capacity Testing, and Use of Functionality Testing. The author elaborates on each item separately.

- ***Potential theme: Use of Test-driven development / Behavior driven development***

This table explains why Agile teams use, or do not use, TDD and BDD.

Tester1	Stated that it is the clients who decide whether the team should use TDD or BDD, but added that they mostly do not use them because they, TDD and BDD, decrease code coverage.
Tester2	TDD and BDD have some advantages and disadvantages but the team's using them depends on the requirements of each assignment, adding that using TDD and BDD takes a great deal of experience.
Tester3	Stated that TDD and BDD are not used in Cybercom, but if they were, they would require more experience and depend on what the clients wanted from the company.
Tester 4	Said that using TDD and BDD takes more time and decreases the code coverage. Therefore, clients do not want it from the company.

Table 12: Use of Test-driven development / Behavior driven development

- ***Potential theme: Uses of Acceptance testing***

As all testers asserted, acceptance testing is performed by the client and not in Cybercom. However, even though the testers do not do it themselves, they all sufficiently know it.

This table discusses the implementation of acceptance testing.

Tester 1	Stated that once the test manager delivers the product to the release section, acceptance testing is carried out. For this test, testers examine all the features again in order to ensure everything is running smoothly. But if bugs are found, testers write a failure report and send their feedback to the development team. Both teams then work closely with each other to fix the problem immediately.
Tester 2	Said that testers do use acceptance tests regularly in release section but added that whether to perform them or not basically depends on what the customer wants. If some error finds, they give feedback to us in order to solve it.
Tester 3	Also confirmed that testers use acceptance test right before releasing the product in release section to double-check all the features, but sometimes some issue see in product, they inform us to remove this.
Tester 4	Explained that the release team use acceptance tests in order to ensure no bugs remain in the codes before delivering the final product to the customer. Test manager is responsible to confirm this.

Table 13: Use of Acceptance testing

- ***Potential theme: Uses of Capacity testing/ Functionality testing***

This table explains the two types of tests performed in Cybercom.

Tester 1	Functionality test: Used for normal traffic, and to assess how the software has been loaded into the hardware. Capacity test: Related to the behavior of new features in the nodes. For instance, the answer to such questions as 'how many users can connect to these nodes?' are determined here
Tester 2	We use capacity test when upload software to hardware in order to know about speed. Functionality test check the behavior of code functionality.
Tester 3	Both of them are useful for product, we use both of them in order to improve and examines product.
Tester 4	Testers ask different questions and test different elements in each category of testing. For instance, in capacity test, testers are responsible for measuring the upload speed, downloading different parameters, and checking the capacity and suitability of the product; for example: How many cell phones are there? To what extent have the number of cell phones increased? and How many users connect to the product's baseband? In functional test, testers examine the following factors: Functional area support, Support of test cases, Code functionality.

Table 14: Uses of Capacity test/ Functionality test

5.3.3. Agile tests in an actual performing

This theme deals with different types of tests in Cybercom, such as Performing tests, Product Validation and Test Cases implemented in Cybercom. It was analyzed in order to see what tests and test cases are implemented and what problems are most likely to occur during validation. This section could also help find part of the answer to the research question.

- ***Potential theme: Performing tests***

This table discusses the implementation of Agile testing in Cybercom.

Tester 1	Stated that in Cybercom, designers design the codes and write the unit tests, but they do not perform the test cases and leave that to the test team. Testers perform many types of tests on all phases in order to improve the quality of testing process and detecting testing failures. Finally, if the design team is busy, the test team helps designers in performing their tasks. Note: Unit test is the only test run by the developers of Scrum team, while Functionality testing, performance testing, capacity testing, and system testing (based on Exploratory testing) are implemented by testers. Unit testing: Developers still do not run a complete test here. In practice, they test some features and in case of problems, alter them accordingly. System testing: Examining all the hardware to understand based on Exploratory testing e.g. how differently the product performs in situation A compared to situation B.

Tester 2	Related that Cybercom has established two design teams, named Santinear and Spektera, with each team including two professional testers. One team is given the task of conducting performance testing, Regression testing, Capacity testing, and Functional testing, while the other is in charge of Integration test, and System test implemented based on Exploratory testing and Session-based testing. Finally, Acceptance testing is conducted at the release section. Numerous tests are also conducted on base band nodes as base band is arguably the core of telecommunication. While the developers have the clear task of writing the codes, testers take on more than one simple task. First of all, as mentioned earlier, before the developers deliver the codes to the test team, testers should determine the scope and test cases. When the codes are ready, developers first run unit test themselves, and if everything goes well, they deliver the code to the test team. The two test teams work in parallel and begin to run tests to detect potential issues.
Tester 3	Explained that there are two teams and each team have two testers. One Agile tester carries out tests and assesses the quality of the work in progress while the developers write the codes and send them to the tester. Testers test codes for bugs and crashes, write trouble reports if needed, and send the codes back to developers for modification. Testers configure the nodes and test the codes for final approval, as well. Testers change specific parameters in base band nodes, change the system constant, and perform some tests manually. Integration test, aims to make sure that different base band codes work together without problems, while Regression test is done in order to not lose the legacy code when something new is added, continuous integration reduce time regression test and release test is done by both teams separately. Exploratory test and Session-based test are performed based on the outcome of System test and Integration test. Implementing exploratory test is also a very good solution. Test manager is responsible over the health and functionality of every code that goes through the test team.
Tester 4	stated that, in Cybercom, writing codes, developers in Cybercom also write basic unit test and extended base test (EBT) which are used in order to observe the functionality changes in the current build, then upgrade the codes, and finally transfer and implement the code to the master software which, again, will be given to testers on which to conduct one final round of tests. Integration test, Exploratory test, Performance test, Capacity test, Regression test, Session-based test and Functionality test implement in Cybercom with test teams.

Table 15: Performing tests

- ***Potential theme: Product validate***

This table elaborates upon problems regarding product validation from the points of views of the Cybercom's testers.

Tester 1	Stated that the main problem in terms of validation is memory shortage, which the developer will then try to solve. The other problem is related to crashes. Crashes are bound to happen once in a while in real networks, tester 1 asserted, and there is not much testers can do to prevent them.
Tester 2	Related that 50 percent of the available hardware is not good enough, and to make matters worse, hardware is very expensive in Sweden. Hence, hardware quality has significant impact on product validation.

Tester 3	Explained that almost all of the problems occur during the testing phase, therefore, they do not run into many problems regarding product validation. Otherwise, developers attempt to solve it.
Tester 4	Also insisted that they hardly find any specific problems in the validation phase because the extensive tests carried out on test cases make sure no bugs remain in the codes.

Table 16: Product validate

- ***Potential theme: Test cases***

This table explains how test cases are performed in Cybercom.

Tester1	Tester should know about test cases, and when the developers write the code, we read the test cases, most of them written by clients.
Tester2	Agile teams perform different test cases, all of which ordered by the clients.
Tester 3	Testers in Cybercom should be aware of the proverbial big picture of what they do. They have to verify ten test cases that may be carried out automatically. After the Ping test case, Uplink test case, Downlink test case, signaling test case, UL/DL test cases, etc. are also conducted successfully, the final codes are delivered to the test manager.
Tester4	Clients give the team ten test cases. When the developers write the code, testers should inspect them in order to find the bugs.

Table 17: Test cases

6. Analysis

This section includes the analysis of the results obtained from the interviews with Cybercom's testers. In order to provide a clear report, I start this chapter with an analysis of the results 'The importance of Agile tester' skills', then move on to the analysis of the 'The importance of automated testing usage and its 'Dos and Don'ts' and finish with the analysis of 'Types of Scrum-based Agile tests and their implementation.'

6.1. The importance of Agile tester' skills

The finding in *the importance of Agile tester' skills* is analyzed and consists of *Testers' qualifications in accomplishing their determined roles* and *Testers' tendency and their Reasons in using Agile methodology* that emerged from the data set. This diversity has effect for implementing Agile in large companies that intended to change the methodology.

6.1.1. Testers' qualifications in accomplishing their determined roles

Two of the findings included the Roles of Testers and Management of Bugs, which combined to generate the more comprehensive theme, Testers' qualifications in accomplishing their determined roles. This refers to the tester's vital role when coming across bugs in each sprint and how the tester should be able to properly deal with them. The answers of Agile testers are similar to each other for both themes.

In Cybercom, Agile testers have different duties, such as defining the scope and doing test cases for every node and perform test cases. They believed that Tester need to understand the feature what it is and what need to define and what testing they need to with that feature. In addition, Agile testers have enough experience in order to find bugs more easily and quickly. All Agile testers in this company have a deep understanding of their work in order to keep the knowledge organized. For instance:

Tester 3: "[..] Testers have an important role in the test team, therefore, we should have deep knowledge and enough experience until pass the assignment successfully ".

Agile testers in Cybercom believe that in ordering managing the bugs, when the testers stumble upon bugs or crashes, they immediately write a failure report and give their feedback to the developers in order to solve the bugs. In addition, the team members work in a friendly environment and carry out several undertakings in pairs. It follows then, that developers and testers in this company can often be found helping each other as they attempt to fix bugs. For instance:

Tester 4: "[...], we run test case if pass it ok but if crash happened or bug happened or functionality broken tester write report for developers and developers and testers work each other and work friendly, we should have a pace and respect to the principles of Agile development and tester ".

6.1.2. Testers' tendency and their Reasons in using Agile methodology

Four Testers in Cybercom stated that Agile development is popular and more flexible because, given the customers' tendency to often change their requests, the testers take account of customer feedback and try to move forward with the new agenda. For instance:

Tester 2: "[...] Tester should listen to feedback from the customer and the colleagues, work with the colleagues friendly in direction of the same goals, same idea. Team prioritize what we should do so I think the process is good and always try to listen to the feedback, try to move forward and improve the group's dynamics."

One of the principles of Agile manifesto states: "***Individuals and interactions*** *over processes and tools*" (Martin, 2003). This was confirmed by the ideas of the testers in Cybercom. They expressed that use of Agile methodology improve communication between the developers, testers and customers, therefore, Team can respond easily to customers feedback and can handle both developing and testing. For instance:

Tester 1: "[...] Cybercom choice of Agile software development is that it makes communication with other testers, developers, and customers simpler; for instance, the development team can respond to customer feedback more quickly."

6.2. The importance of automated testing usage and its 'Dos and Don'ts'

The findings in the section of *the importance of automated testing usage and its 'Dos and Don'ts'* were obtained from the data sets and analyzed. They resulted in the two themes *Automated testing tool features* and *Various testing tools*. This theme could positively affect the quality of companies' products by effective implementation of Agile tests.

6.2.1. Automated testing tools features

There were two findings in this theme including the advantages and disadvantages of testing tools, after asking the opinions of testers in Cybercom regarding these themes, it became clear to the researcher that the advantages of using automated tools outweigh its disadvantages. Tellingly, I did not get any contradicting answers from the four testers.

The interviewed Agile testers in Cybercom stated that Automated testing tools are quite effective and helpful, according to Agile test teams. However, whether to use them or not depends on a number of factors. First of all, teams should know the purpose of the current test. There are certain types of projects which require Automated testing because manual testing takes a great deal of time and also because customers are known to change their minds often. To deal with the latter, for instance, JCAT, a widely used Automation tool, is a flexible, unique, and very fast application that, fortunately, is not very a centralized and could be implemented on different platforms. Furthermore, JCAT meaningfully reduces the implementation time of tests. For instance:

Tester 4: "[...] What makes JCAT superior to other similar tools is that it can operate on different platforms, has a unique structure, and is suitable for Java script. This software collects all the logs and compares them easily. It also provides better performance in terms of test execution."

Agile testers in Cybercom believe that Testing Tools do have some negative points as well. For instance, JCAT takes more time to finish, is a little complex and does not help much in finding bugs, is a little slower than similar applications, and requires a great number of details as input. It is also tricky when it comes to configuring nodes before starting the test. For instance:

Tester 3: "[...] configuration which is not easy to understand, especially when all parameters come from TGF. It is also a complicated affair, having to configure the nodes before starting the

testing process."

6.2.2. Various testing tools

Cybercom testers use JCAT and TGF because it is based on JavaScript. There are also times when it is the customers who determine what tool or programming language they would like to see the company use to develop the product they have ordered. On the other hand, a decision has to make as to whether it is more suitable to go with manual testing or automated testing. Normally, Cybercom testers revealed, the team execute the test with JCAT, and send the outcome of the test to TGF. For instance:

Tester 4: "We use TGF and JCAT for automated testing. Specifically, TGF is used for Performance testing and JCAT for Operation testing."

6.3. Types of Scrum-based Agile tests and their implementation

The findings of the theme *Types of Scrum-based Agile Tests and their implementation* were obtained from the data sets and analyzed accordingly. They include *the significance of Scrum method*, *Various Scrum tests selection*, and *Agile tests in an actual performing*. This theme is characterized by a great deal of variety and examines what types of tests are performed in order to address and resolve all the problems observed during a sprint. However, each Agile company prefers to use the types of tests that they deem more effective in order to enhance their implementation quality and detect bugs more easily and quickly.

6.3.1. The significance of Scrum method

There were two findings in this theme such as Method of choice and Technical debt which were developed Scrum method. It means that in Scrum method everything has planned and can handle project in good direction. I did not see any contracting answers between these themes and related the technical debt is emerged between answer of one of the testers.

The four interviewed testers believed that Scrum is a process with short sprints and thus suitable for parallel projects. One of the testers also stated that scrum is good for planning. This seems true enough as one of the characteristics of Scrum is that the requirements are clear at the beginning of each sprint and fixing the bugs occurs during sprints. If the discovered bugs are not fixed, the team may run out of time and miss the deadline. In this case, the test team should inform the product owner and request that the current part be implemented during the next sprint.

The testers believed that Scrum has the ability to support pair programming and knowledge sharing. The Team is also required to be self-organizing and work cross-functionally. Based on these reasons, Cybercom has chosen to use Scrum. For instance:

Tester 2: "[...] Many projects are interconnected with each other, or at least some of their features are related to each other. Actually, we have an effective way of handling these connections through Scrum which consists of short iterations to reach our determined goals. We engage in pair programming and share our knowledge with one another."

6.3.2. Various Scrum tests selection

There were three findings in this theme which help understand what types of scrum tests

are performed in Cybercom. The answers of the four Cybercom testers were similar to each other, although tester 3 provided more information with regards to the type of tests used in the company.

According to the interviewed Cybercom testers, one team runs *Integration testing* and *Regression testing*, while the other team carries out *Functional testing, Capacity testing, System testing, Exploratory testing, Session-based testing* and *Performance testing.* Should any bugs or other issues be found during the testing process, as is often the case with continuous integration testing, testers write a failure report for the development team based on which efforts are made to try and solve the issues. This process continues until the test team is absolutely convinced the product is free from bugs. At this point, the test manager receives the product from the tests team and checks everything once again. If the test manager gives the all-clear for product release, the release team runs *Acceptance test* in order to verify its quality and deliver it to the customer. It must be mentioned that, in an Agile development team, because the development team works in tune with the test team, developers could never get ahead of testers: in Agile methodology, no feature is ever considered finished until it has been completely tested.

6.3.3. Agile tests in an actual performing

This section deals with three findings: performing the tests, implementation of different test cases, and product validation. With regard to the theme of product validation, the interviewees expressed different opinions. Two of the interviewed testers stated that problems do occur in test validation, but the other two disagreed and insisted no further problems are seen after all the tests are carried out.

According to Agile testers at the company, implementing tests in Cybercom is done as follows: after the sprint planning, the requirements are determined, and the developers begin to design the codes and also write basic unit tests in order to make sure no bugs remain in the system. The developers then pass the codes in full to the test team. Two test teams, each consisting of two testers, work in parallel. In Cybercom run, tests are performed both on the hardware and software, however, the company admits that more focus is allocated to software testing.

Testers in Cybercom has a one idea that they should pay attention to different types of test cases and these test cases perform automatically and continues integration happened in order to reduce the time of implement test, leads to go the project to forward.

Product validation is an essential step for a test team. Some of the interviewed testers asserted that when the team tests everything properly, they almost never face any problems when validation is in progress.

Tester 4: "[...] the extensive tests carried out on test cases ensure no bugs remain in the codes."

However, there were testers who disagreed with this revelation.

Tester 1: "[…] The other problem is related to crashes. Crashes are bound to happen once in a while in real networks and there is not much testers can do to prevent them."

7. Discussion

Despite the small data set from the Semi-structured interview, there is interesting diversity in results. After the interviews with the Agile testers three central themes emerged that indicated the main factors that influence the Agile testing implementation in the view of Agile testers in Scrum framework including *(1) The importance of Agile testers' skills, (2) The importance of automated testing usage and its 'Dos and Don'ts' and (3) Types of Scrum-based Agile tests and their implementation.*

The importance of Agile testers' skills, include details involving Testers' qualifications in accomplishing their determined roles in the right time as well as Testers' tendency and their Reasons in using Agile methodology. Within this theme, the importance of testers proficiency were emphasized throughout the interviews, with the interviewed testers pointing to testers' skill in performing Agile tests and in removal of bugs as crucial components in successful and; accurate testing. These findings indicate a remarkable focus on the attitude of testers. It seems that the questions of skills an underrated and under discussed matter despite its rather vital role in the improvement of the team's performance in general. Based on what was discussed in the second chapter named background of Agile testing application and as per Crispin and Gregory's (2009) assertions, Agile testers are always on the lookout to acquire any new skills, tools, and practices that could potentially help them gain a better understanding of the customers' requirements among other things, and consider it an unbending rule to never forget that "*Without the attitude, the skill is nothing*" (Crispin & Gregory, 2009). Considering the importance of Agile testers, the findings of this research is in line with the findings of Puleio (2006) and Nidagundi and Novickis (2017) that assert that expert Agile testers in Scrum teams increase the cross-functionality of the Agile team.

The second theme that emerged from data analysis is *the importance of automated testing usage and its 'Dos and Don'ts*'. An important point to consider in this category was the features of Automated Testing tools. This finding covered both positive and negative features. Moreover, positive features of automated testing tools are outweighed by negative ones. As Berlowski et al. (2016) assert that some of the advantages of using automated testing include higher test execution speed, higher efficiency, reduced time and costs, and eliminating the need for human intervention. Another interesting point in this theme was related to testing tools, referring only to a few targets set by Cybercom. This might mean that testers focus on other ventures as well. In chapter two (background of Agile testing application), Crispin and Gregory (2009) underline the importance of implementing automated tests in Agile testing quadrants. Quadrants 1, 2 and 4 provide quick access to the information and enable the team to carry out troubleshooting equally quickly by performing automated testing and run it multiple times to ensure no further complications remain. In quadrant 1, tests are performed manually and therefore need human control. In addition, it seems there is no competition when it comes to performing the tests. In short, the findings of the present study support Vishvadeep Tripathi and Arivnd Kumar Goyal (2014), Jakobsen and Johnson (2008), Elallaoui et al. (2016), Berlowski et al. (2016) , Butgereit (2018), Manu GV, Namratha M, and Pradeep (2015) that strongly tend to use Automation testing.

The third theme that is the most important finding of the study, if one could put it that way,

were within *the Types of Scrum-based Agile tests and their implementation* , which consists of details about implementing Agile tests in Scrum teams from the view of tester. The findings, as a whole, imply that Agile tests may be used to accomplish a variety of goals, and that the tests implemented in Cybercom (such as Unit Testing, Exploratory Testing, more.) and their functions are many and quite varied. For instance, implementing Session-based Testing and Exploratory Testing in tandem as a cohesive testing solution has been proven to help improve the quality of testing process in Cybercom. Findings of the present study is in line with the findings of Vishvadeep Tripathi and Arivnd Kumar Goyal (2014) and in contrast with Sumanth Yenduri and L.A. Perkins (2006) that indicated how using test-driven development approach enhances the quality of developed software and its productivity. Unlike Sumanth Yenduri and L.A. Perkins (2006), Testers in Cybercom claim that these type of test need more knowledge. Furthermore, these tests are time consuming and they do not tend to use these types of test in their projects.

Crispin and Gregory (2009) state that organizations tend to make use of all the aforementioned quadrants (introduced in this study) in performing their tasks. Moreover, because Cybercom's experts are not completely certain about all the requirements at the beginning of the process and implement tests on drive coding, these quadrants could help the teams devise a precise testing plan and make sure that all requirements are met at the end. It must be mentioned again that the findings of this particular research are only based on the practices promoted by Cybercom Group, and it is clear that other companies might perform Agile tests in quite different ways.

When gathering the data through interviews with the testers of Cybercom's two Agile teams, the pre- conceived questions were filtered based on the research topic and, in the end; all four Agile testers in the teams were interviewed. Unfortunately, testers and developers of other Agile companies declined to share their opinions and experiences in a formal context. Thus, the number of interviewees and thereby the number of topics covered by this study remain somewhat limited. On another note, because finding other interviewees, conducting interviews, and performing in-depth analysis on the findings within the determined timeframe were massively time-consuming tasks (even with the current, admittedly limited data), the author ultimately decided to stop the efforts to expand the research scope. However, it should also be said that having small data to work with, in retrospect, is not without certain advantages; chief among said advantages is the fact that the interviews could be performed less stressfully and more comprehensively (in spite of the accent of two testers that was a little difficult for me), while the results could be analyzed with extra care, patience, and accuracy, likely leading to more reliable results in general.

The themes, in any event, demonstrate that the four interviewed Agile testers put the most emphasis on common themes, namely, the Roles of Testers, Testers' tendency and their reasons in using Agile methodology, Various testing Tools, and Agile test in actual performing. It seems important to remember that Agile testing, this core element of Agile methodology Crispin and Gregory (2009) is where the methodology truly evinces its advantages and distances itself from the competition, with its greatest strengths being the insistence on adaptive planning and the capacity for constant improvement and elimination of any and all defects (Martin, 2003).

Adaptive planning is a crucial component of Agile methodology and, especially, is given particular emphasis in Scrum. Adaptive planning is what sets Agile methodology apart and gives it its competitive edge. According to Sawhney, "*Scrum and other Agile methods recognize…that software development is evolutionary and creative*" (2010). This mode of planning recognizes the need for devising a plan as per usual, but, as Cohn points out, it also acknowledges that once work starts the plan is bound the change and, thus, the plan needs to reflect the new knowledge generated by the change. For adaptive planning to be effective teams must be able to work off a prioritized backlog, have close collaboration with the customer to monitor and track the work in progress in real-time, deliver a working, tested increment at the end of every iteration, understand that the direction will inevitably change and the team is expected to respond accordingly, and finally, maintain consistent staffing (2006).

Despite the overall positive impact of Agile methodology and the extensive literature on its advantages, it would be unrealistic and simplistic to claim that Agile is all benefit and no harm—or cost. There is extensive research within the literature as to what are mooted as the limitations and drawbacks of the methodology or otherwise the difficulties it may cause. Turk et al. (2005) mention a number of limitations in Agile processes which, although evidently immitigable, are disadvantages nonetheless. These include: limited support for distributed development environments, limited support for subcontracting, limited support for development involving large teams, limited support for building reusable artifacts, limited support for developing safety-critical software, and limited support for developing large and complex software. Fridman (2016) also cites a number of practical drawbacks for the methodology, by attributing less predictability to it, its need for more time and commitment, greater demand from developers and clients, its lack of necessary documentation, and the possibility that projects could easily fall off-track if proper measures are not taken. The cited disadvantages dictate that it is essential that developers or organizations weigh up their options i.e. other methods before making a choice with regard to their development methodology, as no methodology is perfect and each may be suitable to a specific type of project.

8. Conclusion

In this section, a summary of the findings of this thesis regarding the important factors in the Agile testers' perspective for the successful Agile testing implementation in Cybercom is presented. A number of suggestions that may improve the quality of Agile testing in the future projects are also included.

This study presents a detailed description of the Agile testing methodology and explains about important factors for successful Agile testing implementation in the view of Agile testers in Cybercom as a company that uses the Agile methodology with Scrum framework. This is in keeping with the author's predefined objective of the research, which was to provide a list of suggestions regarding Agile testing tend to increase the overall efficiency in the future projects, and perhaps even in the projects that are currently in progress. The way to accomplish this objective was conceived in the form of an investigation of these factors in a real-world company which prefers and utilizes an Agile framework as its method of choice on a daily basis. In the chapter on literature review, articles related to some aspects of Agile testing and Scrum team were examined in order to provide some necessary background information about requirements of Agile testing in Scrum teams and current trends in this methodology. The most recent relevant research works, which are in general in favor of Agile testing, were introduced.

This research, in fact, was of the explorative type, had a real-world organization as case study, and made use of thematic analysis. Data collection was done through semi-structured interviewed with four expert testers in Cybercom Group. The interviews were semi-structured in order to provide more flexibility and obtain more information about the processes associated with Agile testing. Moreover, deductive coding was performed because we are already familiar with the concept of Agile testing and thematic analysis, hence the author could closely monitor the analysis process and find a reliable answer to the research question. In the analysis section, three categories were presented, each of which contained a number of themes.

8.1. Answer to the research question

Following thorough data collection and investigating the implementation methods in Cybercom, the research question "*what factors are of paramount importance in the view of Agile testers for successful Agile testing implementation in Scrum framework?" m*ay be answered briefly, in three parts, as follows:

The first part of the answer to the research question is that *the importance of Agile testers' skills* should be utilized in a way that best helps prevent test failure; something which, in turn, motivates Agile testers to do even better in future undertakings. Agile testers are expected to always be ready to perform tests and find bugs rapidly. One interesting idea that came up during the interviews involved cross functionality and pair programming, which also serve to increase the motivation and effectuality of Agile teams. In addition, skilled testers with more experience may prove invaluable in improving the quality of testing processes and preventing testing failures. Though testers' main responsibility is performing tests, they should also have an appropriate

attitude in order to establish better communication with other team members. In addition, they should not shy away from sharing their knowledge in team efforts, and preferably work in pairs, as such activities may only lead to increased competence within the team (Crispin and Gregory, 2009). It is essential to note that the developers of Agile development teams must have sufficient knowledge as to the principles of Agile testing, in the same way as expert testers who are expected to have a complete understanding of development practices (Crispin and Gregory, 2009).

The second part of the answer is that it is imperative to investigate *the importance of automated testing usage and its 'Dos and Don'ts'* for most scrum tests and to make efforts to implement automated testing from the beginning of any project. Automated testing should be implemented by Agile testers in Agile development teams, while continuous integration should be performed regularly in order to make sure the new code does not disrupt the functionality of the old code. This leads to reduction in the required time for regression test (Schwaber, 2004). Indeed, automated testing does not eliminate the need for manual testing; the latter is useful when there are no changes in test cases or features. Automated testing and manual testing go hand in hand, as it were, until such a time as the assignment is completed successfully (Schwaber, 2004).

The last part of the answer is related to *types of Scrum-based Agile tests and their implementation*. With the use of Scrum testing, for instance, Exploratory testing and Session-based testing have proven to be effective techniques that could help improve the quality of System testing and Integration testing performed by the test team, and to make more informed, assured decisions regarding the implementation of Agile tests (QASymphony 2018). There are, in addition, several types of Agile tests in order to eliminate test failure and enhance the quality of products as well as customer satisfaction. Exploratory testing, for instance, is a type of test designed to improve testing practices, thus reducing the implementation time of tests and therefore delivering the product in timely manner (QASymphony 2018). The findings of this research are the result of meticulous empirical work and purposeful literature review.

8.2. Suggestion

Although this is a relatively young field in software development and the majority of developers are, as expected, young as well, the research findings compel this author to firmly recommend to companies active in Agile software development to look for skilled testers with more experience when hiring their staff. Experienced testers often prove invaluable in delicate situation by improving the quality of testing processes and preventing testing failures (Crispin and Gregory, 2009). In addition, such seasoned technicians tend to voice their opinion more openly, invariably expressing what they believe may be the most efficient solution to maintain or improve the team's testing capabilities. Importantly, experienced testers do not forget to keep up with the pace of the latest advancements in the field, too. They know how to form an effective working relationship – including effective communication – with the development team in order to help the organization deliver on all fronts, especially in terms of product quality and customer satisfaction (Crispin and Gregory, 2009).

Based on what Cybercom testers had to say, Automated testing should be implemented more resolutely in order to prevent – or better yet, to eliminate – failure in Agile teams. The process

could start from the beginning of the sprint and, most tangibly, reduces test implementation time. This researcher suggests that in addition to allowing several repeats of each test and improves the quality of products to meet customer expectations, Automated testing also increases the motivation of testers and causes them to escalate their efforts in reaching a certain objective, give their utmost, and ultimately be satisfied with their achievement (Schwaber, 2004).

There are other prevalent testing solutions in Agile companies that could also be implemented to prevent failure in systems, Exploratory test, Performance test, Capacity test, Regression test, Session-based test and Functionality test are a few of such solutions. In addition to having the aforementioned types of tests, which help improve the quality of System Test and Integration Test, this author also suggests that the peace of mind that testers feel after a successful implementation of an Agile test, which is always followed by product delivery, is also beneficial from a psychological standpoint. According to QASymphony.com (2018), such a sense of reassurance enhances the testers' motivation and self-confidence.

8.3. Experience of research

During the interviews, the author faced a number of challenges. The heavy accent with which two of the interviewed testers spoke English made what they said quite difficult to understand; therefore, a great deal of time was spent on clarifying what was being said so as to make sure everything was recorded accurately. The second challenge was related to the answers regarding product validation which were frankly confusing and, at times, contradictory. However, with the help of the interviewees, the answers were double-checked through a time-consuming process and the contradictions were ultimately resolved. The third issue involved a noticeable lack of information concerning the details of Acceptance testing. Had more data been supplied by the testers on Acceptance Testing, a more accurate data analysis could have been included in this research. Finally, it was quite difficult to stop the interviewees' frequent digressions; they spoke about new subjects which were mostly unrelated to the author's questions. Had these challenges not surfaced that often, as they did, it is the author's unshakeable conviction that a much better final outcome would have resulted.

8.4. Further research

In this section, the author underlines areas upon which further research may be needed in order to build on and complement the findings of the present study.

In order to prevent duplicate research in the future, a possible solution may be to choose a subject out of the range of this study. The readers are also advised to come up with new ideas even though they may go through a similar path and use similar methods on their way to achieving their objective (Ford and Mohapatra, 2011). Given that this study focuses on the Swedish company Cybercom, this author believes it would be a good idea for future research work to find companies active in the same field in Sweden and find out how they, the other companies, perform Agile testing and what different methods they employ so as to form a clearer understanding of the subject and, indeed, to come up with ways of potentially improving the existing methods and practices. On the other hand, given the fact that some companies still prefer to work with the traditional

Waterfall method, it would certainly be interesting to inquire as to what makes them continue to rely on Waterfall, and what developments, if any, could cause them to change their mind. Future research, furthermore, could focus on other, even starkly different software development methods and platforms. An interesting experiment in this regard may be to set out the same objective for two or more groups of software developers and assign each group a different software development method, so as to be able to compare the final outcome in terms of time, cost, precision, etc.

As may have been noticed, the findings of this research demonstrate a sort of expandability found in many fresh approaches and could lead to future research with the aim of making fresh advances across this relatively young field. This author certainly sees room for improvement and progress here, as there are often ways to transform the current methods of Agile testing in Scrum or other platforms toward faster, more efficient ones. This paper, it is hoped, may be regarded as a foundation on which to build future research work.

References:

Bartlett, J. (2018). *Software Testing Life Cycle Phases (STLC) – Test-Lodge Blog*. [online] Test-Lodge Blog. Available at: https://blog.testlodge.com/software-testing-life-cycle [Accessed 1 May 2018].

Berlowski, J., Chrusciel, P., Kasprzyk, M., Konaniec, I. and Jureczko, M. (2016). *Highly Automated Agile Testing Process: An Industrial Case Study*. E-INFORMATICA SOFTWARE ENGINEERING JOURNAL, [online] 10 (1), pp.69-87. Available at: http:// webofknowledge.com [Accessed 3 May 2018].

Bider, I. and Söderberg, O., 2017. *Moving towards Agility in an Ordered Fashion*. *Springer*, [online] Available at: <https://link.springer.com/chapter/10.1007/978-3-319-62386-3_9> [Accessed 6 April 2018].

Braun, V., and Clarke, V. (2006). *Using thematic analysis in psychology*. Qualitative Research in Psychology, 3, 77-101.

Burnstein, Ilene. (2002). Practical Software Testing Process Oriented approach. New York: Springer, pp. 26-29.

Butgereit, L. (2018). *Using Machine Learning To Prioritize Automated Testing In An Agile Environment*. [ebook] Conference on Information Communications Technology and Society (ICTAS). Available at: https://ieeexplore.ieee.org/abstract/document/8703639 [Accessed 4 May 2018].

Carrera, Á., Iglesias, C. and Garijo, M. (2013*). Beast methodology: An Agile testing methodology for multi-agent systems based on behavior driven development.* Information Systems Frontiers, 16 (2), pp.169-182.

Cohn, M. (2006). *Agile estimating and planning*. Upper Saddle River, NJ: Prentice Hall PTR.

Collins, E., Dias-Neto, A. and de Lucena Jr., V., 2012. *Strategies for Agile Software Testing Automation: An Industrial Experience*. *IEEE Conference Publication*, [online] Available at: <https://ieeexplore.ieee.org/document/8421968/> [Accessed 6 March 2018].

Crispin, L. (2006). *Driving Software Quality: How Test-Driven Development Impacts Software Quality*. IEEE Software, [online] 23 (6), pp.70-71. Available at: http:// webofknowledge.com [Accessed 3 May 2018].

Crispin, L. and Gregory, J. (2009). *Agile testing.* 1st ed. Upper Saddle River, NJ: Addison-Wesley.

Cybercomgroup. (2018). *About the Group*. [online] Available at: https://www.cybercom.com [Accessed 19 Apr. 2018].

Daly, J., Kellehear, A., & Gliksman, M. (1997). *The public health researcher: A methodological approach*. Melbourne, Australia: Oxford University Press.

Elallaoui, M., Nafil, K., Touahni, R. and Messoussi, R. (2016). *Automated Model Driven Testing Using AndroMDA and UML2 Testing Profile in Scrum Process*. Procedia Computer Science, [online] 83, pp.221-228. Available at: http://webofknowledge.com [Accessed 3 May 2018].

Ford, E. and S. Mohapatra (2011). *Idea de-duplication in an innovation community*. Urbana, 51, p. 61801-2302.

Fridman, A. (2016) *The Massive Downside of Agile Software Development*. [online] available at: https://www.inc.com/adam-fridman/the-massive-downside-of-agile-software-development.html. [Accessed 05 Sep. 2018].

Ganesh, N. and Thangasamy, S., 2012. *New Agile Testing Modes*. *Information Technology Journal*, 11(6), pp.707-712.

Gibbs, G. (2007). *Analyzing Qualitative Data*. London: SAGE.

Hellmann, T., Sharma, A., Ferreira,, J. and Maurer, F., 2012. *Agile Testing: Past, Present, and Future*. Charting a Systematic Map of Testing in Agile Software Development [Accessed 2 Apr. 2018].

Highsmith, J. (2001). *History: The Agile Manifesto*. [online] Agilemanifesto.org. Available at:

http://Agilemanifesto.org/history.html [Accessed 19 Apr. 2018].

Jakobsen, C. R. and Johnson, K. A. (2008). *Mature Agile with a twist of CMMI*. AGILE 2008 Conference, [online] pp.212-217. Available at: http:// webofknowledge.com [Accessed 4 May 2018].

Jeeva Padmini, K., Kankanamge, P., Dilum Bandara, H. and Perera, G., 2018. *Challenges Faced By Agile Testers: A Case Study* - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/8421968/> [Accessed 22 March 2018].

Kaner, C. (2008*). Defining Exploratory Testing*. [online] CAST Instructors' Tutorial at CAST in Toronto. Available at: http://kaner.com [Accessed 19 Apr. 2018].

Martin, R. (2003). *Agile software development principles, patterns, and practices*. Upper Saddle River, NJ: Prentice Hall.

Martin, A., Sillitti, A. and Wang, X., (2010). *Agile Processes In Software Engineering And Extreme Programming:* 11Th International Conference, XP [Online]. Trondheim, Norway, June 1-4, 2010, Proceedings. Springer Link. Availableat: http:// webofknowledge.com [Accessed 19 Sep 2018].

Ming Huo, J., Verner, V., Liming, Z. and Babar, M.A. (2004) *Software quality and Agile methods*. IEEE Xplore DigitalLibrary [Online].Availableat: http:// webofknowledge.com [Accessed 15 March 2018].

Namratha, M., Manu, G. and Pradeep, 2015. *Benefits of Test Automation for Agile Testing*. International Journal of Science and Research, 4(2319-7064) [Accessed 3 May 2018].

Nidagundi, P. and Novickis, L. (2017). *Introducing Lean Canvas Model Adaptation in the Scrum Software Testing*. ICTE Conference, [online] 104, pp.97-103. Available at: http:// webofknowledge.com [Accessed 3 May 2018].

Oates, B.J. (2006). *Researching information systems and computing*. London, SAGE

Pugh, K. (2011). *Lean-Agile acceptance test-driven development*. Boston, Mass.: Addison-Wesley.

Puleio, M. (2006). *How not to do Agile testing*. AGILE 2006 Conference, [online] pp.305-311. Available at: http://webofknowledge.com.ezproxy.its.uu.se/ [Accessed 3 May 2018].

Qasymphony.com (2018). *Agile Methodology: The Complete Guide to Understanding Agile Testing*. [online] Available at: https://www.qasymphony.com/blog/Agile-methodology-guide-Agile-testing/ [Accessed 19 Apr. 2018].

Saunders, M., Lewis, P. and Thornhill, A. (2003). *Research methods for business students*. 3rd ed. England: Pitman Publishing.

Sawhney, R. (2010) *How to Sustain Adaptive Planning*. [online] available at: https://www.scrumalliance.org/community/articles/2010/february/how-to-sustain-adaptive-planning. [Accessed 05 Sep. 2018].

Schwaber, K. (2004). *Agile project management with Scrum.* Redmond, Wash.: Microsoft Press.

Software Testing Fundamentals. (2018). *Agile Testing - Software Testing Fundamentals*. [online] Available at: http://softwaretestingfundamentals.com/Agile-testing/ [Accessed 28 Apr. 2018].

Tripathi, V. and Kumar Goyal, A., 2014. *Agile Testing Challenges and Critical success factors.* International Journal of Computer Science & Engineering Technology (IJCSET*)* [Accessed 2 Apr. 2018].

Turk, D. E., Bernard R. and Robert France. (2005) *Assumptions Underlying Agile Software Development Processes*. Journal of Database Management. Oct. 2005. 16(4):62-87. [online] Available at: https://www.researchgate.net/publication/220373635_Assumptions_Underlying_Agile_Software_e-Development_Processes. [Accessed 05 Sep. 2018]

Van den Broek, R., Bonsangue, M.M., Chaudron,M. And Van Merode,H. (2014). Integrating Testing Into Agile Software Development Process. [online] https://ieeexplore.ieee.org/. Available at: https://www.researchgate.net/publication/224755611_Integrating_Security_into_Agile_Development_Methods [Accessed 28 Apr. 2018].

Yenduri, S. and Perkins, L., 2006. *Impact of Using Test-Driven Development: A Case study*. IEEE Conference Publication. Available at: http:// webofknowledge.com [Accessed 3 May 2018].

Appendix A : Interview questions and answers

General information

Why do you use Agile methodology in Cybercom?

Tester 1

We use Agile because in Agile everyone has to do everything; everyone should do everything. We like family, so as Agile developer and tester you should do whatever. In Agile testing we do testing normally, but I don't know, Agile is more comfortable. Like, if design team are busy the test team help them. We support and help each other. We follow the principle of Agile methodology. Cybercom is a consulting company and work for different companies [clients, ed.]. For Agile testing, we define goals for three weeks and try to get the goals based on what customers say. So, we feel like working fast, we talk and meeting with other tester easy and we discuss to each other and try to solve problems.

Tester 2

Customers change their idea often, yeah? And I think Agile and scrum are good for doing this kind of project; flexible and everything. Also, tester should listen to feedback from the customer and the colleagues, yeah? We work with colleagues very friendly and in direction of the same goals, same idea, same style, yeah? Team gives priority about what we should do so I think the process is good and discipline is great, we always try to listen to the feedback, we try to move forward and improve the group work.

Tester 3

I started here four years ago. I remember it was decided then to have Agile teams in Cybercom office because I guess it is important for our clients their feedbacks and our clients want from us to do Agile method to support clients' work. They, I mean clients, want a team that handles developing and testing in the same team and to do great job, you know, so we have this cross functional Agile teams and we have two teams and in each team we have three roles: testers, developers, scrum master. so that makes the developer develop the feature we can both develop it and test it with. That is the idea, the goal, and we always do it great.

Tester 4

Agile method is popular because we can identify the task, we can do the task in shorter time and deliver day to day, identify features in product backlog, understand and follow the progress. And these are not everything good with scrum but this is what I can say right now. Testing also happens and that is the most important thing in the Agile for that reason Agile methodology is popular and it has flexibility and functionality.

What is your Agile method of choice? Why?

Tester 1

Scrum! Scrum method is very good for Cybercom because we have many planning and actions with it. We see that we have some issues, we have not long time, every three week we have iterations, you know, iteration is part of sprint and we define some goals for sprint with iterations. I configure base band nodes to different configuration and run some test for base band nodes. In sprint planning, we estimate features. During sprint we try to achieve goals of scrum meetings. We set short goals for every 3 weeks, and we have daily stand up meeting and team knows what we should doing and what other team should do and in review meeting we deliver final product to customer. For that reason, scrum method selected. In daily stand up meeting, you know, scrum meeting, we will get solutions easier. Also in sprint retrospective meeting happened before the next sprint to begin [Retrospective meetings are held after finishing the latest sprint and right before starting the next one, ed.]. In scrum, we analyze plan of every sprint complete and review very much.

Tester 2

We use Scrum. Scrum is a little bit different from Kabana. But It is a process and process is always good for parallel projects and we have many parallel projects. Many projects have relations and interconnected with each other and some features have relations with each other. Actually, we have effective handling too. Scrum has a short iteration for achieving the goals so it's actually so quick. we have pair programming and we share what we know because when colleagues know everything their colleagues know, it's so easy and fun to work.

Tester 3

Depends on the project, you know? So sometimes we use the Scrum or Kanban. Some projects are more better for Kanban. For example, if you are doing a lot of trouble reports, you know, failure reports, and correcting trouble reports, then it is more easier to use Kanban because you cannot really predict how long time it gets [how long it takes, ed.] to fix the trouble report so it will be hard to use Scrum when we do maintenance work on product and we care about our work. Now we are doing capacity and developing new features less, so we must know how long features need to developing and to testing and to predicting the duration so we plan better. We use Scrum instead because it is great and more fit for understanding the tasks in a project. Scrum is Totally fit [suitable, ed.] when we have planning and we have from before a priority of things [pre-determined prioritization, ed.]. When we do sprint planning we can so simple say approximately how much we do in three weeks and scrum is the best to have a complete planning.

Tester 4

Scrum of course! Scrum tells you what activities you need to do on Scrum board, like identifying tasks, features, they can be visualize on the board all of them, so everyone has clearly understand where we are and what is blocking our work and what are issues and in case everyone need to focus. Also, Scrum tells you which part of the project you should deliver in the end of a sprint, you know? So you have better time management. Also, Scrum is great and it is effective when we identify details. Yeah, details are very very important and in Scrum you care about that very much. Usually what we do is this: we get product backlog that is set by requirements, so first we need to find out product backlog, okay? So, for example, in product

backlog you have ten features, we identify very important things that we want to be ready for deliver to clients, okay? So we have sprint and we should deliver some part of the product in a sprint. Usually in Cybercom a sprint is three weeks and during the three weeks, we should do one feature. I have experience of many tools and nothing is better to [than, ed.] Scrum.

Let's assume it's time to deliver the product to the client, but your work is not yet finished. What are you supposed to do?

Tester 1

If team miss the time [deadline, ed.], you know, for all sprints we inform to PO [product owner, ed.] to do the features for [during, ed.] next sprint.

Tester 2

If we want to improve our program or maybe we have the short time [lack of time, ed.], we should talk to PO to continue what is not done in the next sprint.

Tester 3

If we spend time to repair a code and we can't solve it before deadline, we need to continue for next sprint but the PO must know and do [make, ed.] the decision.

Tester 4

When our team lacks time or should improve a part of product, we just work well on what we did to now [we have done so far, ed.] and then for next sprint we ask more time to complete it and there's no problem.

How do you perform Agile testing?

Tester 1

I am tester, okay? Cybercom designers design codes and they do unit test. Unit test is only test designers do and we don't. So they don't perform all tests but in our part of testing we perform many testing and we should understanding the code and features. We do stability test, if some features define limit base band, we try to find crashes, and we test on the limit, we test every phases. After find crash or bug, we report to developer and developer will solve bug etc. etc. Sometimes there are issues for hardware too, but designer responsible is for code not hardware. About hardware problem, there is a lab team that its job is solve this. Testers, you know, I am tester, we find bugs, okay? If we don't have [find, ed.] crash or bug we deliver to test manger and test manager deliver to release part. In that part, I mean release, acceptance test happens and in end, you know, when everything okay and everything work well, you deliver product to customer. But if we find the bugs, we write problem report for developers and they fix it. Design team and test team help each other. In part of Agile testing we have unit testing that developers run, functional testing that testers run, capacity and system testing run by tester. In unit testing developers don't do complete test. They test some feature, some change, etc. Test functionality really test for normal traffic and just load software to

hardware, you know, we put the code and software in hardware to work, okay? If software is load or if it's not load, we tell to capacity and system team "you guys wait, there is problem and we fix it soon." Test capacity really related to behavior of new feature in nodes. For example, how many user can connect to nodes? You know and find capacity for one hardware [we examine and determine the capacity of each hardware, ed.]. System testing find between different hardware, okay? You have one product and you want to know what in each city they perform, problems, everything.

Tester 2

For Cybercom, I mainly work for clients. But everything is good because we have two design teams and professional testers who mix in two design teams; the name of the design teams is "Santinear" and "Spektera." We do integration testing and Regression testing, we do capacity testing and Functional testing and one team does the performance testing and system testing. After, releasing team do the Acceptance testing. Mainly tests get done in Base band and it has a lot of blocks and a lot of paths and a lot of functions. Actually, the base band is the core of telecommunications like 3G, 4G, 5G, Uplink, Downlink... So, let me say like this: Designers write a unit testing and developers check it and pass the tests, yeah? After that we deliver to the test team and they do specific test on it and two test teams work as parallel. Many parameters are important for system testing and performance testing, you know, for example how many users, requirements, subsystems, all of that. But system testing and capacity testing are different and work parallel, yeah? In part of testing they have many activities in companies: for example, 3G, 4G, 5G and between them has a good interface. Then we run system testing, performance testing, functional testing, feature testing, that's it. Acceptance testing is the last one, yeah? Last part of the testing. In Sweden we have three some famous operator companies, yeah? We have Telia, Three (3), Tele2, Telenor and some more, yeah? But we work with Three. This company has four pro [professional] testers too, like Cybercom, and they run the tests most of them [mostly, ed.] by machinery and they often do manual. So, let's say like this: if your telephone doesn't work, you can't talk, yeah? So, you should have a good hardware before you have a software, yeah? We have seven designers and two testers for in our teams. We run the automatic [automated, ed.] testing for to save time and higher [enhance, ed.] the quality. Also, we have support team, yeah? Support team can work with many consult [consulting, ed.] companies. Like my company, our company here, Cybercom, is a consult company, yeah? In support teams again we divide to two parts: Software team and Hardware team, yeah? Software, for example, uplink some features is in Canada, Poland, Beijing. Then we have downlink is in Sweden, Baseband. For hardware part they borrow us [lend us, ed.] some equipments.

Tester 3

We are actually two teams and working in a same project and we have total four professional testers. So, designers develop something and send it to the testers in the team and testers test it and give feedback to designers and maybe they want to add some traces. I mean, if the crash or something has a problem then they can go to the tester and ask 'Please run again with this software, I added some trace.' And you know, the testers do it again. The testers actually look at some codes or configuration of the nodes. Sometimes the testers do manual testing too, sometimes they change a parameter in a base band or sometimes we can change the system constants and we can do some things manually and then do some tests after. Integration testing, regression testing, release testing does it [are done, ed.] by both teams. But it depends from where the feature is in the development so two teams work parallel. And we do continuous integration, you know? To reduce time of regression test. We don't do much functional testing, I think. I think capacity testing can

be in all the interfaces both integration because we try to have a different way, different modules. Actually, we have a really large system in some client company, some small part here, then we delivering something and delivering another part of the software and we increase the capacity. We do some integration testing together with the other software; I mean, so maybe three or four different codes we write for base and then we do the integration testing. And then also we do regression testing to see when we add something, we don't lose the legacy capacity or it isn't affected [affected, ed.] by the changes we make. So, still we actually need to do regression testing for the capacity changes, too. I mean we do it to check legacy, you know? System testing and Integration testing we do after we see what Exploratory testing and session-based testing brings. Actually, Exploratory testing is also a very good solution for remove bugs as machinery. But TDD [test driven development, ed.] and BDD [behavior driven development, ed.] we don't run in Cybercom.

Tester 4

I work in two different projects, here and in Eriksson, too. One is the system testing area, but currently just functional testing. In Cybercom developers develop the codes and then write basic unit tests and then they have to write EBT [extended base test, ed.] in order to see the build and know the functionality changes. We can push the codes to master (actual software) or Upgrades (up). After this part we have verification done by testers and then we go to delivery and again we have up. The part of testing, actually we have different testing methods to perform, for example in my team, testers get responsible for capacity testing, and area of capacity testing in different country like Canada, Poland... every team in different countries have capacity features to the test. We have continuous integration too. It is done automatically and has different functionality and totally upgrades the package in testing part. This happens after the verification and after we decide what features to test. On the other hand, some testing is done by clients, did you know that? And we do not have access to them so we don't know how they are. System testing is performed with the other team. That's usually how we do it.

What are the roles of testers?

Tester 1

Tester will have different roles: we define scope / define some test case for every nodes/perform test case. scope means feature write by developers and tester need to understand the feature what it is and what need to define and what testing we need to with that feature. Actually, experience of tester provides better situation for testing process and detect testing failure. We keep all test cases in excel file, actually what test cases it is and what are test cases step by step. it is easy for another tester what ding also. Actually, professional testers assess the quality of the work. When tester find the issues write the trouble report and sent to developers.

Tester 2

I think it's more free inside Cybercom about [when it comes to, ed.] sharing knowledge but I think Cybercom has successful push and sharing knowledge with clients too. Also, we have pair programming, you know, tester sits with designer and they work and the feedback is on the spot and we call it sometimes synergy. We try to help each other. Actually, testers have a motivation to protect product from the bugs and so we use the different types of testing to make sure no bugs are in code and anyone will [no one can, ed.] find bugs. When the developers write the code, testers should define scope and define the test cases before

designers deliver codes to you. We send feedback to the designers for them to solve or discuss the failure part, yeah? Many communications between designers and testers happens and it's very good because we know everything our colleague knows with communication, meeting, talk, etc.

Tester 3

Testers have a very important roles in test teams, so we should have many skills and knowledge and experience.

Tester 4

In Cybercom tester and developer work with each other and when testers find the bug we write reports for developers and sometime have meeting and stuff with developers to get how to solve that. It is important to understand the features and testing concepts. Testers support and help developers and vice versa too. In my team, my college and me are testers and when we see problems we report to developers and we also do other activities in test area. So, when we are supporting developers, we are doing testing on different clients, projects, etc. and we do all that without problem.

Why do you not perform Test Driven Development and Behavior Driven Development testing?

Tester 1

Clients decide we use this or not, but most times we do not use this because it decreases code coverage.

Tester 2

TDD and BDD have some advantage and disadvantage but it depends of our assignment we [whether we should, ed.] use this, yeah? And on the other hand it need experience very much.

Tester 3

Depends of our client, if they do not want it we don't do it. but it needs more experience to do that, too.

Tester 4

Sometimes it takes more time and decreases level of code coverage. Clients don't want it from us.

What test cases do you use?

Tester 1

Tester should know about test cases, and when developers write codes, we read test cases, most of them wrote clients [are written by clients, ed.].

Tester 2

We perform different test cases, all of them comes from clients.

Tester 3

We start to [with, ed.] configuration node and make sure node can run it. Then we check crash and solve it. Then we use uplink, downlink, we validate again. We have some requirements in this, and we have ten test cases that capacity should get them before we want to deliver the feature to client, and features add time to cells in base band. If we have problem we go and want [get, ed.] help from a expert in download data team. All clients give so much guidance to us, and it is great because if we can't solve, we need help, guidance, ideas, etc. for trouble shouting. We don't see [deal with, ed.] the hardwares, we are just testing softwares. So, we want to test softwares on a real hardware but you know, I said it, we do not test hardware because test of hardware is with another team to do it. We should have a big picture about ten test cases need to be verified for this feature and then we run automatic everything, like signaling test case, Ping test case, Uplink test case, Downlink test case, UL/DL [upload, download, ed.] test case. When we done and everything finish, deliver to test manager and test manger give us requirements and talk to client company. Somebody is responsible for everything run true [flawlessly]. So, first, Capacity/Functional, then integration. We don't check all testing. Then System testing and release testing, this is when we check all testing then other testing in [is performed at the] client company's test teams. So, then we deliver to customer a product and if they find problems, they write TR [Trouble Report, ed.] and send to Cybercom team and we see the report.

Tester 4

Clients give us ten test cases, when developers write the codes, testers should double-check them to find the bugs easier.

Could you explain Capacity testing and Functionality testing a bit further?

Tester 1

Functionality test we use for normal traffic, and to see how our software loads in hardware. Capacity testing is connect to quality of new features work in nodes [relates to the way new features behave in the nodes, ed.]. For example, we ask this question 'how many users can connect to this nodes?' and capacity test answer that very nice. For system testing, we review all hardware to understand how connects with Exploratory testing like how different the product performance in case A and case B.

Tester 2

We use capacity test when upload software to hardware in order to know about speed. Functionality test check the behavior of code functionality**.**

Tester 3

Both of them are useful for product, we use both of them to improve and check products better.

Tester 4

In capacity testing we are responsible for measuring speed of upload and different parameters like speed of download. For capacity testing, we check for example how many numbers of cells we have or how much number of cells are increasing, we check base band of product, we check how many users connect to base band, we check number of cell phones and number of users, and capacity. For functional testing we check to make sure functional areas support all features, then we use real hardwares and on them we test software to make sure features are working. Actually, functional testing is testing features, capacity is a feature and it is not tested. Finally, we run functional test to actually test the functionalness of code and make sure it works well.

How do you manage the situations where bugs are found?

Tester 1

If everything is fine we report to manager and deliver product to the release part. They do acceptance test after they are sure no bugs deliver to customer. Testers check all feature to sure no bugs are in it. But if we meet to bugs we write trouble report and inform to developer fast. You know, we're very nice and friends so we work friendly and work with each other and we work for fix the bugs etc. Acceptance test is for release team when they go to deliver customers product. Release team check all features again, again, again, to sure that everything is okay.

Tester 2

Actually, as I told before Scrum is best method really to find the bug and solve it. They ask us to do pair programming for finding solution and it's great. We work with team members and we don't scare [are not afraid, ed.] to say everything because examining all features, sometimes is surface. Many departments in client company are in different country and maybe for debugging they have more time. The members of the team can have a motivation to use the different types of the test because we don't want any bug to stay in system.

Tester 3

If we deliver something then find problem in it, we write trouble report to client company. If they want us to do [make, ed.] changes, we try to solve it very soon.

Tester 4

We have a requirement sheet and a report sheet and on them we have to write test number. We run test cases, okay? If it's pass, it's okay. But if a crash happened or a bug happened or function was broken, testers write report for developers and developers and testers work with each other and work friendly. We should have peace at work and respect the principles of Agile developers because it is very important and useful.

What are the problems you face regarding product validation?

Tester 1

The main problem is something we call it ‘out of memory’ and designers solve this problem. Designers check other block and if it has free space give to other part. Other problem is crash. If crash in real network happened the nodes goes down and you do not do any things.

Tester 2

The main problem is the test case is not pass, reason can be in troubleshooting of hardware or software reason, yeah? More than Fifty percent problem is the hardware because hardwares are not good enough and they are very bery expensive in Sweden. So, the hardware problem has a bad effect on quality of the product.

Tester 3

if we have problems the we send it back to the designer, so they try to fix. Problem is mostly software problems. Most of the problems happen when we do testing so for validation we see no problem. We test a lot so we do not have many problems in validation, I think.

Tester 4

First developers perform test on software, after performing test we have the validation part and in this part we do not have problems because many tests was already done and we fixed all of them.

Testing using different tools

What tools do you usually use for testing?

Tester 1

Doing test has two ways: one, automated testing that we use the JCAT tool that named Java coden Automation tools. Two, manual test. We have different team for different way of test. We test software to test things we should load to hardware. But, really we do the manual test depends on the model of project. Automation tool repeat test and repeat test a lot and help tester performs test quickly, no need to human interaction, you know? Manually test we do now and always automated test we do now again [we always do the automated and manual testing simultaneously, ed.]. But from a different way of things [on the other hand, ed.] testers have motivation to use automated test for implementation different models of test for repeat test is very very easy automated. And we repeat integration very much to sure all codes, you know, available because perform better for regression test.

Tester 2

We perform the testing machines with JCAT tool to remove failure and improve quality of product according with customer expectation, yeah? We will find stable and upgrade packages. And clients say the framework [determine the framework, ed.] for writing the test case and JCAT is not editable. Client companies have testing team too and the test engineer in the client company writes the test cases and send for us. I said this: we can’t change and modify main tasks, just change or edit the input. The JCAT machines

show with a flag to you that test is done successfully or not. If flag is red, we reject it and report to the designers.

Tester 3

For node testing we have some clients tools like TGF [Transforming Growth Factor, ed.] for integration and regression testing. It is automated testing, you know. TGF collect data from JCAT and put it in a web page/ TGF starts some JCAT and TGF controls some parts. JCAT executed the test and sends back result to the TGF web page. It's complicated.

Tester 4

We use JCAT and TGF as a automated testing. JCAT is good for operation testing, automation basic structure. TGF performs the tests and it is actually the testing operation of JCAT. We think that using automated testing can be useful for parallel projects because it has conditions that help very hell.

What are the advantages of your preferred testing tool?

Tester 1

JCAT is flexible, you know, very flexible because usually for JavaScript and we define test case for java script and it is easy for we write test case because we have this support functions and many different times for use of cases.

Tester 2

It is centralized and work so very fast and 99 percent of the work so yourself do not need more spend time to write test cases. You can also change different parameters to run tests so quicker.

Tester 3

We have manual and automated testing, okay? For automated testing, we don't need the human especially when the test repetition a lot [needs to be repeated many times, ed.] and this type of testing increase efficiency and test coverage, okay? Automated testing shorter [shortens the, ed.] time implemented test and reduce the cost very much. I am a tester so I find the motivation to use the automation tools and different kinds of tests.

Tester 4

It can be performed on different platforms, it's unique, it's really suitable for java script. It collects all logs and compares logs very easy. Generally, it gives better performances for test execution.

What are the disadvantages of your preferred testing tool?

Tester 1

JCAT is very hard sometimes. You know, not easy to track it [it is difficult to monitor all processes with JCAT, ed,] and sometimes spend more time so it is good but yeah, this part very hard.

Tester 2

JCAT is a little slower.

Tester 3

I work on node testing on hard wares, you know. Node testing is small testing for whole system. So client company decide for me and this tricky part: configuration of things and it is not so easy to understand where all the parameters come from TGF. Sometimes for fault values we cannot change, it is very tricky for configuring of nodes before starting tests.

Tester 4

In my opinion, JCAT is a little complicated and only use it for special client. It's not widely used, needs all details and so many things.

Release section

How do you perform Acceptance testing?

Tester 1

Release team do acceptance test, we don't do. They check features again to sure no bugs happened because it is very important and no bug there is in code. Test manager when receive the product, check all things again to no bugs find, check again, check again. But if test manager find bugs, he send back to us so we solve it very fast.

Tester 2

Testers of release section use acceptance testing because they wanna be sure it doesn't have no bugs. They perform tests not based on the customers want. If they find bugs, they write trouble sheet and send for us.

Tester 3

In release section, they use acceptance testing. Normally they check all things in sections until be sure no bugs stay. If they find error so inform us to remove it.

Tester 4

Acceptance testing in Cybercom is the duty of release section. They do a final double-check before delivering the product to customer. Test manager should check everythings before deliver it to release section.

Appendix B: List of 205 articles

	Authors	Article Title
1	Schlunder, C; Puschkarsky, K; Rott, GA; Gustin, W; Reisinger, H	NBTI: Experimental investigation, physical modelling, circuit aging simulations and verification
2	Manu GV, Namratha M, Pradeep	Benefits of Test Automation for Agile Testing
3	Hellmann T.D, Sharma.A, Ferreira.J,Maurer.F	Agile Testing: Past, Present, and Future
4	Garousi, V; Eskandar, MM; Herkiloglu, K	Industry-academia collaborations in software testing: experience and success stories from Canada and Turkey
5	Yaman, SG; Munezero, M; Munch, J; Fagerholm, F; Syd, O; Aaltola, M; Palmu, C; Mannisto, T	Introducing continuous experimentation in large software-intensive product and service organisations
6	Chauhan, BD; Rana, A; Sharma, NK	Impact of Development Methodology on Cost & Risk for Development Projects
7	Gupta, S; Gouttam, D	Towards Changing the Paradigm of Software Development in Software Industries: An Emergence of Agile Software
8	Singh, AK; Vinodh, S	Modeling and performance evaluation of agility coupled with sustainability for business planning
9	Rohatagi, S; Profit, D; Hatch, A; Zhao, C; Docherty, JP; Peters-Strickland, TS	Optimization of a Digital Medicine System in Psychiatry
10	Truong, D; Jitbaipoon, T	How Can Agile Methodologies Be Used to Enhance the Success of Information Technology Projects?
11	Rangel-Landa, S; Casas, A; Davila, P	Facilitation of Agave potatorum: An ecological approach for assisted population recovery
12	Moe, NB; Cruzes, D; Dyba, T; Mikkelsen, E	Continuous software testing in a globally distributed project
13	Bider.L,Söderberg.O	Moving towards Agility in an Ordered Fashion
14	Causevic, A; Sundmark, D; Punnekkat, S	Impact of Test Design Technique Knowledge on Test Driven Development: A Controlled Experiment
15	Tripathi.V, Kumar Goyal.A.	Agile testing challenges and critical success factors
16	Winter, J; Ronkko, K	SPI success factors within product usability evaluation
17	Koyuncu, E; Ure, NK; Inalhan, G	Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs
18	Yenduri .S.,Perkins L.A.	Impact of Using Test-Driven Development: A Case Study
19	Ganesh.N, Thangasamy,S	New Agile testing Modes
20	Li, N; Offutt, J	A Test Automation Language Framework for Behavioral Models
21	Stolberg, S	Enabling Agile Testing Through Continuous Integration
22	Kammergruber, R; Durner, J	Laboratory information system and necessary improvements in function and programming
23	Ibba, S; Pinna, A; Lunesu, MI; Marchesi, M; Tonelli, R	Initial Coin Offerings and Agile Practices
24	Essebaa, I; Chantit, S	Scrum and V Lifecycle Combined with Model-Based Testing and Model Driven Architecture
25	Rao, NR; Sekharaiah, KC; Narayan, R	Aspects Oriented Approach towards the Bug Fixing Performance Lifecycle
26	Teka, D; Dittrich, Y; Kifle, M	Adapting Lightweight User-Centered Design with the Scrum-Based Development Process
27	Srivastava, A; Kapur, PK; Mehrotra, D	Modelling Fault Detection with Change-point in Agile Software Development Environment

28	Lockhart, J; Purdy, C; Wilsey, PA	The Use of Automated Theorem Proving for Error Analysis and Removal in Safety Critical Embedded
29	Teka, D; Dittrich, Y; Kifle, M	Contextualizing User Centered Design with Agile Methods in Ethiopia
30	Paul, M; Covert, C; Cembrinski, T	Quality assurance and Agile Prototype Development for a Small Planetary Lander
31	Hernandez, RVR; Izaguirre, JAH; Mendoza, AL; Escandon, JMS	A Practical Approach to the Agile Development of Mobile Apps in the Classroom
32	Nidagundi, P; Novickis, L	Introducing Lean Canvas Model Adaptation in the Scrum Software Testing
33	Raunak, MS; Binkley, D	Agile and Other Trends in Software Engineering
34	Grashiller, M; Luedeke, T; Vielhaber, M	INTEGRATED TO THE AGILE DEVELOPMENT WITH DESIGN IN AN INDUSTRIAL ENVIRONMENT
35	Castillo-Barrera, FE; Amador-Garcia, M; Perez-Gonzalez, HG; Martinez-Perez, FE	Agile evaluation of the complexity of user stories using the Bloom's Taxonomy
36	Harvie, DP; Agah, A	Targeted Scrum: Applying Mission Command to Agile Software Development
37	Rahayu, P; Sensuse, DI; Fitriani, WR; Nurrohmah, I; Mauliadi, R; Rochman, HN	Applying Usability Testing to Improving Scrum Methodology in Develop Assistant Information System
38	Chukhrai, NI; Shcherbata, TS	Cooperation between IT development enterprises and universities
39	Considine, C; Crowley, S; Gardner, G; Shannon, C; Kwinn, MJ; Henderson, SJ; Santamaria, P	Architecting a Development and Testing Plan for the Army's Common operating
40	Gunsing, J; Gijselhart, F; Hagemans, N; Jonkers, H; Kivits, E; Klijn, P; Kapteijns, B; Kroeske, D; Langen, H; Oerlemans, B; van Stuijvenberg, JOJ	Adaptive Robotic Systems Design in University of Applied Sciences
41	Jha, MM; Vilardell, RMF; Narayan, J	Scaling Agile Scrum Software Development Providing Agility and Quality to Platform Development
42	Eckhart, M; Feiner, J	How Scrum Tools May Change Your Agile Software Development Approach
43	Farid, AB; Fathy, EM; Abd Ellatif, M	Towards Agile Implementation of Test Maturity Model Integration (TMMI) Level 2 using Scrum Practices
44	Raj, G; Yadav, K; Jaiswal, A	Emphasis on Testing Assimilation Using Cloud Computing for Improvised Agile SCRUM Framework
45	Sultania, AK	Developing Software Product and Test Automation Software Using Agile Methodology
46	Mlitwa, NBW; Koka, AI	ACADEMIC OFFERINGS AND OPERATIONAL NEEDS OF THE AGILE SOFTWARE DEVELOPMENT
47	Alaa, G; Samir, Z	A multi-faceted Roadmap of Requirements Traceability Types Adoption in SCRUM: An Empirical Study
48	Jeeva Padmini.K.V, Kankanamge,P.S,Dilum Bandara,H.M.N, and Perera,G.I.U.S.	Challenges Faced by Agile Testers: A Case Study
49	Gregorio, DD	How the Business Analyst Supports and Encourages Collaboration on Agile Projects
50	Caballero, E; Calvo-Manzano, JA	A Practical Approach to Project Management in a Very Small Company
51	Eschbach, R; Jung, J; Trefz, S	Standard compliant quality assurance in agile software development projects for safety-critical agricultural systems
52	Goncalves, J; Santos, C	POLVO - Software for Prototyping of Low-Fidelity Interfaces in Agile Development
53	Pereira, V; do Prado, AF	Introducing a New Agile Development for Web Applications Using a Groupware as Example
54	Reichlmayr, T	Working Towards the Student Scrum - Developing Agile Android Applications

55	Schramm, W; Draeger, C; Grechenig, T	Issues and Mitigation Strategies when using Agile Industrial Software Development Processes in Software Projects
56	Hussain, Z; Slany, W; Holzinger, A	Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective
57	Menachem, S; Reich, Y	IMPROVING EFFECTIVENESS OF AGILE DEVELOPMENT
58	Rottier, PA; Rodrigues, V	Agile development in a medical device company
59	Puleio, M	How not to do agile testing
60	Hodgetts, P	Experiences integrating sophisticated user experience design practices into Agile processes
61	Collins.E, Dias-Neto.A, de Lucena Jr.V.F	Strategies for Agile Software Testing Automation: An Industrial Experience
62	Puleio, M.	How Not to Do Agile Testing
63	Martin, A., Sillitti, A. and Wang, X	Agile project management In Software Engineering And Extreme Programming:
64	Madeja, M; Binas, M	A Risk Poker Based Testing Model For Scrum
65	Van den Broek, R., Bonsangue, M.M., Chaudron,M. And Van Merode,H.	Integrating Testing Into Agile Software Development Process.
66	Berlowski, J., Chrusciel, P., Kasprzyk, M., Konaniec, I. and Jureczko, M.	Highly Automated Agile Testing Process: An Industrial Case Study
67	Lockhart, J; Purdy, C; Wilsey, PA	Modelling Fault Detection with Change-point in Agile Software Development Environment
68	Paul, M; Covert, C; Cembrinski, T	The Use of Automated Theorem Proving for Error Analysis and Removal in Safety Critical Embedded System
69	Nidagundi, P. and Novickis, L.	Introducing Lean Canvas Model Adaptation in the Scrum Agile Testing
70	Kayes, I; Sarker, M; Chakareski, J	INTEGRATED APPROACH TO THE AGILE DEVELOPMENT WITH DESIGN THINKING
71	Heath, CDC; Baron, T; Gary, K; Amresh, A	Product backlog rating: a case study on measuring test quality in scrum
72	Jakobsen, C. R. and Johnson, K. A.	Mature Agile with a Twist of CMMI
73	Mauro, V; Messina, A	Scaling Agile Scrum Software Development Providing Agility to Platform Development by Reducing Time to Market
74	Farid, AB; Fathy, EM; Abd Ellatif, M	AMINSEP-Agile Methodology Implementation for a New Software Engineering Paradigm Definition.
75	Crispin.L	How Session Based Testing Impacts Software Quality
76	Yahya, N; Abu Bakar, NSA	ACADEMIC OFFERINGS AND OPERATIONAL NEEDS OF THE AGILE SOFTWARE DEVELOPMENT
77	Butgereit, L	Using Machine Learning To Prioritize Automated Testing In An Agile Environment
78	Ghazali, SNH; Salim, SS; Inayat, I; Ab Hamid, SH	THE CHANGE IN PROJECT INDUSTRY IN SOUTH AFRICA
79	Elallaoui, M., Nafil, K., Touahni, R. and Messoussi, R.	Automated Model Driven Testing Using AndroMDA and UML2 Testing Profile in Scrum Process
80	Cedeno, JGI; Arcia, LAO; Vera, ELG; Cedeno, AAM	THE CHANGE IN MANAGEMENT STYLE DURING THE COURSE OF A PROJECT FROM THE CLASSICAL
81	Gablas, B; Ruzicky, E; Ondrouchova, M	IN MANAGEMENT STYLE DURING A PROJECT
82	Kammergruber, R; Durner, J	Laboratory information system and necessary improvements in function and programming
83	Ibba, S; Pinna, A; Lunesu, MI; Marchesi, M; Tonelli, R	Initial Coin Offerings and Agile Practices

84	Ghazali, SNH; Salim, SS; Inayat, I; Ab Hamid, SH	A Risk Poker Based Testing Model For Scrum
85	Essebaa, I; Chantit, S	Scrum and V Lifecycle Combined with Model-Based Testing and Model Driven Architecture to Evolutionary System
86	Ertaban, C; Sarikaya, E; Bagriyanik, S	Agile Performance Indicators for Team Performance Evaluation in a Corporate Environment
87	Pastor, D; Arcos-Medina, G; Onate, A; Loaiza, M; Torres, J	Semantic Query System for Moodle Virtual Courses Based on an Ontology
88	Infante, P; Quingatuna, A; Nogales, S; Hidalgo, B; Flores, T; Gilard, F	Development of a Web System-based Geographic Information System Technologies to Mapping Electromagnetic
89	Bagiu, N; Avasilcai, S; Alexa, L	An agile approach to develop a crowdsourcing platform: the case of Cre@tive.biz
90	Rao, NR; Sekharaiah, KC; Narayan, R	Aspects Oriented Approach towards the Bug Fixing Performance Lifecycle
91	Teka, D; Dittrich, Y; Kifle, M	Adapting Lightweight User-Centered Design with the Scrum-Based Development Process
92	Babic, F; Gaspar, V; Satala, P	New trends in mobile technologies education in Slovakia: an empirical study
93	Madeja, M; Binas, M	Implementation of SCRUM methodology in programming courses
94	Liu, Z; Li, F; Wu, CZ; Liu, HT; Zhang, JJ	A Study of Agile Methodology - Scrum for Cockpit Simulation
95	Hanslo, R; Mnkandla, E	Scrum Adoption Challenges Detection Model: SACDM
96	Tyagi, S; Sibal, R; Suri, B; Wadhwa, B; Shekhar, S	Development of Reusable Hybrid Test Automation Framework for Web Based Scrum Projects
97	Thoumy, M; Possik, C	Single Developer Scrum Model: An Exploratory Study
98	Cedeno, JGI; Arcia, LAO; Vera, ELG; Cedeno, AAM	ASSESSMENT MANAGEMENT SYSTEM IN THE POLITECNICA AGROPECUARIA DE MANABI
99	Aamir, M; Khan, MNA	Incorporating quality control activities in scrum in relation to the concept of test backlog
100	Lechler, TG; Yang, SW	Exploring the Role of Project Management in the Development of the Academic Agile Software Discourse
101	Srivastava, P; Jain, S	A leadership framework for distributed self-organized scrum teams
102	Daly, C; Zapata, C; Paz, F	Improving the Usability in a Video Game Through Continuous Usability Evaluations
103	Srivastava, A; Kapur, PK; Mehrotra, D	Modelling Fault Detection with Change-point in Agile Software Development Environment
104	Sensuse, DI; Satria, D; Pratama, AA; Wulandari, IA; Mishbah, M; Noprisson, H	Integrating UCD into Scrumban for Better and Faster Usability Design

105	Butgereit, L	When Agile Meets Rigid: A Case for Automated Testing with Tangible Printed Reports
106	Bomfim, MM; Santos, VA	Strategies for Reducing Technical Debt in Agile Teams
107	Teka, D; Dittrich, Y; Kifle, M	Integrating Discount Usability in Scrum Development Process in Ethiopia
108	Lockhart, J; Purdy, C; Wilsey, PA	The Use of Automated Theorem Proving for Error Analysis and Removal in Safety Critical Embedded System
109	Teka, D; Dittrich, Y; Kifle, M	Contextualizing User Centered Design with Agile Methods in Ethiopia
110	Paul, M; Covert, C; Cembrinski, T	Agile Prototype Development for a Small Planetary Lander
111	Awar, KB; Sameem, MSI; Hafeez, Y	A Model for Applying Agile Practices in Distributed Environment: A Case of Local Software Industry
112	Hernandez, RVR; Izaguirre, JAH; Mendoza, AL; Escandon, JMS	A Practical Approach to the Agile Development of Mobile Apps in the Classroom
113	Nidagundi, P; Novickis, L	Introducing Lean Canvas Model Adaptation in the Scrum Software Testing
114	Raunak, MS; Binkley, D	Agile and Other Trends in Software Engineering
115	Ardito, C; Baldassarre, MT; Caivano, D; Lanzilotti, R	Integrating a SCRUM-based process with Human Centred Design: an Experience from an Action Research Study
116	Poller, A; Kocksch, L; Turpe, S; Epp, FA; Kinder-Kurlanda, K	Can Security Become a Routine? A Study of Organizational Change in an Agile Software Development Group
117	Grashiller, M; Luedeke, T; Vielhaber, M	INTEGRATED APPROACH TO THE AGILE DEVELOPMENT WITH DESIGN THINKING
118	Castillo-Barrera, FE; Amador-Garcia, M; Perez-Gonzalez, HG; Martinez-Perez, FE	Agile evaluation of the complexity of user stories using the Bloom's Taxonomy
119	Gupta, RK; Manikreddy, P; Arya, KC	Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey
120	Kayes, I; Sarker, M; Chakareski, J	Product backlog rating: a case study on measuring test quality in scrum
121	Harvie, DP; Agah, A	Targeted Scrum: Applying Mission Command to Agile Software Development
122	Rahayu, P; Sensuse, DI; Fitriani, WR; Nurrohmah, I; Mauliadi, R; Rochman, HN	Applying Usability Testing to Improving Scrum Methodology in Develop Assistant Information System
123	Elallaoui, M; Nafil, K; Touahni, R	Automatic generation of TestNG tests cases from UML sequence diagrams in Scrum process
124	Sfetsos, P; Angelis, L; Stamelos, I; Raptis, P	Integrating User-Centered Design Practices into Agile Web Development: A Case Study
125	Chukhrai, NI; Shcherbata, TS	Cooperation between IT development enterprises and universities

126	Ngatirin, NR; Zainol, Z; Fang, WC	Mobile Agile Peer Assisted Learning
127	Gupta, RK; Manikreddy, P; Naik, S; Arya, K	Pragmatic Approach for Managing Technical Debt in Legacy Software Project
128	Considine, C; Crowley, S; Gardner, G; Shannon, C; Kwinn, MJ; Henderson, SJ; Santamaria, P	Architecting a Development and Testing Plan for the Army's Common Operating Environment Applying Agile Systems
129	Gunsing, J; Gijselhart, F; Hagemans, N; Jonkers, H; Kivits, E; Klijn, P; Kapteijns, B; Kroeske, D; Langen, H; Oerlemans, B; van Stuijvenberg, JOJ	Adaptive Robotic Systems Design in University of Applied Sciences
130	Fredrick, D; Rogers, G; Yust, T	Playful Rome: Student Game Creation as Critical Learning Practice
131	Elallaoui, M; Nafil, K; Touahni, R; Messoussi, R	Automated Model Driven Testing Using AndroMDA and UML2 Testing Profile in Scrum Process
132	Heath, CDC; Baron, T; Gary, K; Amresh, A	Reflection on Assumptions from Designing Female-Centric Educational Games
133	Jha, MM; Vilardell, RMF; Narayan, J	Scaling Agile Scrum Software Development Providing Agility and Quality to Platform Development by Reducing Time
134	Gupta, RK; Manikreddy, P; Gv, A	Challenges in Adapting Agile Testing in a Legacy Product
135	Berlowski, J; Chrusciel, P; Kasprzyk, M; Konaniec, I; Jureczko, M	Highly Automated Agile Testing Process: An Industrial Case Study
136	Rosado, N; Metrolho, JC; Ribeiro, F	Web application for academic management - a case study
137	Padmini, KVJ; Perera, I; Bandara, HMND	Applying Agile Practices to Avoid Chaos in User Acceptance Testing: A Case Study
138	Eckhart, M; Feiner, J	How Scrum Tools May Change Your Agile Software Development Approach
139	Mauro, V; Messina, A	AMINSEP-Agile Methodology Implementation for a New Software Engineering Paradigm Definition.
140	Agrawal, A; Atiq, MA; Maurya, LS	A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms
141	Scott, E; Rodriguez, G; Soria, A; Campo, M	Towards better Scrum learning using learning styles
142	Farid, AB; Fathy, EM; Abd Ellatif, M	Towards Agile Implementation of Test Maturity Model Integration (TMMI) Level 2 using Scrum Practices
143	Moyo, D; Ally, AK; Brennan, A; Norman, P; Purshouse, RC; Strong, M	Agile Development of an Attitude-Behaviour Driven Simulation of Alcohol Consumption Dynamics
144	Enriquez, C; Gomez, P	A Model for Improving Training of Software Developers in Small Companies
145	Maria, RE; Rodrigues, LA; de Vasconcelos, LEG; Pinto, AFM; Tsoucamoto, PT; Silva, HNA; Adilson, AL; da Cunha, M; Dias, LAV	Applying Scrum in an Interdisciplinary Project using Big Data, Internet of Things, and Credit Cards
146	Raj, G; Yadav, K; Jaiswal, A	Emphasis on Testing Assimilation Using Cloud Computing for Improvised Agile SCRUM Framework

147	Tommy, R; Mhaisekar, M; Kallepally, S; Varghese, L; Ahmed, S; Somaraju, MD	Dynamic Quality Control in Agile Methodology for Improving the Quality
148	Ovad, T; Larsen, LB	The Prevalence of UX Design in Agile Development Processes in Industry
149	van Kelle, E; van der Wijst, P; Plaat, A; Visser, J	An Empirical Study into Social Success Factors for Agile Software Development
150	Sultania, AK	Developing Software Product and Test Automation Software Using Agile Methodology
151	Weinreich, R; Neumann, N; Riedel, R; Muller, E	Scrum as Method for Agile Project Management Outside of the Product Development Area
152	Mlitwa, NBW; Koka, AI	ACADEMIC OFFERINGS AND OPERATIONAL NEEDS OF THE AGILE SOFTWARE
153	Bjerke-Gulstuen, K; Larsen, EW; Stalhane, T; Dingsoyr, T	High Level Test Driven Development - Shift Left
154	Vakaloudis, A; Anagnostopoulos, K	Maximising Productivity and Learnability in Internships
155	Wagenaar, G; Helms, R; Damian, D; Brinkkemper, S	Artefacts in Agile Software Development
156	Rodriguez, G; Soria, A; Campo, M	Virtual Scrum: A Teaching Aid to Introduce Undergraduate Software Engineering Students to Scrum
157	Yahya, N; Abu Bakar, NSA	The Analysis of Programming Competency in Test Driven Development
158	Ur Rahman, AA; Helms, E; Williams, L; Parnin, C	Synthesizing Continuous Deployment Practices Used in Software Development
159	Tytkowska, M; Werner, A; Bach, M	Project Management in the Scrum Methodology
160	Tarhan, A; Yilmaz, SG	Systematic analyses and comparison of development performance and product quality of Incremental process
161	Vallon, R; Drager, C; Zapletal, A; Grechenig, T	Adapting to Changes in a Project's DNA A Descriptive Case Study on the Effects of Transforming Agile Single-site
162	Guerra, VD; Segeti, E; Hino, F; Kfouri, F; Dias, LAV; da Cunha, AM; Mialaret, LFS	Interdisciplinarity and Agile Development A Case Study on Graduate Courses
163	Sarwar, S; Mahmood, Y; Ul Qayyum, Z; Shafi, I	Test Case Prioritization for NUnit Based Test Plans in Agile Environment
164	Lizano, F; Sandoval, MM; Stage, J	Integrating Usability Evaluations into Scrum: A Case Study Based on Remote Synchronous User Testing
165	Hummel, M	State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development
166	Alaa, G; Samir, Z	A multi-faceted Roadmap of Requirements Traceability Types Adoption in SCRUM: An Empirical Study
167	Janakova, M	Errors in E-Learning: The Key is Global Approach for Optimal Education

168	Vani, B; Suriya, B; Deepalakshmi, R	MANAGING PERFORMANCE OF WEB BASED APPLICATION THROUGH AGILE APPROACH
169	van den Broek, R; Bonsangue, MM; Chaudron, M; van Merode, H	Integrating Testing into Agile Software Development Processes
170	Golfarelli, M; Rizzi, S; Turricchia, E	Multi-sprint planning and smooth replanning: An optimization model
171	Melo, CD; Cruzes, DS; Kon, F; Conradi, R	Interpretative case studies on agile team productivity and management
172	Codabux, Z; Williams, B	Managing Technical Debt: An Industrial Case Study
173	Gamble, RF; Hale, ML	Assessing Individual Performance in Agile Undergraduate Software Engineering Teams
174	Gregorio, DD	How the Business Analyst Supports and Encourages Collaboration on Agile Projects
175	Caballero, E; Calvo-Manzano, JA	A Practical Approach to Project Management in a Very Small Company
176	Eschbach, R; Jung, J; Trefz, S	Standard compliant quality assurance in agile software development projects for safety-critical agricultural systems
177	Goncalves, J; Santos, C	POLVO - Software for Prototyping of Low-Fidelity Interfaces in Agile Development
178	Pereira, V; do Prado, AF	Introducing a New Agile Development for Web Applications Using a Groupware as Example
179	Stettina, CJ; Heijstek, W	Five Agile Factors: Helping Self-management to Self-reflect
180	Reichlmayr, T	Working Towards the Student Scrum - Developing Agile Android Applications
181	Williams, L; Brown, G; Meltzer, A; Nagappan, N	Scrum plus Engineering Practices: Experiences of Three Microsoft Teams
182	Schramm, W; Draeger, C; Grechenig, T	Issues and Mitigation Strategies when using Agile Industrial Software Development process
183	Nagmoti, R; Chung, S; Endicott-Popovsky, B	Game Programmers with Secure Coding
184	Erdogan, G; Meland, PH; Mathieson, D	Security Testing in Agile Web Application Development - A Case Study Using the EAST Methodology
185	Nyman, R; Aro, I; Wagner, R	Automated Acceptance Testing of High Capacity Network Gateway
186	Sovik, H; Forfang, M	Tech Challenges in a Large-Scale Agile Project
187	Larusdottir, MK; Bjarnadottir, ER; Gulliksen, J	The Focus on Usability in Testing Practices in Industry
188	Conboy, K	Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development

189	Jakobsen, CR; Sutherland, J	Scrum and CMMI - Going from Good to Great Are you ready-ready to be done-done?
190	Hussain, Z; Slany, W; Holzinger, A	Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective
191	Hussain, Z; Slany, W; Holzinger, A	Current State of Agile User-Centered Design: A Survey
192	Menachem, S; Reich, Y	IMPROVING EFFECTIVENESS OF AGILE DEVELOPMENT
193	Bondi, AB; Ros, JP	Experience with Training a Remotely Located Performance Test Team in a Quasi-Agile Global Environment
194	Maham, M	Planning and facilitating release retrospectives
195	Jakobsen, CR; Johnson, KA	Mature agile with a twist of CMMI
196	Rottier, PA; Rodrigues, V	Agile development in a medical device company
197	Geras, A	Leading manual test efforts with agile methods
198	Shaye, SD	Transitioning a team to agile test methods
199	Sutherland, J; Jakobsen, CR; Johnson, K	Serum and CMMI level 5: The magic potion for code warriors
200	Sumrell, M	From waterfall to Agile - How does a QA team transition?
201	Sulaiman, T; Barton, B; Blackburn, T	AgileEVM - Earned Value Management in scrum projects
202	Puleio, M	How not to do agile testing
203	Hodgetts, P	Experiences integrating sophisticated user experience design practices into Agile processes
204	Brechner, E	Journey of enlightenment: The evolution of development at Microsoft
205	Vanhanen, J; Itkonen, J; Sulonen, P	Improving the interface between business and product development using agile practices