USE OF SOFTWARE METRICS IN THE AGILE SOFTWARE DEVELOPMENT PROCESS

Kumara Vidanalage Jeeva Padmini
(129069h)

Master of Business Administration
In
Information Technology

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2014

USE OF SOFTWARE METRICS IN THE AGILE SOFTWARE DEVELOPMENT PROCESS

By

Kumara Vidanalage Jeeva Padmini

The dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfilment of the requirement for the Degree of Master of Business Administration in Information Technology.

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2014

DECLARATION

I hereby certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also here by give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations.

………… …………………… ....

K.V. Jeeva Padmini

(Signature of the candidate) Date

……………………………….……….……….……………………………….

To the best of my knowledge, the above particulars are correct.

……………………………….. ………………….

(Dr. H. M. N. Dilum Bandara) Date

Signature of the Supervisor

………………………………… …………………..

(Dr. Indika Perera) Date

Signature of the Co-Supervisor

COPYRIGHT STATEMENT

ABSTRACT

Agile Software Development (ASD) process has become the preferred method for modern software development. ASD emphasises iterative and incremental development, where both the requirements and solutions evolve through the collaboration between self-organizing, cross-functional teams. While software metrics help to assess the status of a project, process, product, and resources, very little work exists on the use and suitability of software metric in the context of ASD. Though many established metrics are used in the Traditional Software Development (TSD) process, they cannot be directly applied to the ASD process due to its iterative and incremental development process that is willing to incorporate changes throughout the process. Therefore, it is imperative to understand what metrics are useful in the context of ASD process. The objective of this research is to explore suitable metrics for the ASD process, use of those metrics in practice, and the perceived benefits.

The research was conducted using a qualitative and descriptive analysis method. Initial interviews were conducted to gather information about the metrics currently used in the ASD process and to develop the online questionnaire for the survey. The online questionnaire was shared between 26 different organizations that use ASD methodologies. Five of those companies have projects only based on the ASD process while others have projects on both the TSD and ASD processes. Responses were then analysed and a set of findings was derived. Follow-up interviews were conducted with the subject matter experts for their feedback on the overall research findings, to further understand, verify, and clarify the findings.

The study identified a set of metrics that can be used in the ASD process. The top five metrics include *Delivery on time*, *Work capacity*, *Unit test coverage for the developed code*, *Percentage of adopted work*, and *Bug correction time from "new" to "closed" state*. *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, and *Open defect severity index* metrics can be used in both the TSD and ASD processes. Though *Requirement Clarity Index*, *Defect density*, and *Defect removal efficiency* metrics were heavily used in the TSD process, they were less used in the ASD process. *Thumbs-up rule*, *None compliance index*, and *Top hill view* metrics were identified at the face-to-face interview sessions conducted with the industry experts. *Work capacity*, *Percentage of adopted work*, *Sprint-level effort burndown*, *Velocity*, *Percentage of found work*, and *Focus factor* were the Agile-specific metrics rated as the most used. When talked about tools usage, JIRA/Greenhoper was the most used tool whereas the Microsoft Excel was the second most popular tool. Companies fully into Agile practices mostly used specialized tools like JIRA/Greenhopper. Whereas in the other companies, Microsoft Excel with other supportive tools was the most popular tool.

Use of metrics in an Agile project helps to track the project progress, monitor the quality aspect of the project and it helps the team to forecast and manage the project better. *Unit test coverage for the developed code, Bug correction time from "new" to "closed" state*, and *Open defect severity index* metrics measure the quality of the product. *Work capacity*, *Percentage of adopted work*, *Velocity*, *Sprint-level effort burndown,* and *Percentage of found work* metrics measure the productivity of the team. *Delivery on time* and *Focus factors* metric measure the predictability of the project. Out of the metrics mentioned above, JIRA/Greenhopper tool measure five metrics (*Work capacity*, *Adopted work*, *Sprint-level effort burn down*, *Velocity*, *Found work*) directly and four metrics (*Focus factor*, *Open defect severity index*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*) indirectly. Therefore, JIRA/Greenhopper is an ideal tool to measure the ASD process along with supportive tools.

Keywords: Agile Software Development process, Software metrics, Traditional Software Development process, Scrum Development process, JIRA/Greehopper

ACKNOWLEDGEMENT

I wish to express my deep gratitude to all those who have helped me in successfully completing my research study on “**Use Of Software Metrics In The Agile Software Development Process**”.
First and foremost, I wish to thank my research supervisor Dr. Dilum Bandara for his guidance and unwavering support, which was a big contribution towards the advancement of the thesis. Dr. Chandana Gamage, Head of Computer Science and Engineering, Mrs. Vishaka Nanayakkara, Senior Lecturer and Dr. Malaka J. Walpola, Course Coordinator of MBA in IT program provided me with advice, especially by analysing the project proposal over and over again and at the progress review presentations. I am thankful to Dr. Indika Perera for his guidance. In addition, I am grateful for the diversified knowledge imparted to me, with great skills and concern by all my lecturers who helped me in successfully completing the thesis.

A special word of thanks goes out to Mr. Ajith Gamage, Senior Manager of Process at Virtusa; Mr. Harshana K. Kuruppu, Manager Delivery at Virtusa;Mr. Shamira Dias, Delivery Manager of Exilesoft; Mrs. Janani Liyanage, Certified Scrum Master and Project Manager at IFS; Mr. Rajeev Jayawardena, Program Manager and Certified Scrum Master at Pearson Lanka and Mr. Chandima Vithanage, Test Engineer at 99x technology, who took time off their tight schedules to help me in providing information with regard to metrics usage in scrum adoption in Sri Lanka.

Finally, I thank Mrs. Kalahari Samaraweera, for the help and support given throughout the thesis, which was precious. I gratefully remember the protection and love of my parents. Last but certainly not the least, I thank my colleagues at MBA batch and all my other friends, family and colleagues whom I do not have the capacity to name individually here.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

Abbreviation	Description
ASD	Agile Software Development
BPO	Business Process Outsourcing
CCM	Cyclomatic Complexity Metric
DSDM	Dynamic Software Development Method
EDB	Export Development Board
HCM	Halstead Complexity Metric
ISO/IEC	International Organization for Standardization and the International Electrotechnical Commission.
IT	Information Technology
KLOC	1000 lines of code
LOC	Lines of Codes
PO	Product Owner
SAL	Scrum Adherence Level
SDLC	Software Development Life Cycle
SEA	Software Exporters Association
SLASI	Sri Lanka Association for Software Industry
SLASSCOM	Sri Lanka Association of Software and Service Companies
SLOC	Source Line Of Code
SM	Scrum Master
SPSS	Statistical Package for the Social Sciences
SQuaRE	Software product Quality Requirements and Evaluation
TSD	Traditional Software Development
TVI+	Targeted Value Increase
XP, XP2	Extreme Programming

1. INTRODUCTION

1.1 Background

Software metrics are used to measure the software development project progress and to further understand the behaviour of varying aspects of the code base. These metrics can be used to gain a wide variety of information about the quality of the product delivered to the customer, progress of a software project, cost estimation and size/complexity of software systems. Measurements need to be closely monitored when the requirements of a software system change frequently. *Changing requirements* is one of the major problem arising in the software development process. Agile Software Development (ASD) process successfully handles the reality of change. The departure from the Traditional Software Development (TSD) process to ASD process was a significant departure from the heavyweight, document-driven software development methodologies into lightweight, human-centric software development methodology. Therefore, while selecting software metrics to measure ASD-based projects, it should be handled with a deeper understanding about the differences between the TSD process, ASD process and those metrics. According to an online survey conducted by VersionOne (2012), 84% of respondent's projects were already based on the ASD process. In year 2012, 82% of respondents had planned to implement the ASD process in projects, which was 59% in the year 2011.This shows that ASD process is becoming more popular in the Information Technology (IT) industry.

Software metrics are especially important in the ASD process because it is an iterative process. If measurements occurred, those metrics can easily be used to improve the next iteration. Many well-established software metrics are available for the TSD process. However, not all of them can be applied directly to the ASD process. While some of the traditional metrics can be adapted to use with the ASD process, some are not suitable for the ASD process at all. This is because of the iterative and incremental development process. Moreover, ASD process always willing to incorporate changes throughout the process.

1.2 Motivation

A limited number of surveys and published research had talked about different angles of the ASD process. Versionone (2012) annual survey and the Xabia group (2012) annual Agile survey are two of surveys focusing on the ASD process. The Versionone (2012) survey had addresses following areas throughout the survey, those are, personal experiences of the software development professionals working in the ASD process, enhancement of the Agile development, use of Agile methodology, use of Agile techniques, specific tools used within the ASD process, benefits gained while conducting software product development projects in an Agile, and the reason for adopting Agile. However, the survey does not cover what software metrics are applicable to the ASD process and their appropriateness.

According to the VersionOne (2012), Agile adopted companies that practice Agile across five or more teams have increased from 33% in year 2011 to 48% in 2012. Given that, adaptation into the ASD process is rapidly increasing, it is imperative to identify a set of metrics that is more suitable for the ASD process. When we considered the fundamental differences in the two development paradigms, many metrics proposed for the TSD process cannot directly be used within the ASD process. However, due to the familiarity with the TSD process, many developers and project managers are inclined to use the same metrics for the ASD process. Such attempts lead to wrong interpretation of the progress of the software development project and quality of the deliverables to the customer. This could also lead frustrations among the team members. Although many companies are trying to adapt the ASD process, most of them are not concerned about the metrics that they are going to use within a project. While much literature discusses about the various aspects of metrics used in the TSD process, very little attention is given to the metrics used in the ASD process. Metrics are essential to manage a project successfully and productively. Therefore, if the significant metrics within the ASD process and their benefits can be clearly identified, organizations may not be reluctant to use those metrics. Hence, there is a need for a formal evaluation about using software metrics in the ASD process. Thesis address this gap by identifying the metrics that are more appropriate for projects based on the ASD process and understanding their benefits.

1.3 Research Scope

The research focuses on the companies, whose core business is the software development, IT services and consulting as well as managing at least one project based on ASD methodology. Some companies have projects based on both ASD and TSD processes.

1.4 Problem Statement

There are well-established software metrics for the projects based on the TSD process. Due to the lack of understanding about the metrics to be used within the ASD process and their perceived benefits, as well as the prior experiences with the metrics for the TSD process, software development professionals tend to use software metrics practiced in the TSD process also within the ASD process. However, when considering the fundamental differences between ASD and TSD processes, software professionals cannot expect good results by using the same metrics in both processes. There should be Agile-specific metrics. Otherwise the metrics used in the TSD process needs to be changed accordingly. Therefore, the objectives of this study are to identify a set of software metrics suitable for the ASD process and its usage in the projects based on the ASD process. Further, to identify the benefits of use of software metric in projects based on the ASD process. Moreover, company-specific and project-specific metrics are also explored. Therefore, the problem statement can be framed as:

“What are the important software metrics and their usage in projects based on the Agile Software Development process?”

1.5 Research Objectives

Objectives of the study are to discover the software metrics used by the software industry while developing software using the ASD process and to understand the benefits of those metrics. The following objectives are to be achieved at the completion of the study:

- Identify a set of software metrics suitable for the ASD process
- Identify the software metric usage in projects based on the ASD process
- Identify the benefits of use of software metric in projects based on the ASD process

1.6 Research Significance

This research identifies a set of metrics that is recommended to be used in projects based on the ASD process. 10 metrics were identified, that were used by 75% or more companies. Among them, six metrics are Agile specific. These include *Work capacity*, *Adopted work*, *Sprint-level effort burndown*, *Velocity*, *Found Work*, *Focus Factor*. Remaining metrics were used in the TSD process as well. Those are *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state* and *Open defect severity index*. *Delivery on time* metric got the highest value for the frequency of usage and *Targeted value increase* (*TVI+*) got the lowest value for the frequency of usage. *Defect Density* metric is the only metric, which showed a significant change at companies with projects based only on the ASD process and on companies with projects based on both ASD process and TSD process.

When the most used ten metrics are categorised based on their measurements, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, and *Open Defect Severity Index* metrics are suitable to measure the quality of the product. *Work Capacity, Percentage of adopted work, Velocity, Sprint-level effort burndown, Percentage of found work* metrics measure the productivity of the team. *Delivery on time* and *Focus factor* metrics are suitable to measure the predictability of the project.

JIRA/Greenhoper is the most used Agile-specific tool for the ASD process. JIRA/Greenhopper tool is most popular among the companies who have projects based only on the ASD process. At the same time, the companies also use several other supportive tools. In ASD and TSD process companies, most used tool is the Microsoft Excel. They also use other supportive tools.

1.7 Outline

Chapter 2 provides a detailed overview of the related work on software metrics in ASD and TSD processes. It discusses about the ASD process, software metrics used in TSD and ASD processes, methodologies and relevance metrics. Chapter further discusses about the importance of quality in a software product. The research methodology is presented in Chapter 3. It defines the steps that were followed while conducting the research, research question, and population and sample population. Findings from the surveys and interviews are presented in Chapter 4. Finally, the Chapter 5 discusses the research findings, recommendations and future work. Online survey questionnaire, feedback for preliminary questionnaire, follow-up interview questions, and metric description are given in Appendix A, B, C, and D, respectively.

2. LITERATURE REVIEW

Software solutions need to be delivered in the shortest possible time due to the technological evolution, market demands, rival's products and competitive pressure of today's business. Software development industry had started to use the Agile Software Development (ASD) process and it has become more and more popular at present (Versionone, 2012). One reason for the popularity is the benefits it gives over the TSD process such as the ability to absorb frequently changing requirements and frequent deliverables to the customer.

The chapter describes about the software metric usage at both the ASD and TSD processes. In Section 2.1 describes about the status of the Software Development Life Cycle (SDLC). In addition, it highlights the importance of a new SDLC process in software development industry over the TSD process. Then Section 2.2 presents a brief introduction to the software metrics and its behaviour in both the TSD process and ASD process. Software metrics impact on software product quality is discussed in Section 2.3.

2.1 Software Development Process

SDLC is the development and the maintenance process of a software system, which contains models and frameworks to plan and maintain the entire development process (Mordal et al., 2012). It includes phases such as project initiation, design analysis, system design specification, programming and testing, installation and maintenance, and destruction. Software professionals are familiar with software development models designed and used in SDLC. Each model has its own way of handling the software development process. Waterfall model, iterative model, spiral model, V-model, Big Bang Model, Rapid Application Development Model and Prototyping Model were one of popular SDLC models (Purcell, 2007).

To meet rapidly changing user requirements, product owner expects the continuous improvement process to enhance the product features. TSD process was not sufficient to satisfy the thirst of customers for their frequently changing requirements in the fast moving globalized world. Therefore, industry, academic and standard organizations have been looking for advanced SDLC technique. The advanced SDLC method should also be able to deliver the product faster to the customer with high quality, but cost effectively.

2.1.1 Agile Software Development Process

ASD process is an iterative and incremental software development process. Hyper-productive and self-organizing teams play the key role in the process. Deliver a set of working software to the client after every iteration was the target for a team (Mannila, 2013). Likewise, ASD process is the best solution for today's market. It gives answers to many questions aroused due to the advancement of the technology in a rapidly changing environment. ASD methodologies are more focused on customer satisfaction, project success and risk reduction (Rico, 2008).

The basic idea of the ASD process is clearly defined in the "Agile Manifesto" with the introduction of four basic agile values, which are listed below (Beck et al., 2001):

1. *Individuals and interactions* over process and tools
2. *Working software* over comprehensive documentation
3. *Customer collaboration* over contract negotiation
4. *Responding to change* over following a plan

Seventeen advocates, who jointly documented the Agile Manifesto in year 2001, had stated that they value the items on the left more than the items on the right (Beck et al., 2001). There are 12 principles (Beck et al., 2001) which follow those four values on the left at the Agile Manifesto. "Individuals and interactions over Process and tools" as the first value emphasized that it value more face-to-face communication among team members, stakeholders and developers than limiting it to e-mails. There was no specific person to command for agile team members. They are expected to be a self-organized and hyper-productive team. This leads to effective teamwork and better chances for innovation and job satisfaction. Second value "Working software over Comprehensive documentation" described that team worked in a product, which developed iteratively. At the end of each iteration, customer has a chance to play with the working software. The comments and feedback with lessons learned, to incorporate at the next iteration. Free from comprehensive documentation, therefore, able to maintain whenever required, depending on the project. The third value "Customer collaboration over the contract negotiation" gives the idea of, this is not just developing what you want, however highly satisfy you. Work was not just a contract, but the close interaction with the customer and it helps to enrich his satisfaction. Finally the fourth value "Responding to change over following a plan" was the most important factor when deals with customer requirements, which always change in this dynamic world. The customer wants the best fit to the current industry to compete with others. ASD process was the best fit since it always welcome change environment.

There are dozens of agile development methodologies and customized frameworks under the umbrella of ASD process. Those methodologies help to strip away the heaviness associated in the TSD methodologies (Dyba˚ & Dingsøyr, 2008). ASD process techniques align with any situations where other methodologies cannot be used. Such as, it fitted in small projects with two to 30 members group, or in a continuous interaction with customer and software developers, or a product with frequently changing requirements and in a complex situation. ASD process is not filled up with heavy documentations as in the TSD process (Fathi and Morovat, 2013), but it allows to having required documentation which can differ from project to project, depend on the project requirements. Flexibility and quick response to requirement change, Communication and requirement management were the basic features in the ASD process (Siakas et al., 2005; M T Sletholt, 2011). Active customer involvement was one of the very important features in Agile software development methodology compared to rest of the SDLC methodologies (Oza and Korkala, 2012).

2.1.2 Agile vs. Traditional Software Development Process

Waterfall model was relatively simple to understand. Within the process, phases will not be overlapped. At the end of one phase only, the next phase will start (see Figure 2-1). However, as seen in Figure 2-2, all the phases in the ASD process are based at the same time, iteratively. It absorbs feedback and encourages frequent inspection and adaptation. In TSD process, completed software will be delivered mostly at the end of the process. However, in ASD process development done over the working software and features will be shipped as deliverables within a minor period. TSD process has high risk and high probability of failing the project, since after requirements gathering the customer will not get involved with the product, till the product is handed over, other than if there is no any specific requirement gathering. If customer wants any changes towards the final stage of the project, the cost will be high, since it needs more effort and time to consume. Code change can be huge and effort will be high in cost (see Table 2.1).

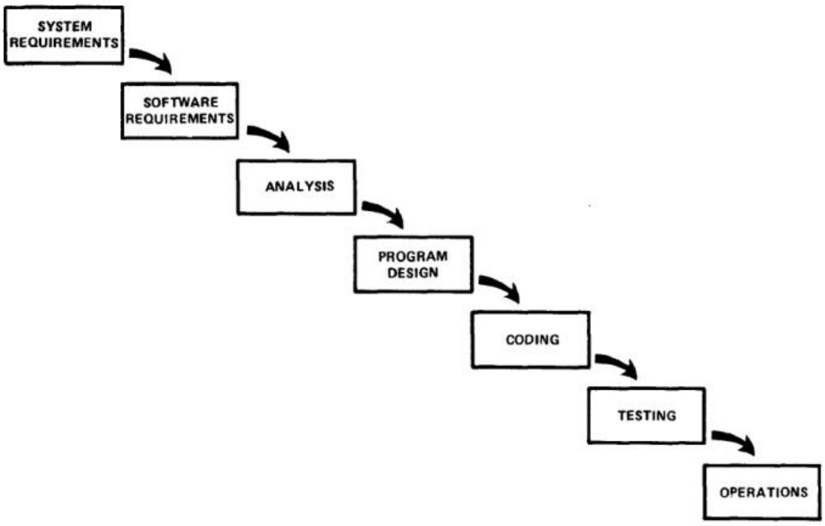


Figure 2-1**:** The linear workflows of the Waterfall methodology. Source: Royce, 1970.

However, the ASD process is more reliable than other Software Development processes, since it always welcomes requirement changes at any time during the project lifecycle (Siakas et al., 2005). Time and effort spend for feature/code changes in TSD method increased with the time but in ASD process, it is almost consistent when time increased (Figure 2-3).

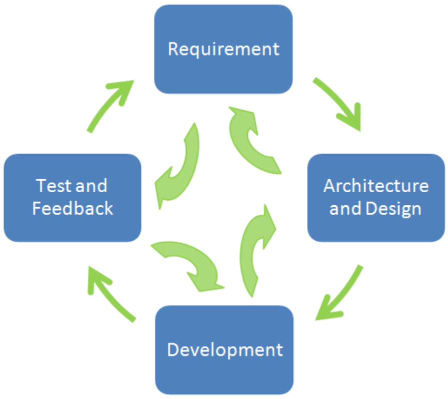
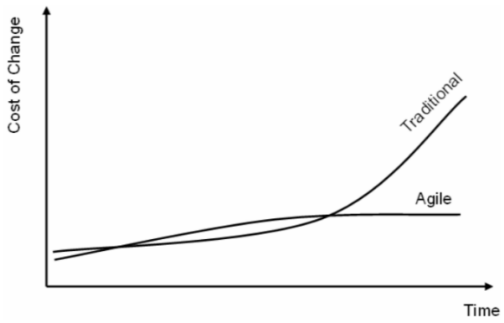


Figure 2-2**:** Agile Software Development process. Source: Rising, 2009.

ASD method does not create any detailed plans as in TSD process. However, it concentrates more to satisfy customer requirement, which helps to maintain sustainability, and improve customers' competitive advantage (Siakas et al. 2005). For scientific software development projects, it is more valuable to use an agile approach, especially for smaller sized teams and projects (M. T. Sletholt, 2011). ASD process value increased due to its ability to absorb frequent requirement change (Kunz et al., 2008). Kayes et al., (2013) have found the success rate of ASD projects were three times more than non-agile projects. Oza and Korkala (2012) reveal that the agile adaptation had made a significant impact to product developments and coordination.



.

Figure 2-3: Cost of change compared to development method. Source: Reiner R. Dumke, 2008, p.1.

Table 2.1: Deviation from Traditional to Agile method.Source**:** Dyba˚ & Dingsøyr, 2008, pp. 836.

Feature	Agile Method	Traditional Method
User requirement	Iterative acquisition	Detailed user requirements are well-defined before coding/ implementation
Rework cost	Low	high
Development direction	Readily changeable	Fixed
Testing	On every iteration	After coding phase completed
Customer involvement	High	low
Extra quality required for developers	Interpersonal skills and basic business knowledge	Nothing in particular
Suitable Project scale	low to medium-scaled	Large-scaled

Until late 2011, there was no publication for measurement in ASD methods. Although many well-known practices and standards were available for traditional methods, the first measurement study on ASD methods appeared only on 2011 (Javdani et al., 2012). According to the Manifesto and its subsequent notes, origin of different agile methods exists based on the areas such as people oriented, embracing changes, focusing on product, simplicity, self-organized team and fast delivery (Javdani et al., 2012). ASD methodologies output an effective, successful software product compared to cost of heavy quality systems (Siakas et al., 2005). Table 2.2

describes about the most referenced agile development methods used in the industry (Dyba˚ & Dingsøyr, 2008).

Table 2.2: Agile development methods.Source: Dyba˚ & Dingsøyr, 2008, p.835.

#	Agile Method	Description
1	Crystal methodologies	A family of methods for co-located teams of deferent sizes and criticality: Clear, Yellow, Orange, Red, Blue. The most agile method, Crystal Clear, focuses on communication in small teams developing software that is not life-critical. Clear development has seven characteristics: frequent delivery, reflective improvement, osmotic communication, and personal safety, and focus, easy access to expert users, and requirements for the technical environment.
2	Dynamic software development method (DSDM)	Divides projects in three phases: pre-project, project life cycle, and post project. Nine principles underlie DSDM: user involvement, empowering the project team, frequent delivery, addressing current business needs, iterative and incremental development, allow for reversing changes, high-level scope being fixed before project starts, testing throughout the lifecycle, and efficient and effective communication.
3	Feature-driven development	Combines model-driven and agile development with emphasis on initial object model, division of work in features, and iterative design for each feature. Claims to be suitable for the development of critical systems. An iteration of a feature consists of two phases: design and development
4	Lean software development	An adaptation of principles from lean production and, in particular, the Toyota production system to software development. Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.
5	Scrum	Focuses on project management in situations where it is difficult to plan ahead, with mechanisms for ''empirical process control''; where feedback loops constitute the core element. Software is developed by a self-organizing team in increments (called ''sprints''), starting with planning and ending with a review. Features to be implemented in the system are registered in a backlog. Then, the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. One team member, the scrum master, is in charge of solving problems that stop the team from working effectively.
6	Extreme programming (XP; XP2)	Focuses on best practice for development. Consists of twelve practices: the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers, and coding standards. The revised ''XP2'' consists of the following ''primary practices'': sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming, and incremental design. There are also 11 ''corollary practices''

2.1.3 Scrum Development Framework

Out of the ASD methodologies described in Table 2.2, the Scrum Development framework is the most popular, commonly used and most adopted (52%) ASD methodology (Kayes et al., 2013; Gustafsson, 2011). Where the framework was derived from the best practices of successful companies like Fujixerox, Honda, Canon and Toyota (Sutherland, 2009). Scrum framework (Figure 2-4) allows average teams to self-organize into hyper-productive state with the application of simple constraints (Downey and Sutherland, 2013).

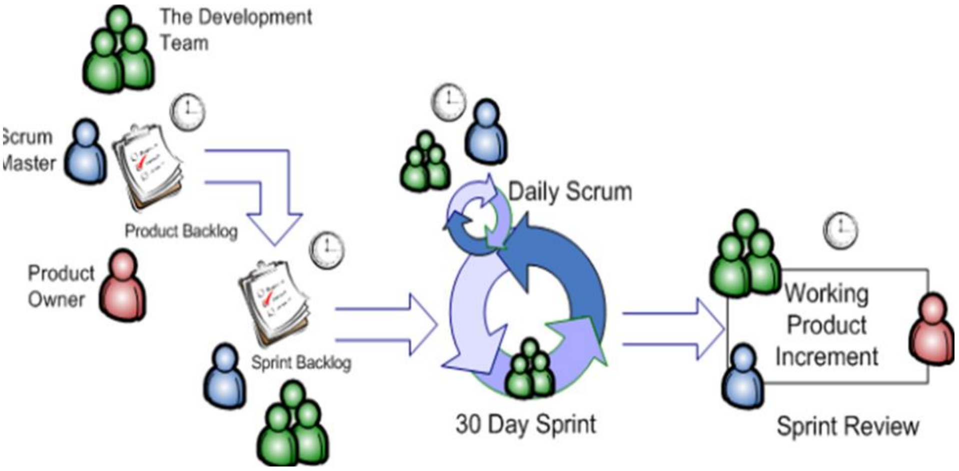


Figure 2-4: Scrum Framework. Source: Mountain Goat Software, 2013.

The word Scrum derived from the Rugby sport which denotes the spirit of the teamwork and self managed members in a cross functional team, working together to achieve the product targets (Zoysa, 2011). Yahoo! emphasized as the successful large enterprise adopted for Agile and Scrum practices (Benefield, 2008). Scrum increased the productivity by 5-10 times over industry average and many teams had achieved it (Sutherland, 2009). There were three distributed scrum modules (Figure 2-5) commonly used in practice, namely, Isolate Scrums, Distributed Scrums of Scrums and Fully Distributed Scrums used specially to win the promised benefits of outsourcing (Sutherland, 2009).

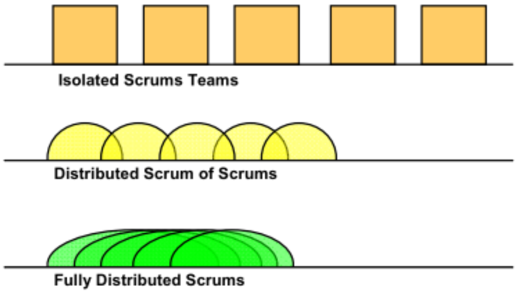


Figure 2-5 : Distributed Scrum Team Strategies

Source: Sutherland, 2009, p.2.

2.2 Software Metrics

Measuring software development project and working product quality isimportant. Measurement makes easy to understand the situation in a better way and assist in project management (Downey and Sutherland, 2013).It is also important to have a real understanding about what we are trying to measure, for the growth of ideal metrics. Metrics provide visibility and insight about what we do and how well we did it (Eeger, 2008). To measure a software quality or software development project progress, there should be an identified set of metrics. When it comes to the selection of metric, it should assist for better results. Therefore, the good software metric should be simple, precisely definable and attainable the objective at a reasonable cost and should be measurable what it is intended to. Other than those characteristics, metrics should be able to assist in developing models, which are efficient in predicting process of the product spectrum (Rawat et al., 2012). Metrics helps to gauge the quality of the software development product, standard of software testing effort further the progress. "Software metrics are a valuable entity in the entire software life cycle. They provide measurement for the software development, including software requirement documents, designs, programs and tests" (Rawat et al., 2012).

In project management, software metrics are essential. Because project metrics helps to get an understanding about the project status, it tracks the project for numerical ratings to quantify some characteristics or attributes of a software entity. There was a set of metrics used in the TSD process. Research papers, standard institutions, books had talked a lot about metrics used in TSD process. When ASD process was introduced to the software development industry, relevant metrics were not stated at the beginning. When it started to spread rapidly, needfor metrics also rose. TSD process was planned driven and ASD process was result driven (Oza and Korkala, 2012). Therefore, the metrics used in TSD process cannot be directly used in ASD process (Reiner R. Dumke, 2008), (Gustafsson, 2011), (Jacdani et al., 2012). In contrast to the TSD process, at ASD process only a simple set of metrics may assist to maintain safe and consistent growth in a hyper-productive Agile teams (Downey and Sutherland, 2013).

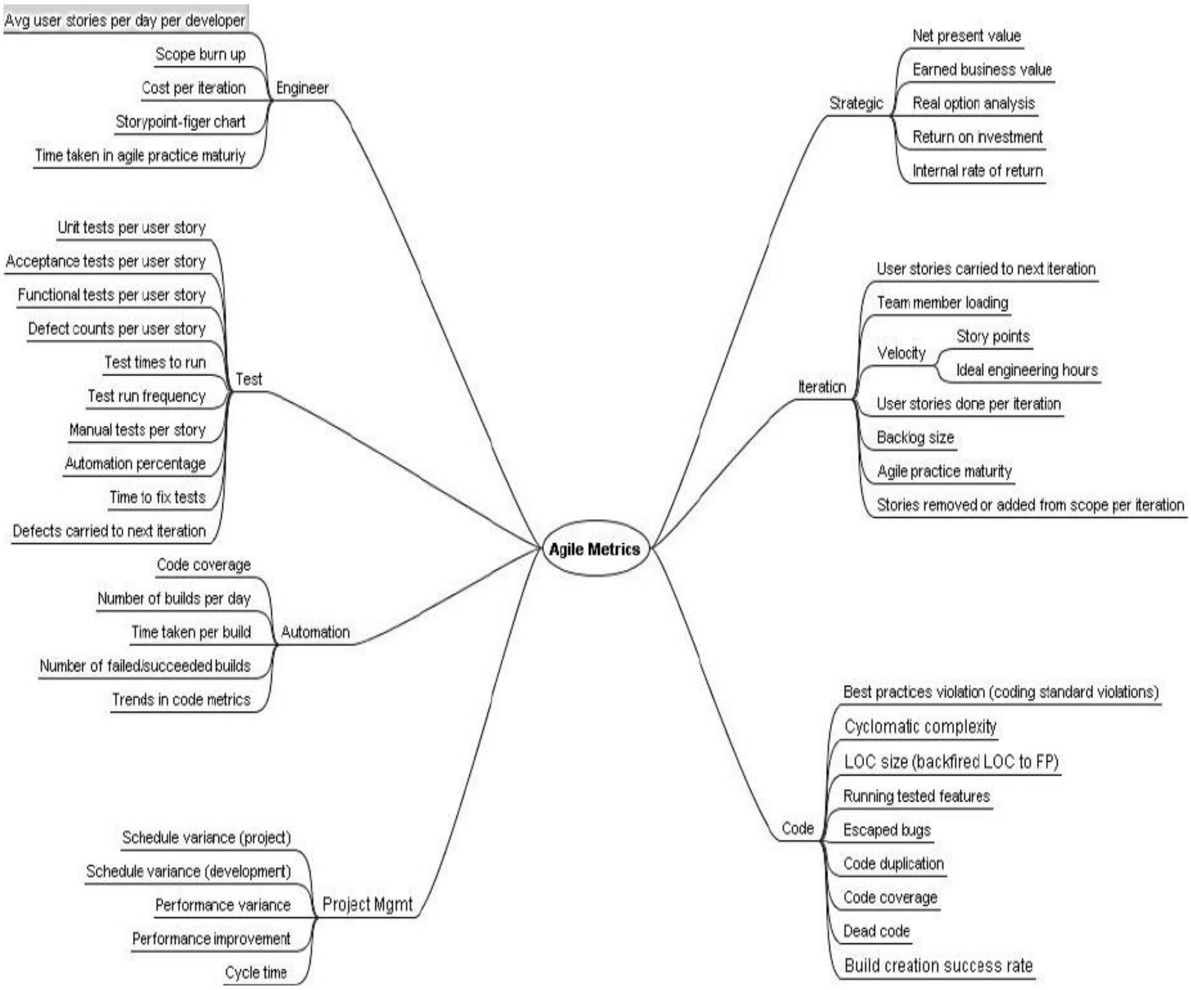


Figure 2-6: Agile metric mind map shows agile metrics used in different areas

Source: Oza and Korkala, 2012.

2.2.1 Software Metrics in Traditional Software Development Process

In TSD process, software metrics can be divided into 3 sections, as, product metrics, process metrics and project metrics. Furthermore, metrics can be used at process monitoring and improvement, product improvement, quality control and for software estimations (Siakas et al., 2005; Kitchenham, 1996). As mentioned above, in Kan (2002) also classified software metrics at TSD process into three categories as product, process and project metrics. Product metrics define the characteristics of the product, size, complexity, design aspects, performance and quality. Process metrics measure the process of software development, maintenance and testing. Project metrics measure various project characteristics, e.g., time and productivity (Rawat et al., 2012).

The purpose of measuring software complexity of a developed code was to reduce the complexity in software and further to reduce the software cost. However, this metric would no longer used in the ASD process as described in Figure 2-3. In TSD process, Lines of Codes (LOC), Halstead Complexity Metric (HCM), and Cyclomatic Complexity Metric (CCM) metrics were used to measure the software complexity. McCabe's Cyclomatic complexity metric, defined in 1976, measure the number of independent paths through a software module. Defect Removal Efficiency metric was one of the very important measurements of software quality (Jones, 2008). Defect Density metric was measured as defects per function points or defects per KLOC (1000 lines of code) to measure the software reliability (Malaiya, 1998). Source line of code or SLOC metric, Function point metric and Object oriented metrics (Rawat et al., 2012) were practiced in the TSD process. Duplicated code metric and dead code metric to measure duplicate code and never used code. Database metrics to measure the quality of the database structure (Rentrop, 2006).

2.2.2 Software Metrics in Agile Software Development Process

Oza and Korkala (2012) classified the metrics used in the ASD process, as Code level (e.g., Running tested features, Leffingwell’s iteration and release perspectives, and code quality and design metrics), Productivity/effort level (e.g., burn-down charts and project size units) and Economic metrics (e.g., earned business value and break-even point). From the Code level metrics its try to provide visibility into code quality and from Productivity and Economic metrics to support the decision making process. Still authors were not satisfied about the metrics and their classification. As shown in Figure 2-6 authors again classified the metrics across seven categories. Though it represents many metrics, depending on the metric selection by the company, number of metric practiced differ from company to company (Oza and Korkala, 2012). Good metrics in ASD process may lead to enhance team performance. Therefore, by using software metrics, team can be managed to optimize the work. In ASD process selected metrics should not be a burden to the team members to carry on the work they are assigned. It should be simple and easy to maintain (Downey and Sutherland, 2013). Downey and Sutherland (2013) identified ten essential metrics. He had mentioned that, those were meaningful and further can be used by management fordecision-making. Ten essential metrics were *Velocity*, *Work Capacity*, *Focus Factor*, *Percentage of Adopted Work*, *Percentage of Found Work*, *Accuracy of Forecast*, *Targeted Value Increase* (*TVI+*), *Success at Scale*, and *Win/Loss Record* (see Appendix E). Downey and Sutherland (2013) had highlighted eight essential metrics by avoiding the *Success at scale*metric and *Win/Loss record* metrics out of the list mentioned above. Manila (2013) had come up with a set of customized metrics by doing a survey for a selected organization. These metrics were *Fault correction time to "Closed" state*, *Delivery on time*, *Technical debt*, *Unit test coverage for the developed code*, *Smoke test cycle time*, *Regression test cycle time*, *Future measurement - definition of done check list*. That organization had recently taken the step to move into ASD process-Scrum development process from TSD process-waterfall development process. Therefore, it seems they were still using some customized metrics which were commonly used in TSD process such as Unit test coverage for the developed code, Smoke test cycle time, Regression test cycle time. Downey and Sutherland (2013) and Manila (2013) had introduced different sets of metrics. Gustafsson (2011) devised metrics into five categories as Quality, predictability, Business value, Lean and Cost. Under quality, it described about three metrics in each, Defect count, Technical dept, Faults-slip-though and Lead-time, Work in progress, Queues respectively. Average cost per function metric measure under cost attribute and predictability and value measure using velocity, Running automated test cases and Customer satisfaction survey, Business value delivered respectively.

2.3 Software Development Product Quality

Product quality in software has become highly important. For example,Rawat et al., (2012) state that "Given the penetration of computer code into everyday objects like washing machines, automobiles, refrigerators, toys and even things like the mars rover, any system be in a large one or a small system running embedded IC technology, ensuring the highest levels of software quality is paramount". Therefore, software metrics are becoming part of the software development fabric. It is really in need to understand that, the quality of the software we are building to achieve the expectations (Eeger, 2008). ISO/IEC 9126:1991 defines a *software quality metric* as a *quantitative scale and method that can be used to determine the value*, *a feature takes for a specific software product* (Emam, 2002). It is better to focus on improving the product quality, since it gives the higher productivity (Rawat et al., 2012). Emam, (2002) stated that “Software product metrics play a central role in software engineering, and their proper validation will ensure that there is a compelling case for their use in practice.” Software product metrics validation can be done as follows:

1. The product metric measures what it purpose to measure. For example, that a coupling metric is really measuring coupling.
2. The product metric is associated with some important external metric (such as measures of maintainability or reliability).
3. The product metric is an improvement over existing product metrics. An improvement can mean, for example, that it is easier to collect the metric or that it is a better predictor of faults (Emam, 2002)

ISO published the first international consensus for the quality characteristics naming “software product evaluation quality characteristics and guidelines for their use” in 1991 with the purpose of standardizing the software product quality measurement process. Then they continuously expanded it with some changes from 2001-2004

(ISO 9126: 1991). These standards specifically for the TSD process, not for ASD process. It is needed to identify new set of metrics, thresholds and measurement artefacts, which are suited for agile software developments (Reiner R. Dumke, 2008). Software Product Quality Requirements and Evaluation (SQuaRE - ISO 25000) had introduced by ISO as next generation software product quality standards, which will replace the current ISO 9126 and ISO, 14598 series of standards (Abran et al., 2005). The new ISO 25000 serious consist of five standards, such as; Measurement reference model and guide (ISO 25020), Measurement primitives (ISO 25021), Measurement of internal quality (ISO 25022), Measurement of external quality (ISO 25023), Measurement of quality in use (ISO 25024). ISO 9126:1991 six software product quality categories were replaced by eight quality characteristics of ISO 25000, such as Functional suitability, Reliability, Security, Compatibility, Performance efficiency, Maintainability and Portability.

2.4 Summary

As mentioned in the literature presented above metrics play vital role in the SDLC. TSD process has well-established set of software metrics and had benefitted out of those. However, when it comes to the ASD process, industry still does not have a clear idea about the metrics suitable for the ASD process and which metric would contribute to a given project. Therefore, identifying a set of metrics for the ASD processwill help to reduce the burden of the team members arise due to the use of metrics, which are not aligned with the process. Quality, schedule (time) and cost are key factors for any project. Therefore, use of suitable metrics helps to enhance the project quality, productivity of the team, and for future prediction purposes.

3. RESEARCH METHODOLOGY

The chapter describes the steps and procedures practiced during the data gathering and analysis of the research. Findings from the literature review played an important role at this stage, because it helped to lay the foundation in the development of the research methodology. Section 3.1 describes the research problem. Research methodology is described in Section 3.2. Section 3.3 presents details about the data population and sample population used to conduct the research.

3.1 Research Problem

In-depth analysis conducted for many software metrics exists in the Traditional Software Development (TSD) process by both the industry and academia. While doing that their usage and benefits also considered. However, for the Agile Software Development (ASD) process, very little work exists on the use and suitability of software metrics. Therefore, objective of this research is to fill that gap by identifying a set of metrics suitable for the ASD process. Software metrics help to evaluate the status of a project, process, product and resources based on both the TSD and ASD processes. Existing software metrics used in the TSD process cannot be directly applied to projects based on the ASD process, because it absorbs requirement changes at its short iteration periods. Projects based on the ASD process maintain closer relationship with the client, which is not visible in the TSD process. ASD process is more human-centric. Therefore, Agile-specific metric can increase the accuracy of the measurements and the information derived out of those measurements.

Specific objectives of this research are as follows:

- Identify a set of software metrics suitable for the ASD process
- Identify the software metric usage in projects based on the ASD process
- Identify the benefits of use of software metric in projects based on the ASD process

3.2 Research Method

The research was conducted based on the qualitative and descriptive data analysis methodology. The initial objective and the focus of the research were to identify the benefits of the metric usage in the ASD process. However, after several discussions with the professionals working in the companies that use the ASD process, it was identified that the companies are still looking for a more suitable set of metrics to be used within a project based on the ASD process. Though there are many metrics available for the projects based on the TSD process, only a limited number of metrics were used in the ASD process. Therefore, the research focus was diverted to identify a more suitable set of metrics and its usage in the ASD process.

As illustrated in Figure 3.1, at the stage 1 of the process, analysis of literature helped to identify a set of metrics used in both TSD process and ASD processes. Face-to-face interviews were then conducted with four professionals (from four different reputed IT companies in Sri Lanka) who were involved in projects based on ASD process. One of the companies is in to service delivery while the other three are product-base companies. One company has projects based on both the TSD and ASD process and the other three companies have projects based only on ASD process. Face-to-face interviews were conducted with four professionals, who had more than four years of experience. Professionals include two Quality Assurance (QA) Leads, one Associate Quality Assurance Manager, and a Delivery Manager. Interviews were conducted with the help of a questionnaire, given in Appendix A, to get an understanding about the practical use of metric. This questionnaire was shared with the interviewees before conducting the interviews. The questionnaire was prepared based on the information gathered from the literature survey. Associating the collected data during the face-to-face interviews and knowledge absorbed from the literature survey, online survey questionnaire was developed. The feedback comments, given in Appendix B, was gathered by sharing the questionnaire with three QA leads working in three different companies who have more than two years of experience in the ASD process. Those comments were incorporated with the online survey questionnaire given in the Appendix C. Then, the online survey questionnaire was shared with 26 identified potential software development companies (listed in Table 3.2), who have projects based on the ASD process. Accepted responses were analysed and the findings were summarised accordingly. Follow-up interview questionnaire, given in the Appendix D, was developed upon the summarised research survey finding, to engrave the research findings and to get clarify the doubtful situations. Face-to-face interviews were conducted using the follow-up interview questionnaire along with the research survey findings, to get the subject matter experts' judgements.

3.3 Data Collection

3.3.1 Population and Sample Selection

Targeted companies were any company, which had at least one project based on the ASD process. Since Sri Lanka is a reputed Business Process Outsourcing (BPO) destination, many global IT companies have placed their branches in Sri Lanka. AT Kearney ranked Sri Lanka among the top 50 Global Outsourcing destinations and among top 20 emerging cities by Global Services Magazine (Sri Lanka the hub of Asia, 2013).Research data were collected from the software development professionals working on the projects based on the ASD process. Target population and the sample population for this research are present at the Table 3.1.

Table 3.1: Population and sampling population.

Population	Software development projects based on ASD process
Sample Population	Product or service oriented small, medium, or large scale software development organizations that have projects based on ASD process, where the organization is a member of one of the following: Sri Lanka Association of Software and Service Companies (SLASSCOM), Sri Lanka Association for Software Industry (SLASI), Software Exporters Association (SEA), and Export Development Board (EDB)

Snowball (Mack, et al., 2005) sampling method was used to identify the potential software development companies in Sri Lanka. Companies listed in Table 3.2 were selected to share the online survey questionnaire. Out of those foreign companies only Sri Lankan operations considered in the survey and analysis. The combination covers organizations ranging from small, medium, to large scale. Those companies have at least one project based on ASD process.

Table 3.2: List of companies used for the study.

#	Company Name	#	Company Name
1	99x Technology Ltd	**14**	Informatics Holdings Ltd
2	Aeturnum Lanka (Pvt) Ltd	**15**	Infosoft Lanka (Pvt) Ltd
3	Aepona Int Lanka (Pvt) Ltd	**16**	Innovative-e Pvt Ltd
4	Millenium InformationTechnologies (Pvt) Ltd	**17**	John Keells Computer Services (Pvt) Ltd
5	Content Management and Solutions (Pvt) Ltd	**18**	Leapset (Pvt) Ltd
6	Dialog Business Services (Pvt) Ltd	**19**	Cambio healthcare system (Pvt) Ltd
7	Mubasher (Pvt) Ltd (DirecFN)	**20**	Netstarter (Pvt) Ltd
8	eBuilder Technology Centre (Pvt) Ltd	**21**	Pearson Lanka (Pvt) Ltd
9	Embla Solutions (Pvt) Ltd	**22**	Ridgecrest Asia (Pvt) Ltd
10	Exilesoft (Pvt) Ltd	**23**	ShipXpress (Pvt) Ltd
11	Hemnette Web Solution (Pvt) Ltd	**24**	Sim Centric Technologies (Pvt) Ltd
12	HSenid Business Solutions (Pvt) Ltd	**25**	Virtusa (Pvt) Ltd
13	IFS R and D International (Pvt) Ltd	**26**	Zone 24x7 (Pvt) Ltd

3.3.2 Process of Data Collection

Data collection process is described in this section. That was conducted in three separate stages as presented in Figure 3-1, such as stage 1, stage 2 and stage 3.

Software Metric Selection for the Online Survey Questionnaire

Agile metric selection for the online survey questionnaire was done very carefully. Sixteen metrics were collected only from the interviews carried out. Out of that, eight metrics (*Cost of quality*, *Customer satisfaction survey*, *Defect density*, *Defect removal efficiency*, *Defect severity index*, *Defect slippage rate*, *Requirement clarity index*, and *Sprint-level effort burndown*) were added to the online survey questionnaire. Those eight metrics used in more than one company. Manila (2013)

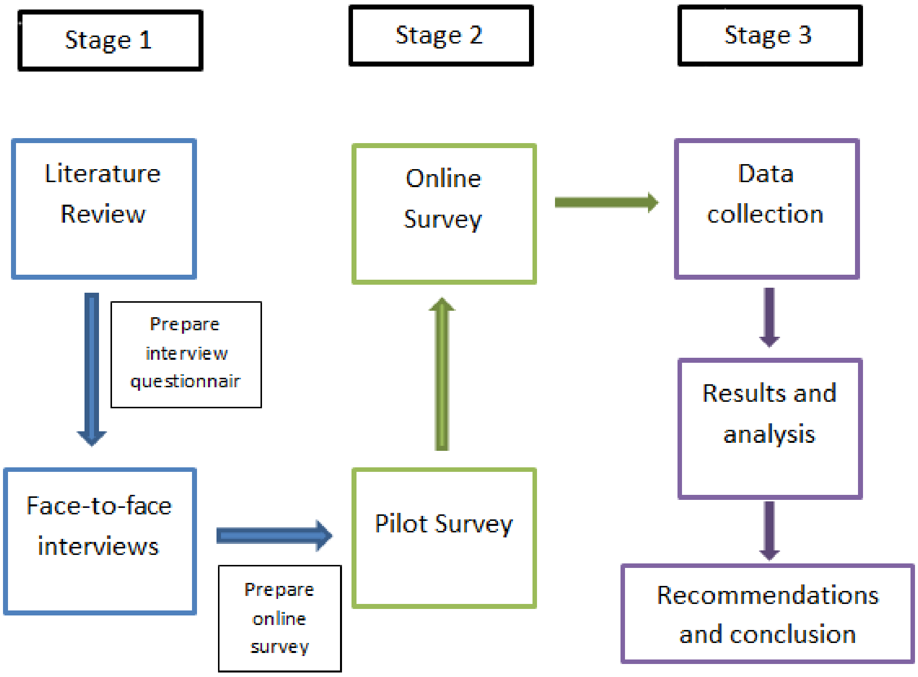


Figure 3-1: Process of data collection.

discussed about seven metrics. Only five metrics (*Delivery on time, Fault correction time to "closed" state, Open defect severity index, Technical dept, and Unit test coverage for the developed code*) were identified during the face-to-face interviews. Therefore, those five metrics also counted for the online survey questionnaire. Eight metrics out of the ten metrics mentioned in Downey and Sutherland (2013) had also taken into consideration, because those were identified as useful metrics to learn from Scott (2013). It was also mentioned that those eight metrics (*Accuracy of estimation, Accuracy of forecast, Focus factor, Percentage of adopted work, Percentage of found work, Targeted value increase* (*TVI+*), *Velocity, Work capacity*) used for hyper-productive Agile teams. *Net promoter score* metric was taken from Agile metrics (2013).

Stage 01

As illustrated in the Figure 3-1 face-to-face interview questionnaire was developed from the knowledge gathered from the literature and given in the Appendix A. Face-to-face interviews were conducted with four professionals from four different, well-known IT companies. There were one ASD and TSD process company. Rest are ASD process only companies. Only one company is a service delivery company and other three of them were product-base companies. Face-to-face interview questionnaire was shared beforehand with the interviewee to get prepared and to minimize the time taken to interview.

Stage 02

Online survey questionnaire was prepared with the knowledge gathered from the face-to-face interview questionnaire and literatures (Figure 3-1). The questions added to the online survey questionnaire were intended to understand the metrics used in ASD process and their usage. Separate set of questions were added to verify whether they are using ASD process techniques. Online survey questionnaire is given in Appendix C. Pilot survey conducted by sharing the online survey questionnaire among three QA leads, who had more than two years of experience. Those professionals were from three different companies, who have projects based only on ASD process. After incorporating the feedback given in the Appendix B, online survey questionnaire was shared among professionals working in the IT companies listed in the Table 3.2. The company identified for the research survey, according to the snowball sampling concept and was able to share the online survey questionnaire within 26 companies, who had projects based on ASD process.

Stage 03

The online survey was available for responses for about one and half months time from 14th October to 29th November, 2014. Survey participants were invited via e-mail. Follow-up interviews were conducted at the completion of the data analysis. The Follow-up interview questionnaire document was prepared and presented with the research findings, to the selected industry experts. The purpose was to compare and contrast the findings with expert judgements as well as, to get further feedback. Follow-up interview questionnaire is given in Appendix D and metric description is given in Appendix E.

4. DATA ANALYSIS

The chapter provides a detailed qualitative and quantitative analysis of the results. Section 4.1 provides an overview about the data gathering and data preparation for further analysis. Section 4.2 describes descriptive data analysis for the software metric usage within the projects based on the Agile Software Development (ASD) process. Analysis of demographic data is presented in Section 4.3. Scrum adherence of the projects is discussed in Section 4.4. Section 4.5 describes about the use of Agile tools. In Section 4.6 describes the summary of the findings.

4.1 Data Preparation for Analysis

The survey is conducted via an online questionnaire. Data collection was completed within a period of one and half months. Figure 4-1 illustrates the number of responses over time.

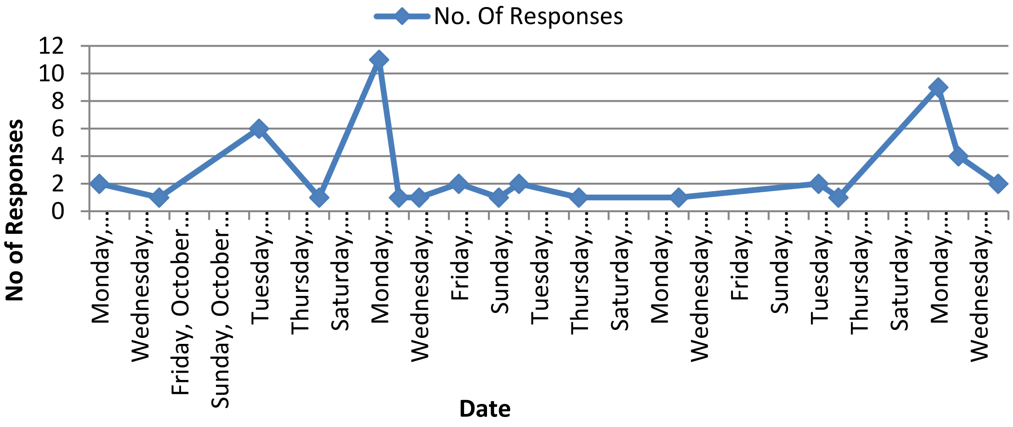


Figure 4-1: Number of responses per day from 14th Oct. to 28th Nov. 2013.

While 51 responses were received, three of them were not answered properly. Hence, in the following discussion we only considered remaining 48 acceptable responses. The weights were assigned to the responses based on the Likert scale (Boone and Boone, 2012). The research data sample was selected from the companies those having at least one project based on the ASD process. The selected companies are small, medium or large in scale and some companies were into service delivery while others were into product-based software development.

The online survey (see Section A in Appendix C) evaluated the use of 22 software metrics. The usage was evaluated in the form of a Likert-type scale, where the possible answers were *Used in Every project*, *Used in Some projects*, *Hardly used in projects*, *Never used in projects* and *Not Applicable*. The purpose of having the *Not Applicable* option was to identify whether the participants were able to recognize a particular metric as not relevant/applicable to the project context. However, while analysing the data, it was observed that the respondents seem to have used *Never used in project* option with the intention of *Not Applicable*. Therefore, when assigning weights, it was decided to combine the responses for both *Never used in projects* and *Not Applicable* together as represented in Table 4.1.

Table 4.1: Weight table for Likert-like scale.

Selection	Weight
Used in Every project	4
Used in Some project	3
Hardly used in projects	2
Never used in projects/ Not Applicable	1

4.2 Descriptive Statistics for Metric Usage

Responses were then weighted as per the Table 4.1. Resulting values are plotted in Figure 4-2. As illustrated in the graph, it can be seen that, *Delivery on time* metric is the most used metric and *Targeted Value Increase* (*TVI+*) metric is the least used metric. This behaviour can be further observed in Figure 4-3, which plots the median and mode of use of software metrics. In the Section 5.1, the reasons for this behaviour is discussed in detail with the expert judgement from subject matter experts.

As listed in Table 4.2, metrics selected for the survey were classified based on the SDLC process depending on where they are mostly used. Figure 4-4 shows the metric distribution based on which SDLC process they are classified. As illustrated in the Figure 4-4, when considering the metric usage as a percentage, 10 metrics were used more than 75% of the time. Among those 10 metrics, four metrics were identified as mostly used in the TSD process. Those metrics are *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, and *Open defect severity index*. However, according to the research findings, those four metrics are frequently used in ASD process as well. Therefore, these four metrics can be used in both the ASD and TSD processes. Remaining six metrics are *Work capacity*, *Percentage of adopted work*, *Sprint-level effort burndown*, *Velocity*, *Percentage of found work*, and *Focus factor*. These metrics were identified as Agile-specific metrics during the literature review and face-to-face interviews.

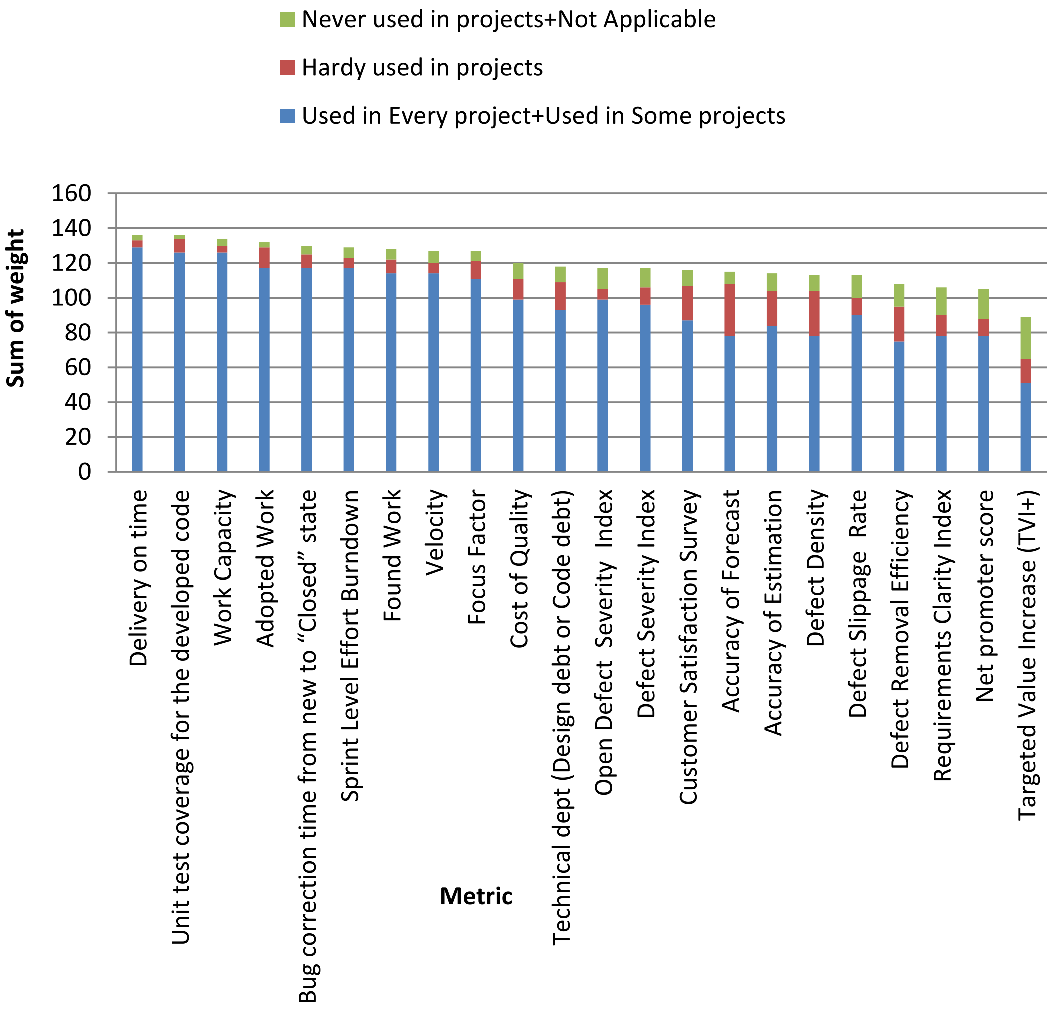


Figure 4-2: Metric usage against sum of weight.

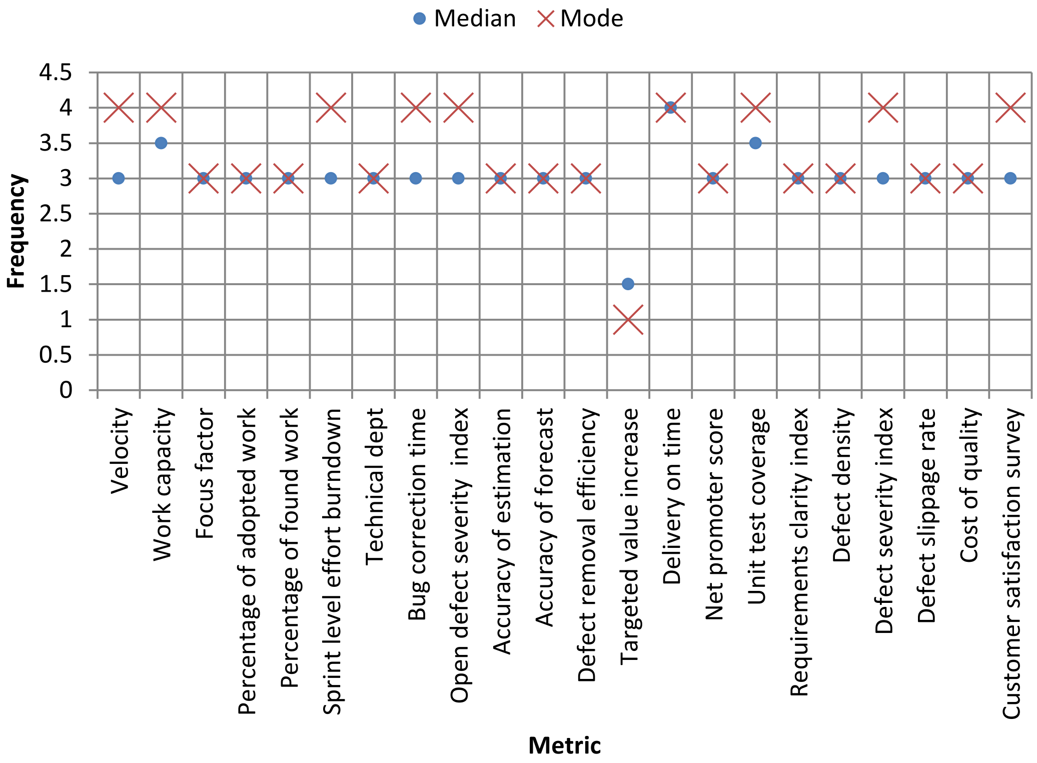


Figure 4-3: Median and mode of the metric usage.

There are eleven metrics with 50% to 75% of metric usage (see Table 4.3). Four of those metrics are identified as mostly used in the ASD process only companies. Those metrics are *Technical dept* (*Design debt or Code debt*), *Accuracy of estimation*, *Accuracy of forecast, and Net promoter score* metrics. Those are agile-specific metrics. Remaining seven metrics are identified as most used in TSD process. Those include *Cost of quality, Defect severity index*, *Defect slippage rate*, *Customer satisfaction survey*, *Requirement clarity index*, *Defect density*, *Defect removal efficiency*, *Accuracy of estimation*, *Accuracy of forecast*, and *Net promoter score*. Those metrics are most used in TSD process. *Requirement clarity index*, *Defect density*, and *Defect removal efficiency* metrics had 54%, 54% and 52%metric usage, respectively. These metrics are identified as most used in the TSD process. However, according to the research findings, those metrics are not frequently used in the ASD process. Among the 22 metrics, there is only one Agile-specific metric (i.e., *Targeted value increase* (*TVI+*)) with the metric usage of less than 50%.

Table 4.2: Classification of metric into relevant SDLC processes.

#	Metric Name	SDLC Process (Used mostly in)
1	Accuracy of Estimation	ASD process
2	Accuracy of Forecast	ASD process
3	Cost Of Quality	TSD process
4	Customer satisfaction survey	TSD process
5	Defect Density	TSD process
6	Defect Removal Efficiency	TSD process
7	Defect Severity Index	TSD process
8	Defect slippage rate	TSD process
9	Delivery on time	TSD process
10	Fault correction time to “Closed” state	TSD process
11	Focus Factor	ASD process
12	Net Promoter Score	ASD process
13	Open Defect Severity Index	TSD process
14	Percentage of Adopted Work	ASD process
15	Percentage of Found Work	ASD process
16	Requirements Clarity Index	TSD process
17	Sprint-level effort burndown	ASD process
18	Targeted Value Increase (TVI+)	ASD process
19	Technical debt	ASD process
20	Unit test coverage for the developed code	TSD process
21	Velocity	ASD process
22	Work Capacity	ASD process

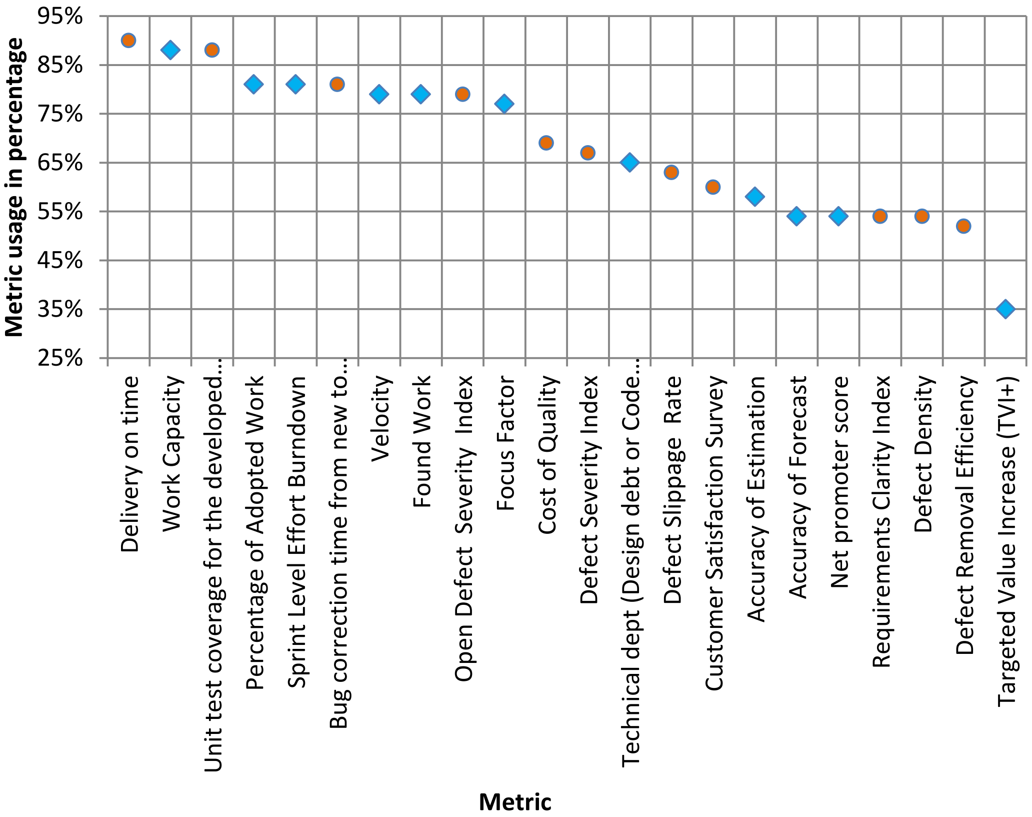


Figure 4-4: Metric distribution based on the SDLC process. Circles indicate the metrics mostly used in the TSD process and rest indicates the metrics most used in the ASD process.

Defect density metric was not used frequently by the Agile process only companies (see Figure 4-5). However, TSD and ASD process companies used it frequently. The percentage of the metric usage was 23% for the ASD process only companies and 71% for ASD and TSD process companies. During the face-to-face interview sessions, it was identified that two different definitions were used for the D*efect density* metric by ASD process only companies and ASD and TSD process companies. Metric definition in TSD process is “number of defects per unit of product size”. In the ASD process, it is defined as the “number of defects per story point of the story”.

Table 4.3: Percentage of metric usage.

#	Metric	% Usage	Identified as Used In
1	Delivery on time	90%	TSD process
2	Work capacity	88%	ASD process
3	Unit test coverage for the developed code	88%	TSD process
4	Percentage of adopted work	81%	ASD process
5	Sprint-level effort burndown	81%	ASD process
6	Bug correction time from “new” to “closed” state	81%	TSD process
7	Velocity	79%	ASD process
8	Percentage of found work	79%	ASD process
9	Open defect severity index	79%	TSD process
10	Focus factor	77%	ASD process
11	Cost of quality	69%	TSD process
12	Defect severity index	67%	TSD process
13	Technical dept (Design debt or Code debt)	65%	ASD process
14	Defect slippage rate	63%	TSD process
15	Customer satisfaction survey	60%	TSD process
16	Accuracy of estimation	58%	ASD process
17	Accuracy of forecast	54%	ASD process
18	Net promoter score	54%	ASD process
19	Requirements clarity index	54%	TSD process
20	Defect density	54%	TSD process
21	Defect removal efficiency	52%	TSD process
22	Targeted value increase (TVI+)	35%	ASD process

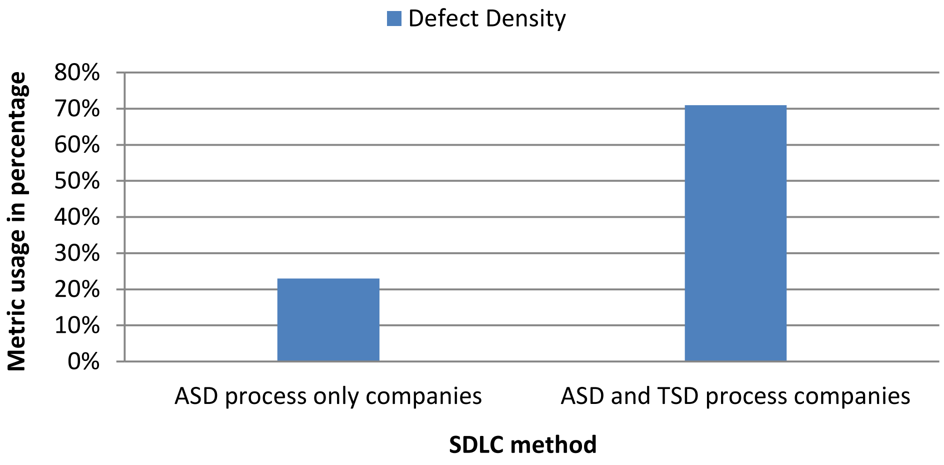


Figure 4-5: *Defect density* metric usage in ASD process only companies and ASD and TSD process companies.

After identifying that the usage of *Defect density* metric differ from ASD process only companies and ASD and TSD process companies, two separate graphs are drawn to find out whether such differences are observed for other metrics as well. Metric usage in ASD process only companies are illustrates in Figure 4-6 and metric usage in ASD and TSD process companies are illustrated at the Figure 4-7. As in Figure 4-6, both *Sprint-level effort burndown* metric and *Unit test coverage for the developed code* metric are 100% used by the participants from ASD process only companies. However, Figure 4-7 shows that the *Delivery on time* metric is the most used metric and *Targeted Value Increase* (*TVI+*) metric is the least used metric.

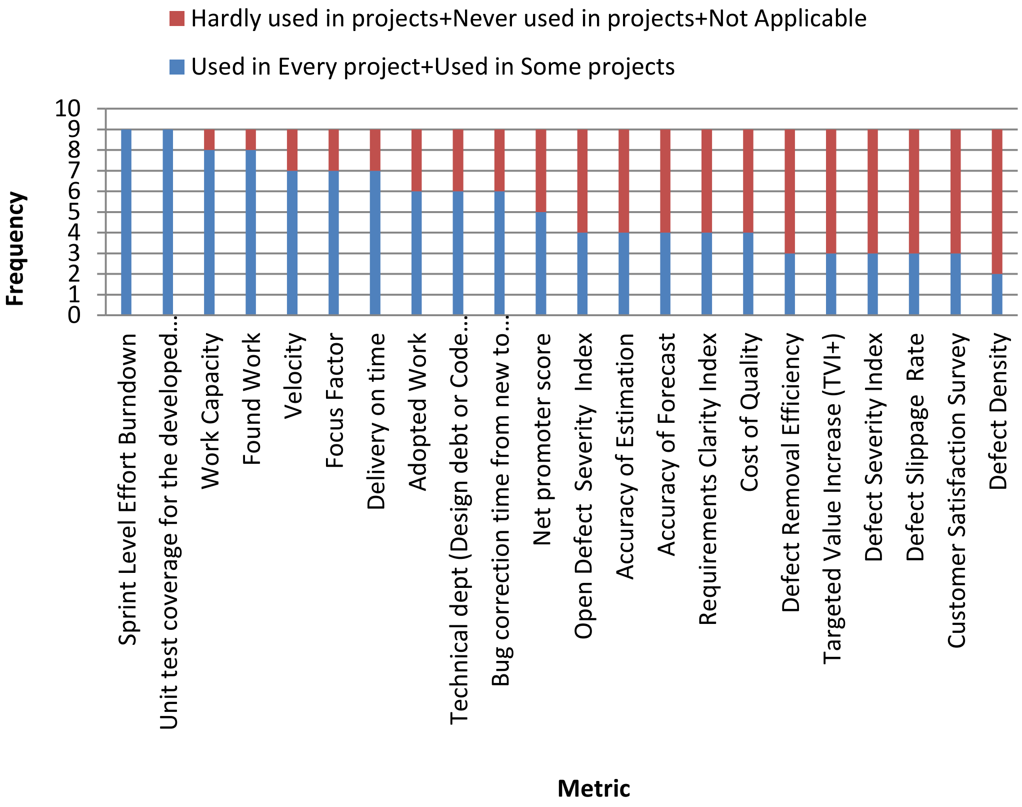


Figure 4-6: Use of metrics in ASD process only companies.

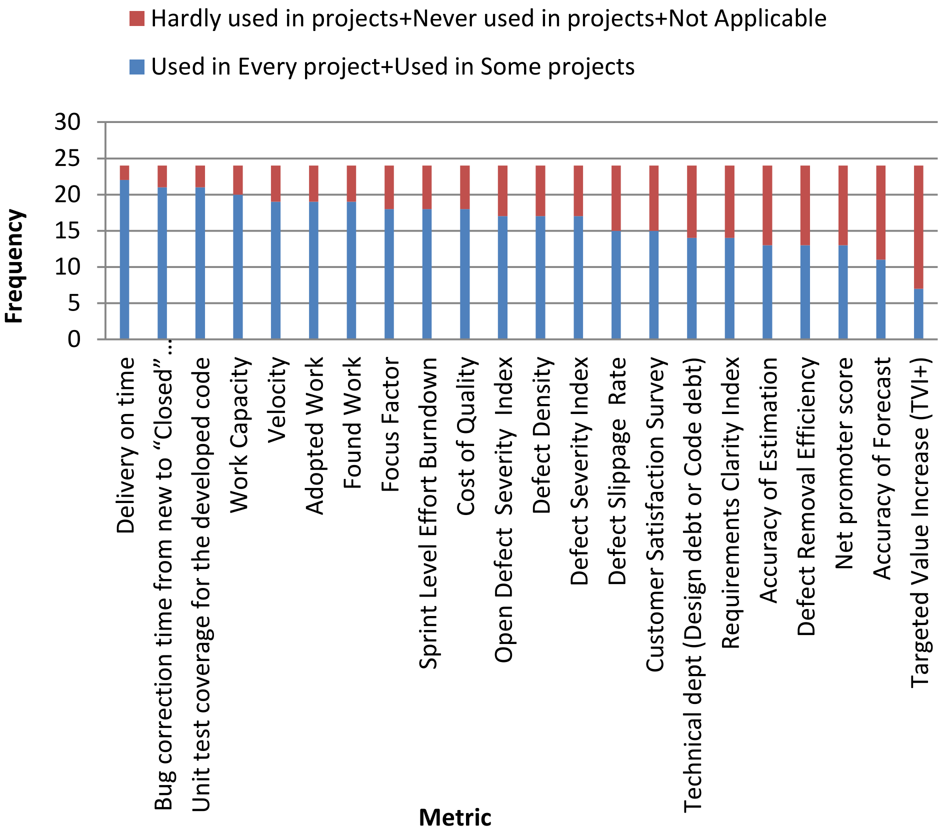


Figure 4-7: Use of metrics in ASD and TSD process companies.

4.2.1 Number of Agile Projects in a Company

Number of Agile projects in a company differs from company to company. Therefore, Figure 4-8 and Figure 4-9 are drawn to find out any significant different exists in the metric usage, depending on the number of Agile-base projects in a company.

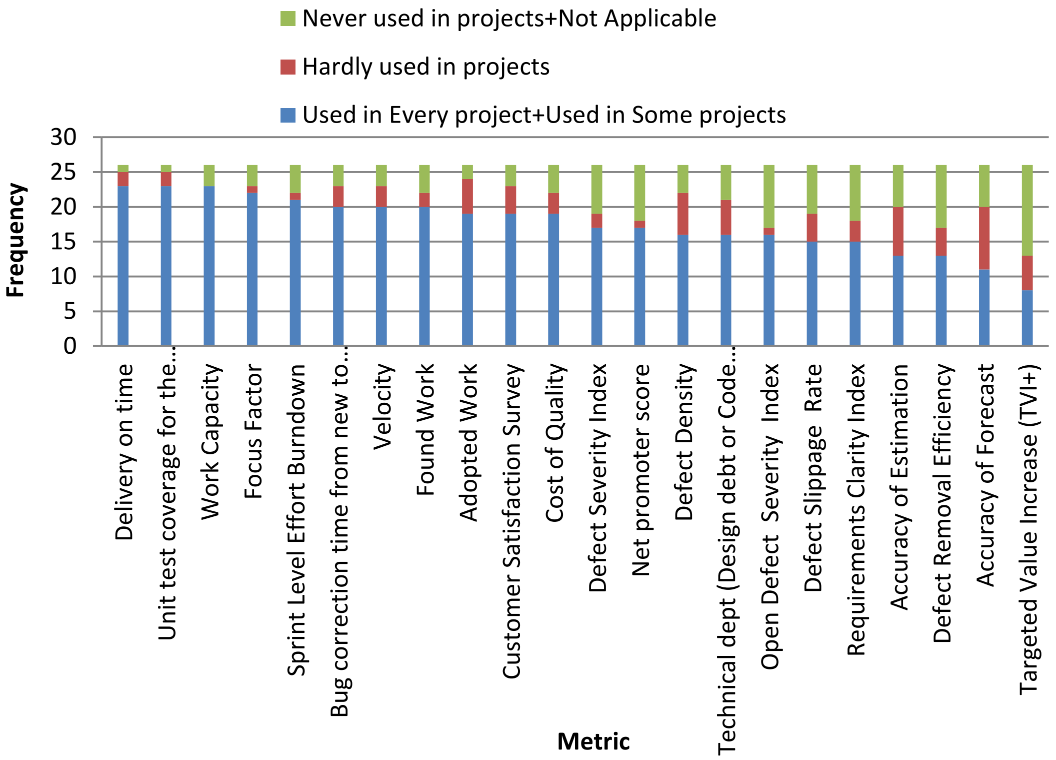


Figure 4-8: Companies having 10 or less than 10 Agile projects against the metric usage.

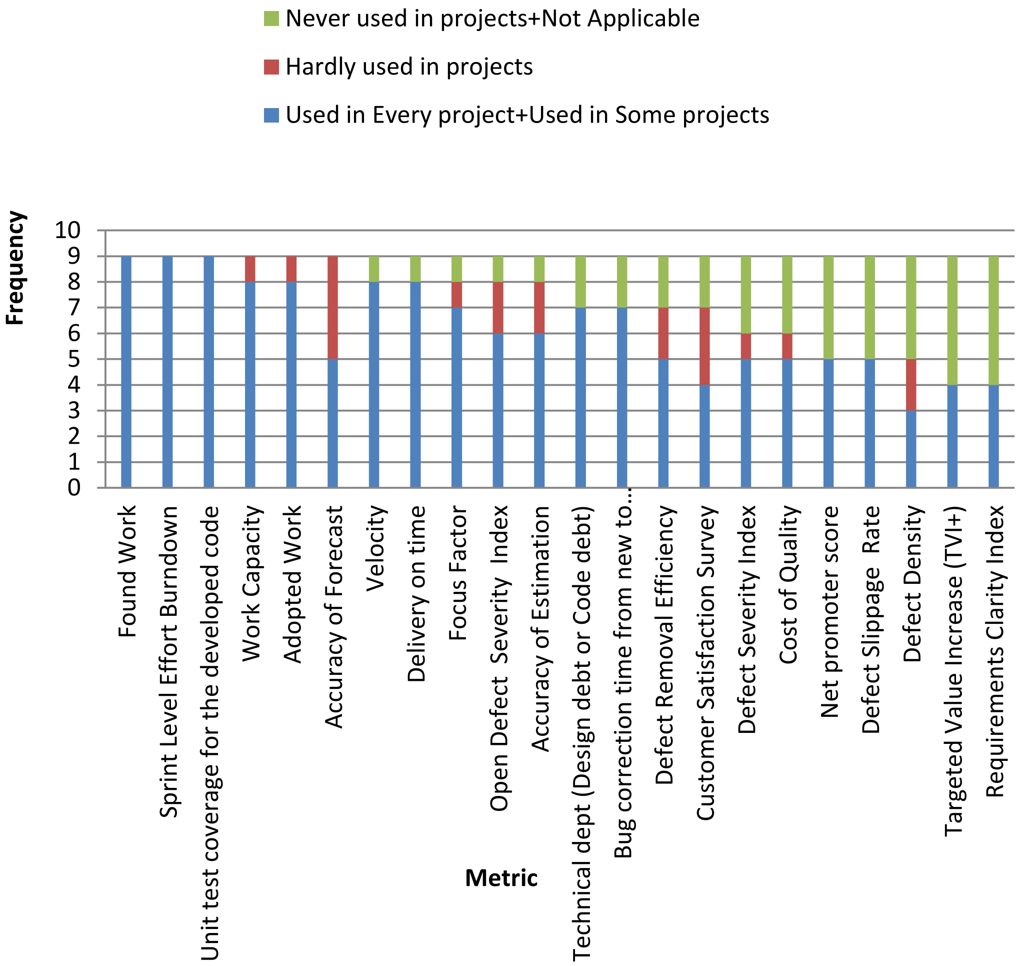


Figure4-9: Companies with more than 10 Agile projects against metric usage.

As illustrated in Figure 4-8, *Delivery on time* metric has the highest value for percentage of usage and *Targeted value increase* (*TVI+*) metric has the least value. Figure4-9 shows that *Percentage of found work*, *Sprint-level effort burndown*, and *Unit test coverage for the developed code* metrics are used in every project. *Requirement clarity index* metric, *Defect density* and *Targeted value increase* (*TVI+*) metrics are least used in those projects as well.

4.3 Descriptive Statistics for Demographic Data

Purpose of analysing the demographic data is to identify to what extent the demographics have an impact on metric usage. For the analysis of demographic data, following criteria were considered. Those are, the respondents role in the scrum team, respondents years of experience in the ASD process, respondents knowledge level about the ASD process and metric usage in the ASD process, and completion of the ASD process specific trainings or workshops.

4.3.1 Role Played in the ASD Process

As shown in Figure 4-10, 39% (18) of the participants are Quality Assurance (QA) team members and 33% (15) of them are developers. Only 26% (12) of respondents are scrum masters and only one product owner. When considering the percentage most of the participants were team members, who were into QA and Development.

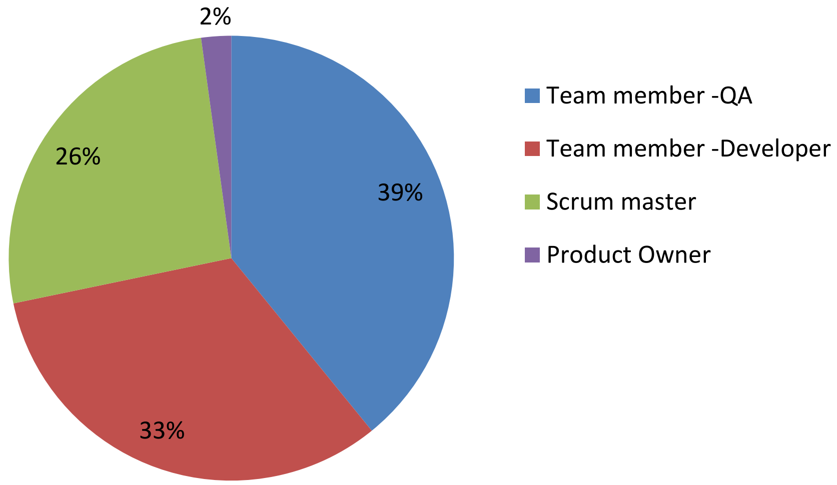


Figure 4-10: Role distribution of respondents in research survey.

4.3.2 Experience in the ASD Process

When considering the participants experience level in ASD process, 31% (15) of them had 1-2 years of experience and only 4% (2) of participants with more than 5 years of experience as illustrated in Figure 4-11. Therefore, majority of the participants have 1-2 years and 3-4 years of experience.

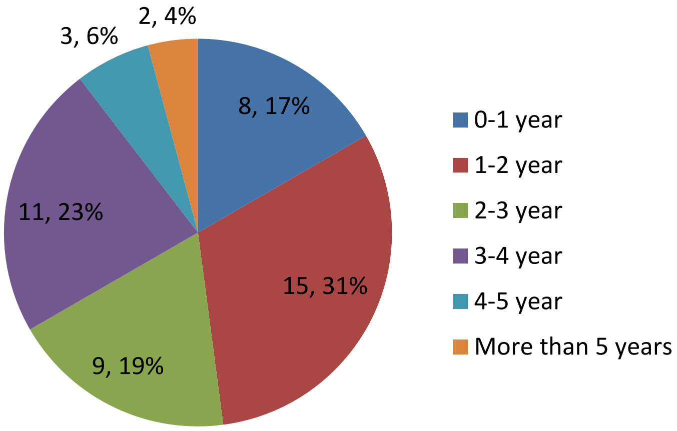


Figure 4-11: Years of Agile experience of participants.

4.3.3 Knowledge Level of ASD Process and Metric Usage in the ASD Process

To conclude it is important to get an understanding about the participant's knowledge level about the ASD process and metric usage in ASD process. Therefore, when considering the responses there are 63% (30) participants, who have average knowledge level about the ASD process (see Figure 4-12). Almost the same numbers of participants (60%, 29) have average knowledge level about the metric usage in the ASD process. Therefore, research survey participants have average knowledge level about ASD process and metrics usage in ASD process. Though there are 23% (11) of participants with above average knowledge level of ASD process, only 15% (7) of participants have above average knowledge level of metric usage in ASD process.

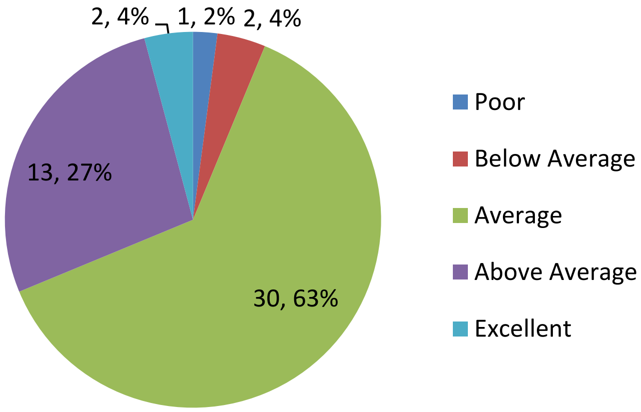


Figure 4-12: Knowledge level of participants about the ASD process.

Figure 4-13 illustrates the participants' knowledge level about the metric usage in the ASD process. In the below average and poor category, there are 21% (10) of participants. Therefore, majority of the participants have average knowledge level about metric usage in the ASD process.

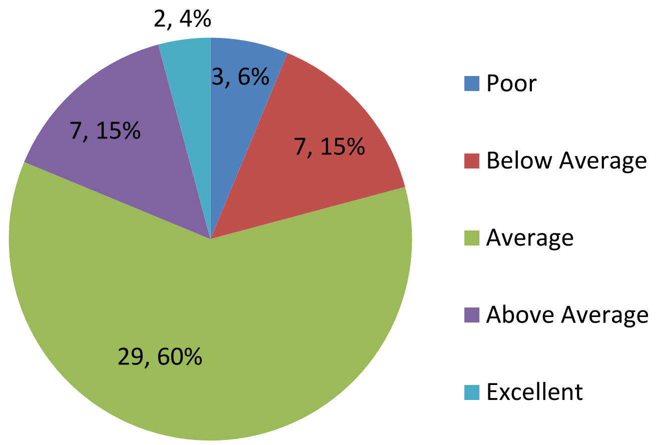


Figure 4-13: Knowledge level of participants about the metric usage in the ASD process.

Figure 4-14 illustrates the graph of metric usage against above average and excellent knowledge level about the ASD process. All the participants are using *Velocity*, *Work capacity*, *Sprint-level effort burndown*, and *Unit test coverage for the developed code* metrics. In that, other than the *Unit test coverage for the developed code* metric, rest of the metrics are Agile-specific metrics.

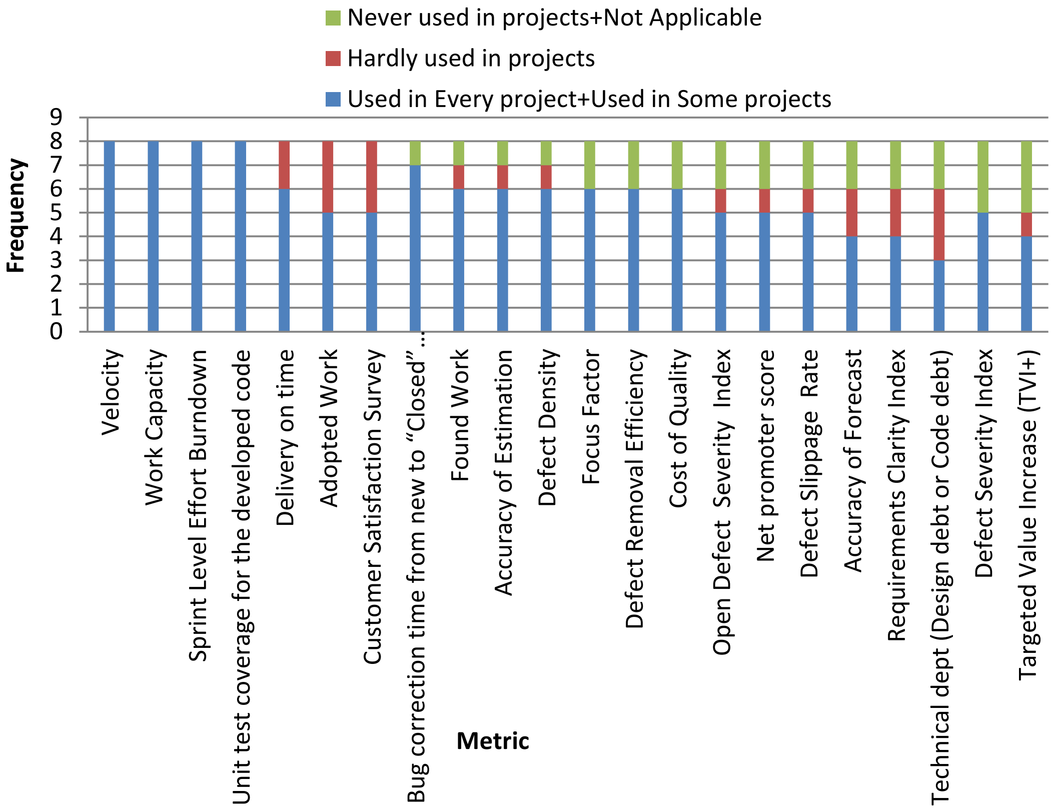


Figure 4-14: Metric usage against above average and excellent knowledge level about ASD process.

4.3.4 Knowledge Level Against Level of Experience

Figure 4-15 illustrates the distribution of experience against respondents with average knowledge level about ASD process. Least number of participants has more than 5 years of experience. Majority of the participants with average knowledge level about the ASD process are having 1-2 years or 3-4 years of experience.

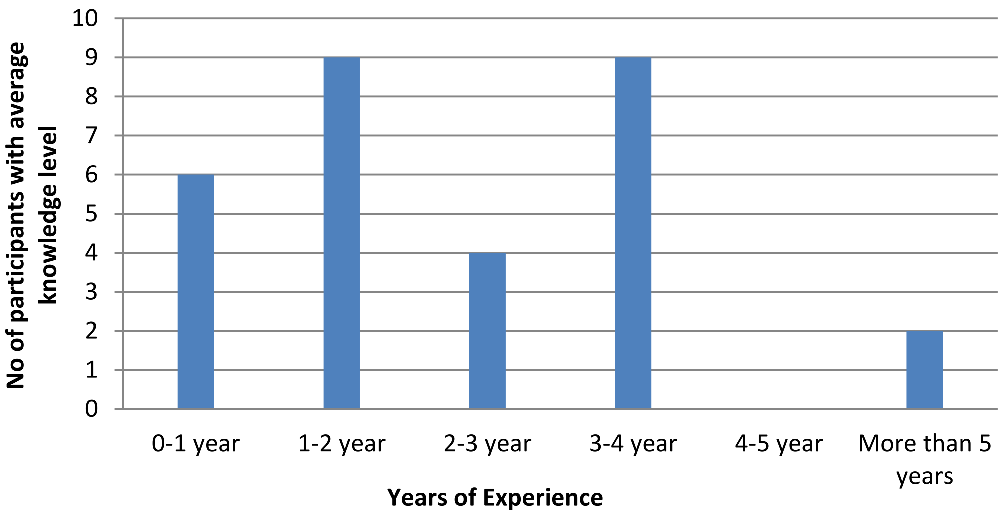


Figure 4-15: Average knowledge level about the ASD process against participants' experience.

Figure 4-16 represents the experience level of the participants who have an average knowledge level about metric usage in ASD process. Number of participants from 1-2 years or 3-4 years had reduced when compare with the Figure 4-15. The participants with more than 5 years of experience have average knowledge level in both the situations.

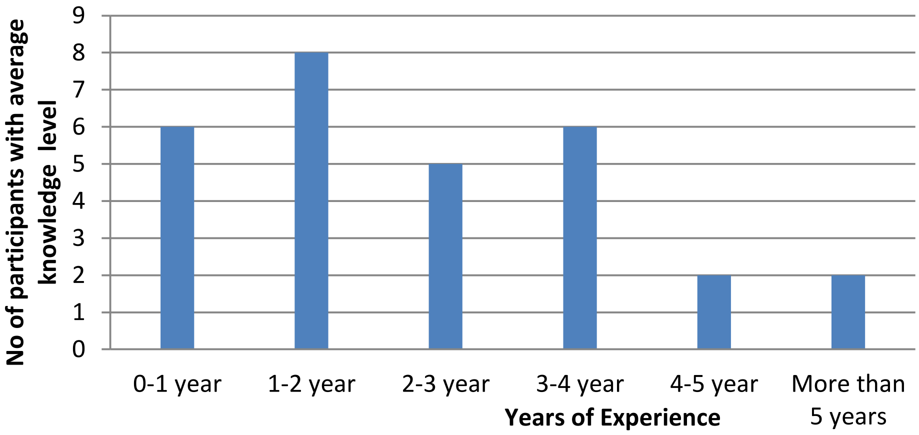


Figure 4-16: Average knowledge level about metric usage in the ASD process against the participant's experience.

Figure 4-17 illustrates the below average and poor knowledge level of metric usage in ASD process against the years of experience. In that, participants who have more than 2 years of experience have poor or below average knowledge level about metric usage in an ASD process, which was 13% (6).

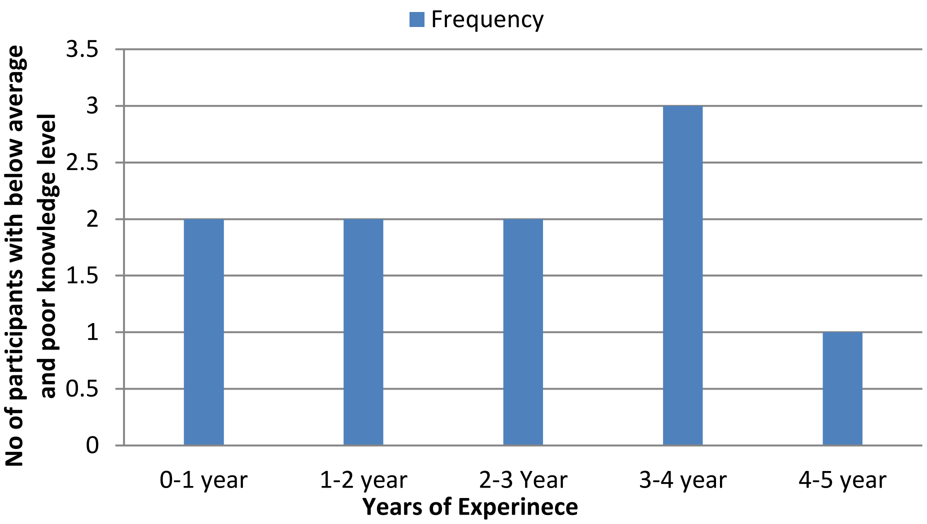


Figure 4-17: Knowledge level of poor and below average of metric usage in ASD process against years of experience.

4.3.5 Completion of ASD Processes Specific Training or Workshops

As illustrates in the Figure 4-18, 48% (23) of participants had followed trainings related to ASD process. Majority of the participants had followed Agile trainings.

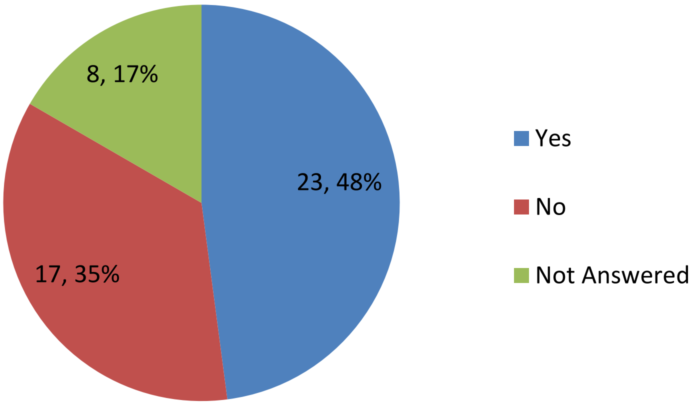


Figure 4-18: Followed ASD process training/workshops.

Participants had received the Scrum training in three ways, such as, through Scrum training workshops, by participating for Agile conferences and during their personal studies. As illustrated in Figure 4-19 participating for the Scrum training workshops was the most used training method.

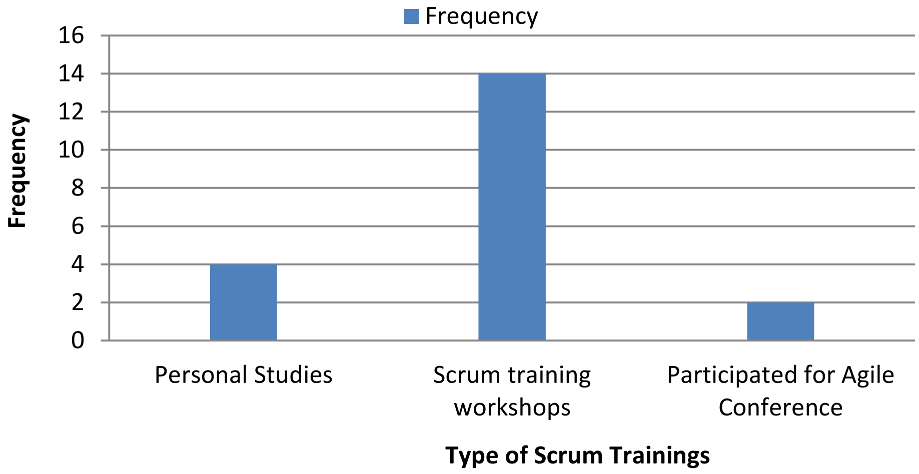


Figure 4-19: Type of Scrum trainings followed by the participants.

4.4 Analysis of Scrum Adherence

The Scrum Development Process was the most used ASD methodology practiced by the participants. There are 63% (43) of participants for Scrum development process. Several companies were practicing in more than one ASD methodology, as illustrated in Figure 4-20.

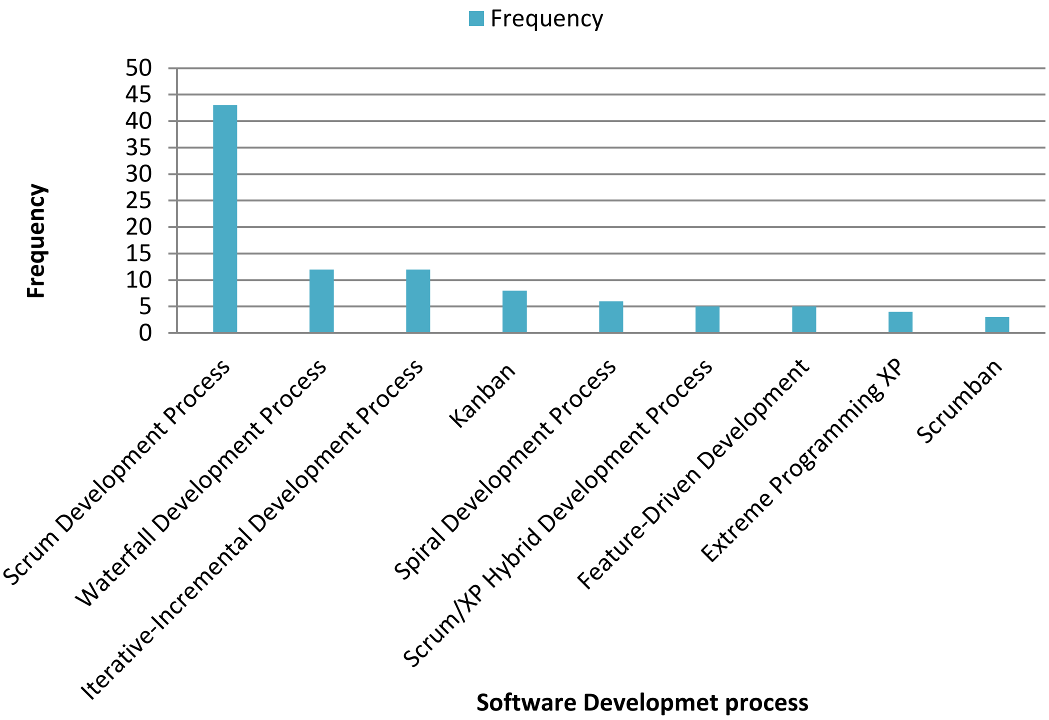


Figure 4-20: ASD methodology against participant's usage.

More than half of the participants were practicing in the Scrum Development process. 5% (3) and 7% (5) participants are practicing Scrumban and Scrum/XP Hybrid development Process respectively. By combining all the Scrum variations, there is 75% of Scrum and customized scrum development processes in use within survey participants, which illustrated in Figure 4-21. Therefore, it is better to verify that the participants are practicing in Scrum Development process rather than verifying it for all the ASD methodologies.

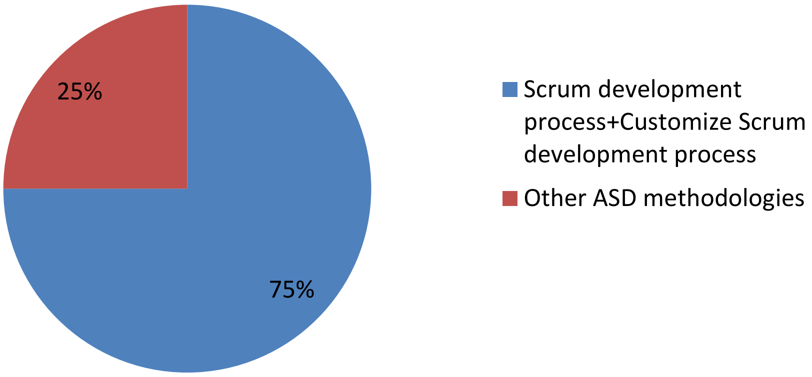


Figure 4-21: Respondents usage of Scrum and customize scrum development process against other ASD methodologies.

4.4.1 Common Agile Practices Followed in the Scrum Development Process

When considering the common Agile practices followed in the Scrum Development process, most of the participants are conducting sprint planning meetings, which is 90% (43), daily stand-up (85%, 41) and release planning meetings (75%, 36). Few participants are using the Analog or Digital task board and Working product whereas, 33% (14) and 40% (19) respectively in the frequency of the response. As illustrated in the Figure 4-22, other than the Working product and Analog or Digital task board practices remaining five practices are used by more than 50% participants.

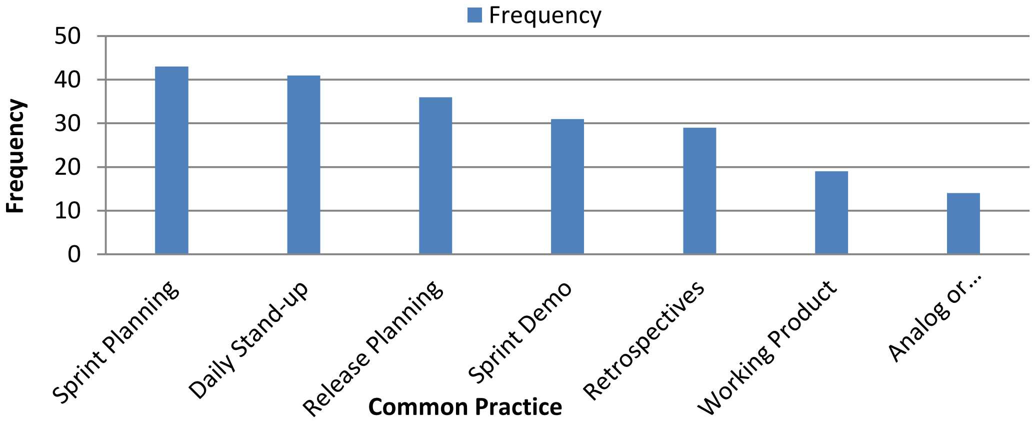


Figure 4-22: Common Agile practices followed in the Scrum Development process.

4.4.2 Number of Team Members in Scrum Development Process

Typically, scrum team members are calculated as 7+/7- ~ 5 to 9 team members in a project (Schwaber & Sutherland, 2013). In accordance with the responses represented in Figure 4-23, there are 57% (33) of response for the 6-12 members category.

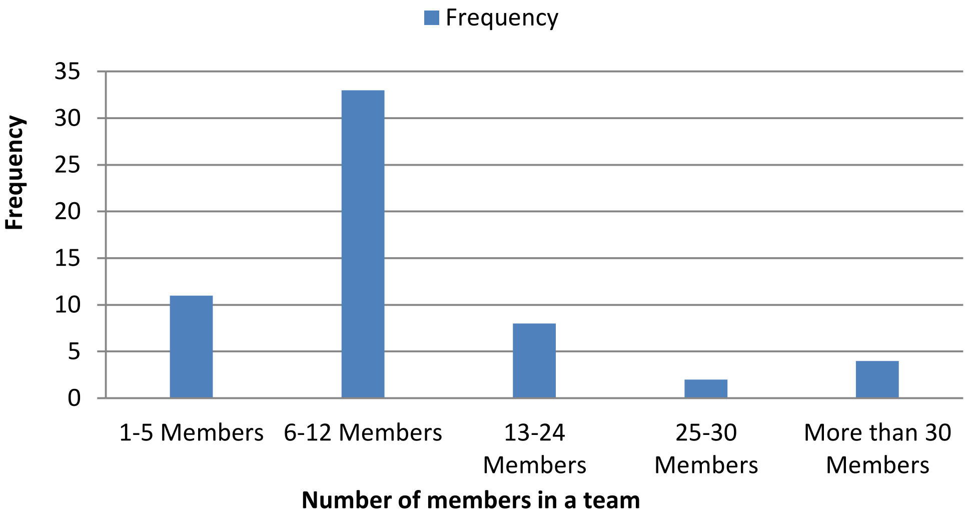


Figure 4-23: Number of team members in a Scrum development process team.

4.4.3 Sprint Plan Duration in the Scrum Development Process

The typical duration of a sprint cycle time was 2-4 weeks (Schwaber & Sutherland, 2013). As shown in Figure 4-24 75% of the participants indicated that their sprints span for 2 weeks.

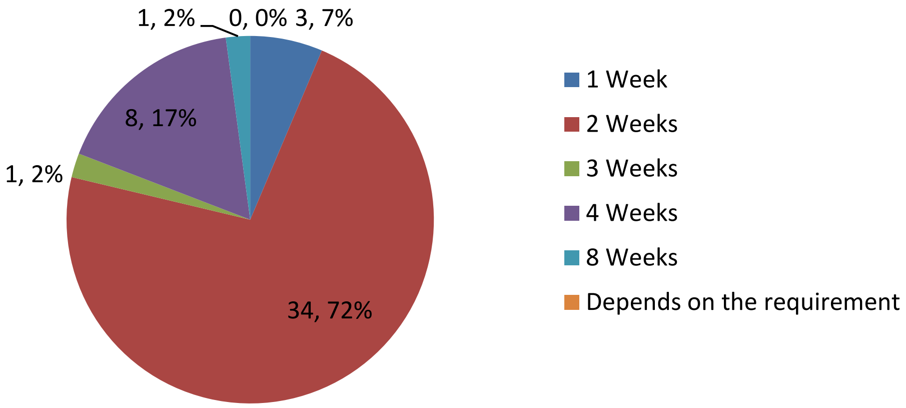


Figure 4-24: Sprint plan duration in the Scrum Development process.

4.4.4 Release Plan Duration in the Scrum Development Process

Most of the release plans were scheduled for 2 weeks time (46%, 22). Release plan was not strictly maintained in the ASD process. The duration will be different due to the project environment and client requirements (Figure 4-25).

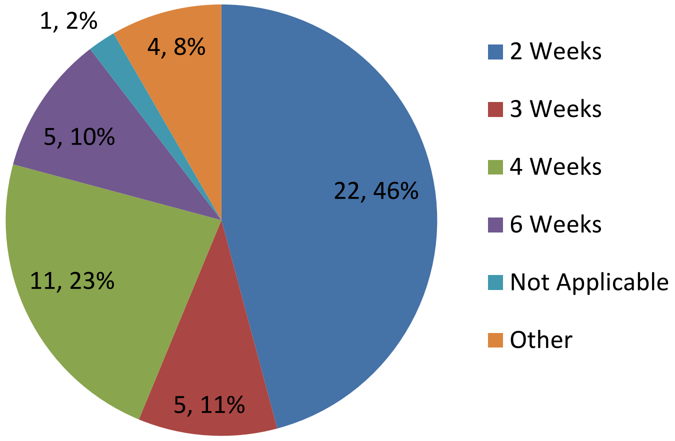


Figure 4-25: Release plan duration in the Scrum Development process.

4.4.5 Retrospective in Scrum Development Process

32 participants were conducting retrospectives, which was 67% out of accepted responses. There was no relationship with the use of metric and conducting retrospective meetings. Most of the participants were conducting retrospectives at the end of the sprint or at the sprint demo. Some projects even conduct retrospectives with clients and internal teams separately. Some having at the end of release or every 6 months and some are conducting the meeting only at a critical situation.

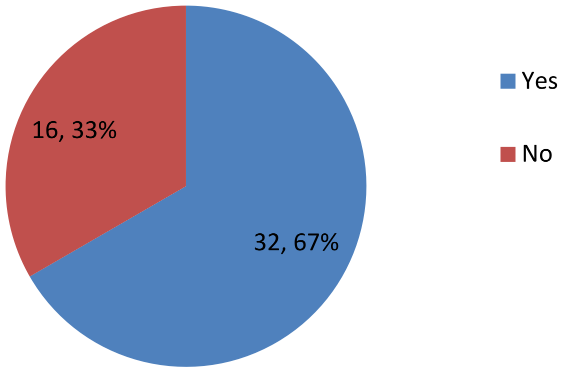


Figure 4-26: Conduct retrospective within the project.

4.4.6 Benefits of Metric Usage in the ASD Process

Question was added to get an understanding about the perception of software metrics usage in Agile process among software professionals. As per the responses, each participant agrees with the idea of metric usage in an Agile process. There were 4% (2) irrelevant responses. One responded had stated benefit of ASD process usage as "Agile process increases the verification (i.e., process of ensuring making the right product according to the client requirements) which is one of the most important aspects of a project or a product", who was a scrum master from a product development company. Three respondents out of 48, does not had an idea about metrics. Most of the participants agree that the metric usage is valuable to track the project progress and the project. In addition, participants had mentioned that, metric used to check the project health as well as to evaluate the current progress of the project. Some had mentioned that metrics helps to identify the areas where improvements required within the project. At the same time, participants had mentioned that they should be very careful when selecting metrics and most relevant metrics should be selected to use within the project. Otherwise, extra burden to the team would spoil the project progress. Comments from the participants' are presented in Table 4.4.

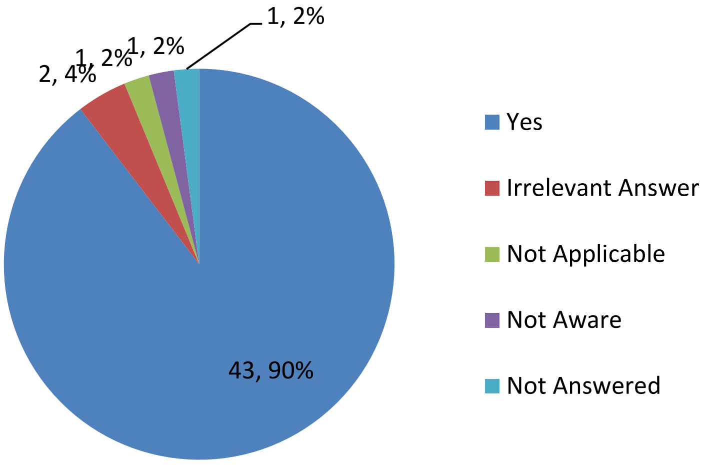


Figure 4-27: Benefits of metric usage in the ASD process.

One respondent had mentioned that, metric usage with ASD process would be beneficial if it is automated. This is discussed in detail in Section 4.5. A respondent had mentioned that selected metric for the project should be easy to understand and calculate. One respondent, who was a certified scrum master from a product development company, had stated that "Yes it is, but use it in a better way is not visible from most Sri Lankan companies."

Table 4.4: Benefits of metrics usage in the ASD process

#	Comments	Frequency
1	To track the project progress	14
2	To monitor the quality aspect of the project or product	8
3	Helps team to forecast and manage the project better	6
4	Identify the areas needs to be improved	3
5	Helps to improve the estimation	1
6	To check project health	1
7	It helps to take maximum out of the resources	1
8	Helps to improve the development process	1
9	Brings alignment to across cross-functional teams working in the same project.	1

4.5 Use of Agile Tools

As illustrated in Figure 4-28 JIRA/Greenhoper was the most used tool out of all the accepted responses. However, when considering ASD and TSD process organizations, they heavily use Microsoft Excel as a tool. This was marked as the second popular tool. ASD process only companies are using tools, which are specific for Agile environment, with the Microsoft excel as a support tool.

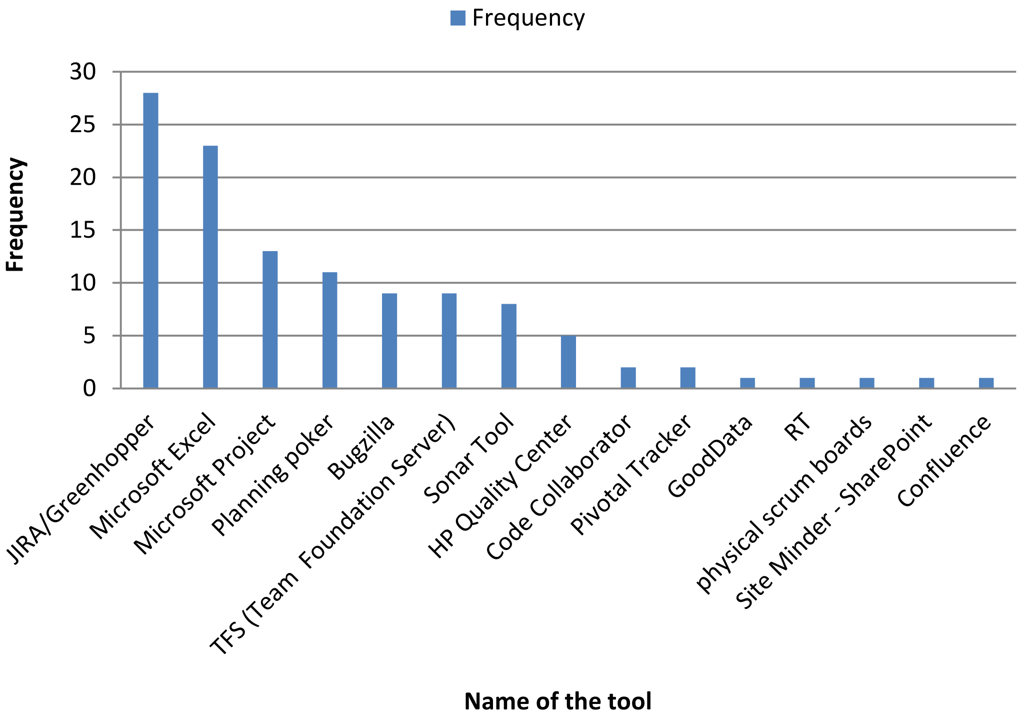


Figure 4-28: Frequency of Tool usage.

As illustrated in Figure 4-28, it depicts frequency of tools usage in ASD process only companies. Due to client requests, some companies are using other tools like Microsoft Excel, Hp Quality Center etc., while they use JIRA/Greenhopper tool. Below shows the identified combination of tools usage within projects, out of all the responses.

1. JIRA/Greenhopper
2. JIRA/Greenhopper, Bugzilla, Microsoft Excel, Code Collaborator
3. JIRA/Greenhopper, Bugzilla, Microsoft Excel, Microsoft Project, Sonar Tool
4. JIRA/Greenhopper, HP Quality Center

5. JIRA/Greenhopper, HP Quality Center, Code Collaborator, Planning poker, physical scrum boards
6. JIRA/Greenhopper, Microsoft Excel

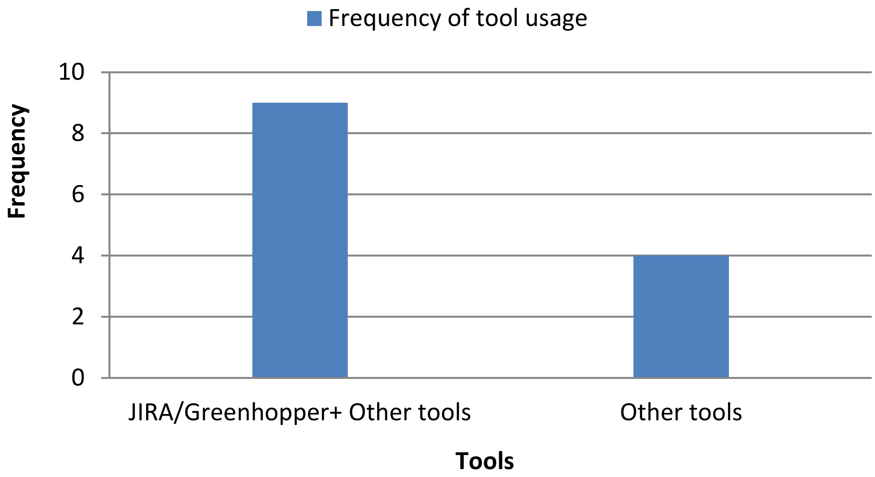


Figure 4-29: Tool usage in ASD process only companies.

There is a considerable difference in the tool preference, depending on the use of tools in the projects, based on the ASD and TSD process companies and the ASD process only companies. As shown in Figure 4-30, the Microsoft Excel with other support tools was the most used tool in ASD and TSD process companies. Most of the time companies try to use the same tools rather than going for a new, expensive tool. Other than that, software engineering professionals were already having hands on experience on Microsoft Excel, since they were using that tool for the projects based on TSD process. As described earlier, in ASD process only companies are using JIRA/Greenhopper with other support tools. That is because they find it more reliable to use Agile specific tool.

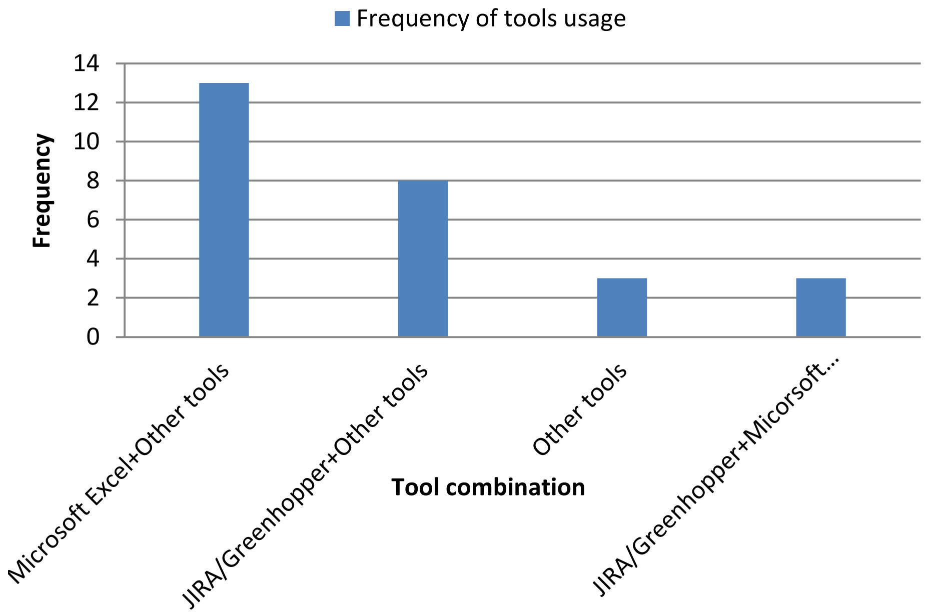


Figure 4-30: Tool usage in ASD and TSD process companies.

4.6 Summary

It is beneficial to adopt metrics in an Agile project, as it helps to track the project progress, monitor product quality, and enables better forecasting and project management. Ten metrics are suitable for the ASD process. Those include *Delivery on time*, *Work capacity*, *Unit test coverage for the developed code* , *Percentage of adopted work*, *Bug correction time from "new" to "closed" state*, *Sprint-level effort burndown*, *Velocity*, *Percentage of found work*, *Open defect severity index*, and *Focus factor*. It is also identified three new ASD-specific metrics namely; *Thumbs-up rule*, *Noncompliance index*, and *Top hill view*. *Work capacity*, *Percentage of adopted work*, *Sprint-level effort burndown*, *Velocity*, *Percentage of found work*, and *Focus factor* are the mostly used Agile-specific metrics. *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, and *Open defect severity index* metrics can be used in both TSD and ASD processes. *Delivery on time* metric has the highest value for percentage of usage and *Targeted value increase* (*TVI+*) metric has the least value, though it is a Agile-specific metric. *Defect density* metric does not use frequently in Agile process only companies. Companies fully into Agile practices mostly use specialized tool like

JIRA/Greenhopper to keep track of metrics while others relied heavily on Microsoft Excel.

When considering the percentage most of the participants were team members, who were into QA and Development. Majority of the participants have 1-2 years and 3-4 years of experience and they have average knowledge level about metric usage in the ASD process. Majority of the participants with average knowledge level about the ASD process are having 1-2 years or 3-4 years of experience. There are 23% (11) of participants with above average knowledge level of ASD process, only 15% (7) of participants have above average knowledge level of metric usage in ASD process. Participants had received the Scrum training in three ways, such as, through Scrum training workshops, by participating for Agile conferences and during their personal studies.

The Scrum Development Process was the most used ASD methodology practiced by the participants. Most of the participants are conducting sprint planning meetings, release planning meetings, and retrospectives. Few participants are using the Analog or Digital task board and Working product. Most of the teams are with 6-12 team members with 2 weeks sprint span. However, release plan duration is different due to the project environment and client requirement.

5. RECOMMENDATIONS AND CONCLUSION

The chapter discuss the recommendations based on the research findings. Section 5.1 explains about the research implications. Section 5.2 and 5.3 explain the recommendations and, future work respectively. Section 5.4 discusses about the overall conclusions of the research findings.

5.1 Research Implication

As the Agile software development process becomes mainstream, it is important to identify the most suitable set of software metrics within the ASD process. When analysing the responses about the benefits of use of metrics in the ASD process, listed in the Table 4.4, almost all the participants like to use metrics in the ASD process. This emphasizes the importance of finding an appropriate set of metrics to be used in the ASD process.

As illustrated in Figure 4-4, when considering the metric used in the ASD process, following results standout at each and every time:

- *Delivery on time* was the most used metric
- *Targeted Value Increase* (*TVI+*) was the least used metric

There are ten metrics, which was used by more than 75% of participants out of all the accepted responses (see Table 4.3). Six of those metrics are Agile-specific. Those are the *Work capacity*, *Adopted work*, *Sprint-level effort burndown*, *Velocity*, *Found Work*, and *Focus Factor*. Rest of four metrics were used in the TSD process as well. Those are the *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, and *Open defect severity index*.

Delivery on time metric got the highest value for the frequency of usage. While conducting follow-up interviews with subject matter experts, they agreed this metric was very important to measure apart from product base or service delivery organization. The metric gives a good indication of whether the scope being managed or understood. It also assists in tracking the project progress. Moreover, it can be used for prediction purposes.

According to the responses from the subject matter experts proves that, *Unit test coverage for the developed code* is important to measure in the ASD process. It contributes to the quality of the code. One interviewee from the ASD process only company had mentioned that they are using separate tools to automate and run unit tests. Therefore, the use of the metric is not a burden to the team.

Open defect severity index metric can be used to measure the quality of the product. The interviewee from an ASD process only company had mentioned that they measure it at the end of each sprint. They make sure to maintain zero open defects at the end of each sprint. If they had any, they bring it into the sprint retrospective and discuss to avoid those situations in future.

When it comes to the *Bug correction time from "new" to "closed" state* metric, there was a discrepancy between the interviewees ideas. Certified scrum master and a subject matter expert from a product development company, practicing ASD process in every project, had mentioned that, "they are not concerned about this kind of measures as long as the defects are closed". According to his point of view, this metric focused more on the individual level. Agile is more human-centric and encourages growing as a team. Therefore, this metric does not give any value to the Agile team. Further, he mentioned that, the metric could be practised within the team if they find a value of use of the metric. The interviewee from a company which have projects based on both ASD and TSD processes mentioned that, *Bug correction time from "new" to "closed" state* metric brings value to them and gives defect fixing efficiency of the team, which they used for future planning and estimation. They also use a separate tool to measure the metric, which reduces the burden of the team. Out of the online survey responses for *Bug correction time from "new" to "closed" state* metric, participants had selected *Hardly used in projects, Never used in projects* and *Not applicable* options who were from ASD process only companies. Rest of the participants had selected the *Used in Every project* and *Used in some projects* options.

In accordance with the responses as shown in Table 4.3, there were 11 metrics, which were used by less than 75% and more than 50% of responses out of all the accepted responses. There were only four metrics (*Technical dept* (*Design debt or Code debt*), *Accuracy of Estimation*, *Accuracy of Forecast*, and *Net promoter score*) which were identified as the Agile-specific metrics. The rest of the metrics were identified as most used in the TSD process. As shown in Table 4.3, there is only one metric (*Targeted value increase* (*TVI+*)) which were used by less than 50% of responses. That metric is an Agile-specific metric. As illustrated in the Figure 4-6, ASD process only companies are mostly using Agile-specific metrics. All of them used the *Sprint-level effort burndown* metric. *Defect Density* metric showed a significant difference of metric usage between ASD process only companies and ASD and TSD process companies. 23% participants in ASD process only companies and 71% participants from ASD and TSD process companies use it (see Figure 4-5). Hence, it is identified that the metric selection differs depending on the process they are using in their projects, as illustrated in Figure 4-6 and Figure 4-7.

Table 5.1 shows subject matter experts' point of view of relevant metrics. It gives a better idea of why some metrics can be used in both TSD process and ASD process and why some metrics cannot be used in some processes.

Table 5.1: Subject matter experts' point of view.

#	Metric		Subject matter experts' point of view
1	Delivery on time		Provides a good indication of whether the scope being managed or understood.
			Can used for future prediction purposes.
			Assist to track project progress.
2	Targeted Value Increase (TVI+)		What actually concern about actual against planned.
			This metric does not bring value to the project or company.
			If product backlog not groomed properly hard to measure the metric.
3	Unit test coverage for the developed code		Contributes to maintain the quality of the product.
			Reduce the time for re-testing.
			Tool can be used to automate the unit test cases.
4	Bug correction time from	Negative (Experts from process only	Not a problem as long as defect being closed.
			More towards to measure individual level not team

	“new” to “closed” state	companies)	level.
		Positive (Experts from ASD and TSD process companies)	Important to use in the ASD projects as well.
			Better for pure maintenance of defect fixing project where product backlog is refreshed every time.
			Measure the defect fixing efficiency of the dev team.
			Used for future planning and estimations.
5	Open defect severity index		Contribute to maintain the quality of the product.
			Make sure to maintain the “0” count of defects at the end of each sprint.
6	Requirement Clarity Index		Due to more communication, use of this metric does not bring any value.
			This is often checked in sprint planning and stand ups.
7	Defect Density		Does not concern as long as defects were closed for the sprint.
			Depends on the project type.
8	Defect Removal Efficiency		Does not concern as long as defects are closed for the sprint.

Relationship between tools used in the ASD process against metric usage

JIRA/Greenhoper was the most used tool (Figure 4-28). Microsoft Excel was the second most used tool. In accordance with the responses, they are using Microsoft Excel as a support tool. Organizations more towards ASD process are using tools, which are specific for Agile environment with the Microsoft Excel as a support tool (see Figure 4-29). Companies into both the ASD and TSD process were mostly using Microsoft Excel tool with other support tools (Figure 4-30). As listed in the Section 4.8, Bugzila and HP Quality Center tools were also used along with the JIRA/Greenhopper tool. This had happened due to a client request, and sometimes due to the lack of knowledge about the use of JIRA/Greenhopper tool.

Out of the eight metrics introduces by Downey and Sutherland (2013), there were five metrics (*Velocity*, *Work Capacity*, *Focus Factor*, *Percentage of Adopted Work*, and *Percentage of Found Work*) which got more than 75% response. *Targeted Value Increase* (*TVI+*) metric was rejected with the responses of 35%. The four metrics (*Bug correction time from “new” to “closed” state*, *Delivery on time*, *Unit test coverage for the developed code*, and *Open Defect Severity Index*) received more than 75% responses metrics introduced from Mannila (2013) findings, and the rest of metric (*Technical debt*) also got 65% of responses. Metrics gathered from face to face interview, only one metric (*Sprint-level effort burndown*) got an 81% responses. The rest of the metrics got less than that (Table 5.2).

Table 5.2: Metric distribution with its usage.

#	Metric Title	Reference	% of metric usage from responses
1	Delivery on time	(Mannila, 2013) /Interview	90%
2	Work capacity	(Downey and Sutherland, 2013) / Interview	88%
3	Unit test coverage for the developed code	(Mannila, 2013) /Interview	88%
4	Percentage of adopted work	(Downey and Sutherland, 2013)	81%
5	Bug correction time from “new” to “closed” state	(Mannila, 2013) /Interview	81%
6	Sprint-level effort burndown	Interview	81%
7	Velocity	(Downey and Sutherland, 2013) / Interview	79%
8	Percentage of found work	(Downey and Sutherland, 2013)	79%
9	Open defect severity index	(Mannila, 2013) /Interview	79%
10	Focus factor	(Downey and Sutherland, 2013) / Interview	77%
11	Cost of quality	Interview	69%
12	Defect severity index	Interview	67%
13	Technical debt	(Mannila, 2013) /Interview	65%
14	Defect slippage rate	Interview	63%
15	Customer satisfaction survey	Interview	60%
16	Accuracy of estimation	(Downey and Sutherland, 2013)	58%
17	Accuracy of forecast	(Downey and Sutherland, 2013)	54%
18	Net promoter score	Online document	54%
19	Requirements clarity index	Interview	54%
20	Defect density	Interview	54%
21	Defect removal efficiency	Interview	52%
22	Targeted value increase (TVI+)	(Downey and Sutherland, 2013)	35%

Research findings against research objectives

Table 5.3 depicts the mapping of the research findings with research objectives. In that, it separately mentioned about the additional findings of the research.

Table 5.3: Mapping of research findings with research objectives.

Research objectives	Research findings
Identify set of software metrics suitable for the ASD process	Ten metrics identified and categorized in to 3 different sections (Table 5.3) • Product quality o Unit test coverage for the developed code o Bug correction time from "new" to "closed" state o Open Defect Severity Index • Team productivity o Work Capacity o Percentage of adopted work o Velocity o Sprint-level effort burndown o Percentage of found work • Predictability o Delivery on time o Focus factor
Identify the software metric usage in projects based on the ASD process	Selected twenty two metrics usage in ASD projects was identified (Table 5.2) • Delivery on time metric is the most used metric • Targeted Value Increase (TVI+) metric is the least used metric • Delivery on time, Unit test coverage for the developed code, Bug correction time from "new" to "closed" state, and Open defect severity index metrics highly used in both ASD and TSD process. • Work capacity, Percentage of adopted work, Sprint-level effort burndown, Velocity, Percentage of found work, and Focus factor are Agile-specific metrics mostly used in ASD process. • Requirement clarity index, Defect density, and Defect removal efficiency metrics got 54%, 54% and 52% of percentage of metric usage, respectively. These metrics identified as most used in the TSD process but comparatively less usage in ASD process.
Identify the benefits of use of software metric in projects based on ASD process	Benefits of software metric usage was identified (Table 4.4) • Track the project progress and project health • Monitor the quality aspect of the project or product • Helps team to forecast and manage the project better • Identify the areas needs to be improved • Improve the estimation • Improve the development process

	• Brings alignment to across cross-functional teams working in thesame project
Additional findings	
Better to use JIRA/Greenhopper tool with supportive tools in ASD projects. Following metrics can be measured directly and indirectly using JIRA/Greenhopper tool (Table 5.4) • Directly ○ Work Capacity ○ Adopted work ○ Sprint-level Effort Burndown ○ Velocity ○ Found work • Indirectly ○ Focus Factor ○ Open defect severity index ○ Unit test coverage for the developed code ○ Bug correction time from “new” to “closed” state	

5.2 Recommendations

For companies having projects on the ASD process software metrics can be used to measure the quality of the product and team productivity and predictability. Metrics with the highest frequency of responses (more than 75%) from the research survey are recommended as the metrics to be used in projects based on ASD process. Those metrics are listed in Table 5.3. These metrics may be further categorized into sections as product quality, team productivity and predictability based on the follow-up interviews conducted by subject matter experts.

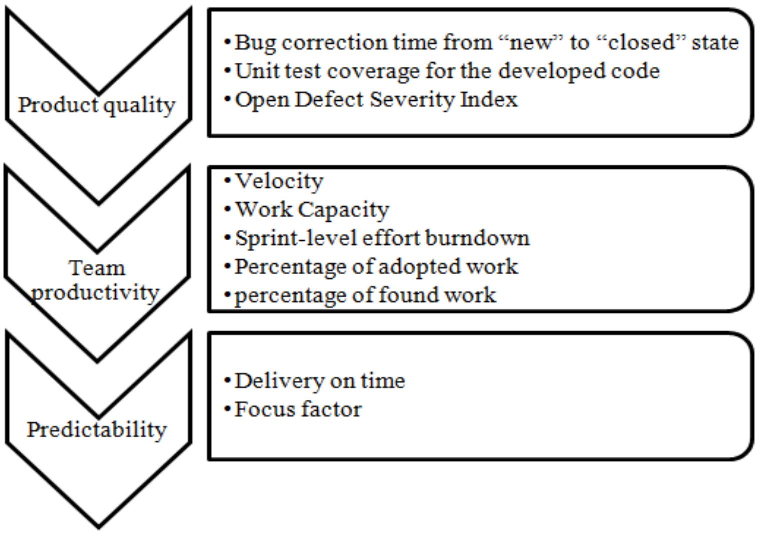

Product quality
•Bug correction time from “new” to “closed” state
•Unit test coverage for the developed code
•Open Defect Severity Index
Team productivity
•Velocity
•Work Capacity
•Sprint-level effort burndown
•Percentage of adopted work
•percentage of found work
Predictability
•Delivery on time
•Focus factor

Figure 5-1: Recommended metrics to use in the ASD process

It was better to use JIRA/Greenhopper tool with supportive tools. By using JIRA/Greenhopper tool metrics listed in Table 5.4 can be measured directly and indirectly.

Table 5.4: Metrics that can be measure using JIRA/Greenhopper tool

Directly	Indirectly
Work capacity	Focus factor
Percentage of Adopted work	Open defect severity index
Sprint-level effort burndown	Unit test coverage for the developed code
Velocity	Bug correction time from “new” to “closed” state
Percentage of Found work	

5.3 Future Work

While this research identify a set of metrics which are appropriate for the ASD process and their perceived benefits, further research need to be conducted on the use of software metrics in ASD-based projects. It is important to come up with a set of standards and guidelines for Agile practitioners to follow as a further continuation of this research. There should be a user manual with the clear description of metrics, how they can use, in what kind of a situation they can use, standard definition, etc. The research findings of this research can be further evaluated based on the geographical situation of the company or type of the company, whether the company is product development or service delivery company. The set of metrics found in this research can be further analysed by evaluating the relationship between the use of metrics in Agile base projects and success and failure of the project. As this research focused more of a qualitative analysis, it is better to conduct quantitative analysis in a wider range. Case base research can be conducted by practically applying the findings of this research.

5.4 Conclusions

This research helped to identify the significant metrics and their usage in projects based on Agile software development process in the Sri Lankan software development industry. It also identifies the benefits of software metric usage in an Agile software development process. Based on survey results *Delivery on time* was the most used metric and *Targeted Value Increase* (*TVI+*) was the least used metric. *Work capacity*, *Percentage of adopted work*, *Sprint-level effort burndown*, *Velocity*, *Percentage of found work*, *Focus factor*, *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, *Open defect severity index* metrics were the most rated Agile-specific metrics. *Delivery on time*, *Unit test coverage for the developed code*, *Bug correction time from "new" to "closed" state*, *Open defect severity index* metrics can be used in both TSD process and ASD process. While *Requirement Clarity Index*, *Defect density,* and *Defect removal efficiency* metrics are used in the TSD process, they are hard to use in ASD process. As per the subject experts' comments, *Requirement clarity index* metric does not bring any value to the project, since the requirements clarified at the sprint grooming sessions or daily stand-ups. *Defect density* and *Defect removal efficiency* does not concern as long as defects were closed for the sprint. ASD process only companies using their own Agile specific metrics, which were identified while conducting face-to-face interviews. *Thumbs-up rule*, *None compliance index*, *Top hill view* were those three metrics. *Thumbs-up rule* metric use to measure the customer satisfaction at the end of each sprint. *Non compliance index* is used to check the project's compliance as per their company standards. *Top hill view* is used to track the project progress.

When considering the Agile specific tools JIRA/Greenhoper was the most used tool out of all the responses. Microsoft excel was the second best tool. JIRA/Greenhopper tool was most popular with the companies who have projects based only on ASD process. At the same time, companies were using supportive tools. Whereas the rest of the companies, the most used tool was the Microsoft Excel with other supportive tools (Figure 4-30). Finally, set of metrics was recommended and it was further classified as relating to product quality, team productivity, and predictability.

Use of metric depends on how the organization environment is set up and how they measure the quality of the product. When selecting the metric for the project, it should be considered about the metric value brings to the project or the company. The use of metric was not an extra burden. It was part of the process. Selection of the metric for the project was the very important part. Therefore, metrics should be selected very carefully with the help of subject matter experts. Automated metric or use of a tool to measure metric would reduce the risk of the team being burdened.

References

Abran, A., Qutaish, R. A., Desharnais, J.-M., and Habra, N. (2005). An Information Model for Software Quality Measurement with ISO Standards. (pp. 104-116). International Conference on Software Development, Reykjavik, Islande , University of Iceland.

Agile Metrics- Let the Numbers tell the Tale. (2013, 12 5). Retrieved from http://www.scrumdayeurope.com/: http://www.scrumdayeurope.com/prowareness/website/scrumday.nsf/Agile_Metrics.pdf

Aktunc, O. (2012). Entropy Metrics for Agile Development Processes. Dallas: Software Reliability Engineering Workshops (ISSREW), 2012 IEEE 23rd International Symposium on.

Al-Qutaish, R. E. (2009). Measuring the Software Product Quality during the Software Development Life-Cycle: An International Organization for Standardization Standards Perspective. *Journal of Computer Science, 5*(5), 392-397.

Beck, K., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., . . . Thomas, D. (2001). *http://agilemanifesto.org/.* Retrieved 10 12, 2013, from http://agilemanifesto.org/: http://agilemanifesto.org/

Benefield, G. (2008). Rolling Out Agile in a Large Enterprise. Waikoloa: Hawaii International Conference on System Sciences.

Boone, H. N.,and Boone, D. A. (2012). Analyzing likert data. *Journal of Extension*, 50.

Cockburn, A. (2000). *Agile Software Development.*

Downey, S., and Sutherland, J. (2013). Scrum Metrics for Hyperproductive Teams:. *System Sciences (HICSS), 2013 46th Hawaii International Conference on.* Hawaii: IEEE.

Dyba˚, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), pp.833-59.

Eeger, S. L. (2008). Software Metrics: Progress after 25 Years? *Software, IEEE, 25*(6), 32 - 34.

Emam, K. E. (2002). A Methodology for Validating Software Product Metrics. (p. 39). NRC Publications Archive (NPArC).

Fathi, P., and Morovat, A. (2013). The Role of Scrum Methodologies in the Design and Production Software . *Journal of Basic and Applied Scientific Research* , 57-61.

Fenton, N. E., and Neil, M. (1998). Software metrics: success, failures and new directions. *Systems and Software*(2-3), 149-157.

Fenton, N. E., and Neil, M. (2000). Software Metrics: Roadmap. (p. 370). Proceedings of the conference on The future of Software engineering.

Glib, T. (1977). *Software Metrics.* Winthrop Publishers.

Griffiths, M. (2013, August 28). *9 Ways PMOs Can Help Agile Projects.* Retrieved September 3, 2013, from http://leadinganswers.typepad.com/leading_answers/

Gustafsson, J. (2011). Model of Agile Software Measurement: A Case Study. *Master of Science Thesis in the Programme Software engineering, Chalmers University of Technology, Sweden*.

Javdani, T., Zulzalil, H., Abd Ghani, A. A., Md Sultan, A. B., and Parizi, R. M. (2012). On the Current Measurement Practices in Agile Software Development. *International Journal of Computer Science Issues (IJCSI);, Vol. 9*(4,), p127.

Jones, C. (2008). Measuring defect potentials and defect removal efficiency. *CrossTalk The Journal of Defense Software Engineering 21.6*, (pp. 11-13).

Kan, S. H. (2002). *Metrics and Models in Software Quality Engineering*

Kaur, A., and Singh, G. (2004). *Quality Methodologies for software.* Concordia : Concordia Institute for Information System Engineering.

Kayes, I., Sarker, M., and Chakareski, J. (2013, October 09). Product Backlog Rating: An Empirical Study On Measuring Test Quality in Scrum.

Kitchenham, B. A. (1996). *Software Metrics: Measurement for Software Process Improvement.* NCC Blackwell.

Kulas, H. (2004). Product metrics in agile software development. Master's Thesis, Univ. of Tampere,Finland.

Kunz, M., Dumke, R. R., and Schmietendorf, A. (2008). How to Measure Agile Software Development. Springer Berlin Heidelberg.

M T Sletholt, J. E. (2011). A Literature Review of Agile Practices and Their Effects in Scientific Software Development. (pp. 1-9). New York: SECSE '11 Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering.

Malaiya, Y. K. (1998). Estimating defect density using test coverage. *Rapport Technique CS-98-104, Colorado State University* .

Mack, N., Woodsong, C., MacQueen, K. M., Guest, G., and Namey, E. (2005). Qualitative research methods: a data collectors field guide.

Mannila, J. (2013). *Key performance in Agile software development.* Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences: Degree Programme in Information Technology.

Mindlance. (n.d.). *Mindlance.* Retrieved August 28, 2013, from http://www.mindlance.com/documents/test_management/testing_metrics.pdf

Mordal, K., Anquetil, N., Laval, J., Serebrenik, A., Vasilescu, B., and Ducasse, S. (2012). Software Quality Metrics Aggregation in Industry. *JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE*, 1–19.

Oza, N., and Korkala, M. (2012). Lessons Learned In Implementing Agile Software Development Metrics. UK Academy for Information Systems Conference Proceedings 2012.

Purcell, J.E., 2007. *Comparison of Software Development Lifecycle Methodologies*. Information Systems Security Professional Consortium, Inc.

Rawat, M. S., Mittal, A., and Dubey, S. K. (2012). Survey on Impact of Software Metrics on Software Quality. International Journal of Advanced Computer Science and Applications .

Reiner R. Dumke, A. S. (2008). Software Metrics for Agile Software Development. Perth: Software Engineering, 2008. ASWEC 2008. 19th Australian Conference.

Rentrop, J. (2006, August 31). Software Metrics as Benchmarks for Source Code Quality of Software Systems. *Master Course Software Engineering, University Amsterdam*.

Rico, D. F. (2008). What is the ROI of Agile vs. Traditional Methods. *TickIT International 10*, (pp. 9-18).

Rising, J. (2009, 06 06). *http://www.managedmayhem.com/2009/05/06/agile-software-development-process/*. Retrieved 11 05, 2013, from http://www.managedmayhem.com: http://www.managedmayhem.com/2009/05/06/agile-software-development-process/

Royce, W. (1970). Managing the Development of Large Software Systems. *IEEE. WESCON*, (pp. 1-9).

Schwaber, K. & Sutherland, J., 2013. *The Scrum guide*. Scrum.org.

Scott. (2013, 11 12). *Scrum Alliance certification*. Retrieved from Rapid Scrum: http://www.rapidscrum.com

Siakas, K. V., Georgiadou, E., and Berki, E. (2005). Agile Methodologies and software process improvement. *26.* IADIS (International Association for development of the Information Society) International Virtual Multi Conference on Computer Science and Information Systems (MCCSIS 2005)-SEA (Software Engineering and Applications).

Sri Lanka the hub of Asia. (2013, 12 22). Retrieved from http://www.srilankaexpo.com/: http://www.srilankaexpo.com/index.php/products-a-services/export-of-services/information-a-communication-technology-ict-profile

Sutherland, J. (2009). Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams. (pp. 1 - 8). Big Island, HI: System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference.

Tallinn, H., Aavik, H., Virkus, M., and Liive, R. (2013). Using Scrum in a side project with distributed teams. *University of Tartu Viljandi Culture Academy, Master Thesis*.

Trivedi, P., and Kumar, R. (2012). Software Metrics to Estimate Software Quality using Software Component Reusability. *International Journal of Computer Science Issues (IJCSI), Vol. 9*(Issue 2), p144.

VersionOne. (2012). *7 th Annual state of Agile Development Survey.* Atlanta: VersionOne.com.

Xabia group. (2012). Xabia Agile survey 2012., (p. 30).

Zoysa, L. D. (2011). Software Quality Assurance in Agile and Waterfall Software Development Methodologies: A Gap Analysis. *Master of information system management, University of Colombo*.

APPENDIX A

FACE TO FACE INTERVIEW QUESTIONNAIRE

1. Your current organization
2. Software development processes used in your company is/are;
3. If your company following SEI Capability Maturity Model, specify its level
4. If your company following any ISO standards what are they?
5. Number of people engaged in Agile software development process at your company
6. Have you followed/completed any Agile focus training/certification
7. What is/are the Agile techniques practiced at your project
8. How many years have your company been practicing agile development methods?
9. How many years have you engaged with agile development methods at your company?
10. Number of team members in a proejct?
11. What is the duration of a sprint in calender days in your project?
12. What is the duration of a release in calender days in your project?
13. Specific Agile tools used at your project
14. Which of the following best describes your knowledge level about Agile Metrics
15. Which of the following best describes your team members knowledge level about Agile Metrics
16. Which of the following best describes your knowledge level about Agile Development Methodology
17. Which of the following best describes your team members knowledge level about Agile Development Methodology
18. Which of the following best describes your role in Agile project(s)?
19. Do you conduct retrospectives?
20. How many individual projects are run using agile practices within your organization currently?

21. What percent (%) of your company's projects use an agile methodology currently?
22. Failing project percentage (%) which uses an agile methodology as well as metrics?
23. Is your team trained for Agile process?
24. Are you using metrics (Ex. Burdown charts, Velocity, Work capacity, Customer satisfaction... etc) in each Agile practice projects?
25. If you are using metrics at your projects;
26. Do you prefer Agile development method over the other software development methods?
27. What is your idea about metrics using in Agile development process

APPENDIX B

FEEDBACK FOR PRELIMINARY QUESTIONNAIRE

The followings are the summary of the suggestion made by the professionals who goes through the online survey questionnaire.

1. It is better to make simple and understandable the help text used as metrics.

2. If it is possible add the metrics into grid format which will help to avoid the complexity.

3. Add a text under each metric to get the information on the status to collect the information when they reject the metric. This will helps to get the real idea of metrics rejection.

4. Some questions order needs to be changed

APPENDIX C

ONLINE SURVEY QUESTIONNAIRE

Study on use of Software Metrics in Agile Software Development process

Dear Sir/Madam,

As part of my MBA in IT project at the University of Moratuwa, I am conducting a survey to analyse the use of software metrics in Agile software development projects. As a member of the Sri Lankan software engineering community, we are inviting you to participate in this study by completing the following survey. It will take about ~15 minutes to complete.

The data gathered from the survey will only be used for the thesis requirement of the MBA in IT, at the University of Moratuwa, Sri Lanka. This survey is stipulated confidential and anonymous. Your responses will not be identified with you personally and all findings will appear in aggregated form. You and your organization will not be linked in any manner.

Your participation in the research would be greatly appreciated. If you have any suggestions, would like more clarification about the questions, or how the data will be used, please feel free to contact us using the details provided below.

Survey
Link: https://docs.google.com/forms/d/1fsCLJ_67T5QLxmnhSluPw9tKqpt3CuCXNO4oy_pMiqc/viewform

Thank you very much for your time and help in making this study possible.

K.V.Jeeva Padmini and Dilum Bandara

jeevapadmini@gmail.com, dilumb@cse.mrt.ac.lk

MBA (IT) 2012

Dept. of Computer Science and Engineering,

Section A- Use of Metrics in projects that practice Agile Software Development process

If you have not completed any project using Agile Software Development process yet, think of the most recent sprint where your team completed all tasks in the sprint and delivered the shippable product increment. How frequently does your team use the following metrics in Agile Software Development projects?

*Required

Velocity *

Sum of original estimates of all completed work in the sprint

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Work Capacity *

Sum of all actual work committed during the Sprint

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Focus Factor *

(Sum of completed work within the sprint) ÷ (Sum of committed work within the sprint)

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Adopted Work *

Additional story points taken from product backlog for the sprint

○ Used in Every Project

○ Used in Some Projects

○ Hardly Used in Projects

○ Never Used in Projects

○ Not Applicable

Found Work *

Work associated with completed work, which needs to be completed within the sprint

○ Used in Every Project

○ Used in Some Projects

○ Hardly Used in Projects

○ Never Used in Projects

○ Not Applicable

Sprint-level Effort Burndown

Effort Burndown Chart created for relevant sprint

○ Used in Every Project

○ Used in Some Projects

○ Hardly Used in Projects

○ Never Used in Projects

○ Not Applicable

Technical dept (Design debt or Code debt) *

Shortcuts implemented and left to the code that requires later re-factoring for getting the code working well

○ Used in Every Project

○ Used in Some Projects

○ Hardly Used in Projects

○ Never Used in Projects

○ Not Applicable

Bug correction time from "new" to "closed" state *

○ Used in Every Project

○ Used in Some Projects

○ Hardly Used in Projects

○ Never Used in Projects

○ Not Applicable

Open Defect Severity Index *

An index representing the average of the severity of the defects

- ○ Used in Every Project
- ○ Used in Some Projects
- ○ Hardly Used in Projects
- ○ Never Used in Projects
- ○ Not Applicable

Accuracy of Estimation *

(Sum of original Estimates for the sprint) ÷ (Sum of actual effort for the sprint)

- ○ Used in Every Project
- ○ Used in Some Projects
- ○ Hardly Used in Projects
- ○ Never Used in Projects
- ○ Not Applicable

Accuracy of Forecast *

Sum of original Estimates for the sprint ÷ (Sum of original Estimates + Sum of adopted Work + Sum of found Work)

- ○ Used in Every Project
- ○ Used in Some Projects
- ○ Hardly Used in Projects
- ○ Never Used in Projects
- ○ Not Applicable

Defect Removal Efficiency *

Total Number of company found, product Defects in the client release * 100 ÷ (Total Number of company found, product Defects in the client release + Total Number of Client Reported product Defects in the client release)

- ○ Used in Every Project
- ○ Used in Some Projects
- ○ Hardly Used in Projects
- ○ Never Used in Projects
- ○ Not Applicable

Targeted Value Increase (TVI+) *

Current Sprint’s Velocity ÷ Original Velocity

- ○ Used in Every Project
- ○ Used in Some Projects
- ○ Hardly Used in Projects

○ Never Used in Projects
○ Not Applicable

Delivery on time *
Ratio of features done in the planned release schedule

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Net promoter score *
How likely customer recommend your organization to others

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Unit test coverage for the developed code *

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Requirements Clarity Index *
Each of the requirements on a scale for internal releases ÷ Number of requirements for internal releases

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Defect Density *
Number of defects per story point of the sprint

○ Used in Every Project
○ Used in Some Projects

○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Defect Severity Index *
Number of defects added during the sprint

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Cost of Quality *
Sum of costs incurred in maintaining acceptable quality levels

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Defect Slippage Rate *
Number of production defects

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Customer Satisfaction Survey *
Customer Delight Index

○ Used in Every Project
○ Used in Some Projects
○ Hardly Used in Projects
○ Never Used in Projects
○ Not Applicable

Do you see any benefits in the use of metrics at the Agile Software Development process? *
Please explain

Is your project(s) using any Customer-Specific or Project Specific metrics? If so, please specify.

Especially the ones not mentioned above

*Required

Section B- General Information

How long have you been involved in Agile Software Development process? *

○ 0-1 year

○ 1-2 year

○ 2-3 Year

○ 3-4 year

○ 4-5 year

○ More than 5 years

How knowledgeable are you about the Agile Software Development process *

○ Poor

○ Below Average

○ Average

○ Above Average

○ Excellent

How knowledgeable are you about metrics used in the Agile Software Development process *

○ Poor

○ Below Average

○ Average

○ Above Average

○ Excellent

Have you followed or completed any Agile Software Development process specific trainings or certifications?

If so, please specify

Currently, how many individual projects are based in Agile Software Development process within your organization?

What percentage (%) of your company’s projects use an Agile Software Development process currently?

What are the Software development processes used in your company? *

□ Scrum Development Process
□ Scrum/XP Hybrid Development Process
□ Kanban
□ Scrumban
□ Feature-Driven Development
□ Extreme Programming XP
□ Waterfall Development Process
□ Iterative-Incremental Development Process
□ Spiral Development Process
□ Other:

What are the common Agile practices followed in a project? *

□ Daily Stand-up
□ Sprint Planning
□ Release Planning
□ Retrospectives
□ Working Product
□ Analog or Digital(e.g., JIRA Agile Classic Taskboard) Taskboard
□ Sprint Demo
□ Other:

Number of team members in a project? *

□ 1-5 Members
□ 6-12 Members
□ 13-24 Members
□ 25-30 Members
□ More than 30 Members

What tools are used to manage the projects based in Agile Software Development process? *

□ Bugzilla
□ Microsoft Excel
□ HP Quality Center
□ JIRA/Greenhopper
□ Microsoft Project
□ TFS (Team Foundation Server)
□ Code Collaborator
□ Sonar Tool
□ GoodData
□ Planning poker
□ Other: □

What is the typical duration of a sprint in calendar days? *

○ 1 weeks
○ 2 weeks
○ 3 weeks
○ 4 weeks
○ Other: □

What is the duration of a release in calendar days in your project? *

○ 2 weeks
○ 3 weeks
○ 4 weeks
○ 6 weeks
○ Other: □

Which of the following best describes your role in Agile project(s)? *

○ Scrum master
○ Product owner

○ Team member -Developer
○ Team member -QA
○ Customer
○ Other:

Do you conduct retrospectives? If yes, when? *

It would be a great help if you can provide further feedback by participating in a follow up interview. Please provide your current organization and contact mail address if you wish to participate in follow up interview. We respect your privacy and will not use contact information indiscriminately.

Organization Name :

Contact e-mail address:

APPENDIX D

FOLLOW UP INTERVIEW QUESTIONS

1. While Agile means less documentation why everybody like to use metrics in a project environment? Isn't it an extra burden?

2. Why do you think "Delivery on Time" metric is used frequently?
   It got the heights number of responses from online survey.
   (Delivery on Time: - Ratio of features done in planned release schedule)

3. Why do you think Targeted Value Increase (TVI+) got the least amount of responses out of 22 metrics used in the research survey, since it used to measure agile feature?
   (Targeted Value Increase: - Current Sprint's Velocity ÷ Original Velocity)

4. While metrics heavily used in Traditional development process, following got the highest number of responses among agile development process.
   a. Unit test coverage for the developed code
   b. Bug correction time from "new" to "closed" state
   c. Open defect severity index

   Will these metrics suitable for Agile Software Development process also?
   Can they measure easily in ASD process?
   What is the purpose of using above mention metrics?
   a. ……………………………………………………………………………
   b. ……………………………………………………………………………
   c. ……………………………………………………………………………

5. Responses show that 'Net Promoter Score' metric got an less number of responses. Do you think it is a good metric to be used in an agile development process?

(Net Promoter Score: - How likely customer recommends your organization to others)

'Net promotor score' metric got maximum number of 'Not Applicable' responses as well. Do you think is it due to lack of knowledge about the metric?

6. Following metrics were heavily used in the Traditional development process. But got a lesser number of responses for usage in an agile development process.
   a. Customer Satisfaction Survey
   b. Defect slippage rate
      (:-Number of production defects)
   c. Defect Removal Efficiency
      (:-Total Number of company found, product Defects in the client release * 100 ÷ (Total Number of company found, product Defects in the client release + Total Number of Client Reported product Defects in the client release)
   d. Defect Density
      (:-Number of defects per story point of the sprint)
   e. Requirement Clarity Index
      (:-Each of the requirements on a scale for internal releases ÷ Number of requirements for internal releases)

   Is there any specific reason why we can't use these metrics in an agile development process?

7. Following metrics were heavily used in the Traditional development process. Also, these metrics fairly used in an agile development process also.
   a. Defect Severity Index (Number of defects added during the sprint)
   b. Cost of Quality (Sum of costs incurred in maintaining acceptable quality levels)

Is there any specific reason why we can’t use these metrics in an agile development process?

8. Why do you think people should use the Scrum development process?
   There are a lot of Scrum variations. But most of them were using the Scrum development process. Do you see why it is?

9. What did you discuss during the sprint retrospective?

10. Most of the respondents were not using Analog or Digital(e.g., JIRA Agile Classic Taskboard) Taskboard. Professionals who are using JIRA/ TFS also mentioned that they were not using Analog or Digital(e.g., JIRA Agile Classic Taskboard) Taskboard. Why do you think they were not using that?

11. Do you think it is better to use a tool which support in the scrum development process?

12. Most used tool was the JIRA/Greenhopper (28, 58%). Do you think it is a good tool to be used in ASD process. If so why?

13. The second best tool was Microsoft Excel (23, 48%). But for what purpose people are using Microsoft excel, since it is not that advanced and properly created as a tool for Agile development process. Why people like to use Microsoft Excel as a too in ASD process?

14. Why people are using bugzilla, since it is only used for bug tracking? Even when there are tools which help to manage almost overall Agile development process.

APPENDIX E

METRIC DESCRIPTION

#	Metric Title	Metric Description	How to measure	Reference
1	Accuracy of Estimation	1-(∑(Estimate Deltas) ÷ Total Forecast) ,The Team’s ability to correctly estimate the body of work during Sprint Planning	∑Original Estimates for the sprint/ ∑Actual effort for the sprint	(Downey and Sutherland, 2013)
2	Accuracy of Forecast	(∑Original Estimates) /∑ (∑Original Estimates + ∑Adopted Work + ∑Found Work)	∑Original Estimates for the sprint /∑ (∑Original Estimates + ∑Adopted Work + ∑Found Work)	(Downey and Sutherland, 2013)
3	Cost Of Quality	Percentage of effort spent on all activities other than core development activities such as requirements, architecture, design and coding (including CM) over the total project effort	Cost of Quality = Total Quality Effort [Submitted Time] in the internal release * 100 / Total Effort [Submitted Hrs] in the internal release	Interview
4	Customer satisfaction survey	Customer Delight Index		Interview
5	Defect Density	Number of defects per story point of the sprint		Interview
6	Defect Removal Efficiency	To understand the efficiency of the team in detecting the defects induced in to the system and take necessary steps to improve the same	Total Number of company found, product Defects in the client release * 100 ÷ (Total Number of company found, product Defects in the client release + Total Number of Client Reported product Defects in the client release)	Interview
7	Defect Severity Index	Number of defects added during the sprint		Interview
8	Defect slippage rate	Number of production defects		Interview
9	Delivery on time	Ratio of features done in planned release schedule		(Mannila, 2013)/Interview
10	Bug correction time from “new” to “closed” state	Time from “new” to “closed” state, internal faults		(Mannila, 2013)
11	Focus Factor	Velocity ÷ Work Capacity		(Downey and Sutherland, 2013) / Interview
12	Net Promoter Score	Measurement of customer satisfaction	How likely are you recommending to this team/organizatioin to a friend or colleague? 0-6 - Detractors 7-8 -Nueatural Customers 9-10 -Promotors	Online document
13	Open Defect Severity Index	Number of open defects at the time of sprint end.		(Mannila, 2013)/Interview

14	Percentage of Adopted Work	∑(Original Estimates of Adopted Work) ÷ (Original Forecast for the Sprint)	Additional story points for sprint taken from product backlog	(Downey and Sutherland, 2013)
15	Percentage of Found Work	∑(Original Estimates of Found Work) ÷ (Original Forecast for the Sprint)	Work associated with forecast work which needs to be completed within the sprint	(Downey and Sutherland, 2013)
16	Requirements Clarity Index	Average of Requirement Clarity Index (RCI) measured for each of the requirements on a scale of 1 to 5	Sum of Requirement Clarity Index (RCI) measured for each of the requirements on a scale of 1 to 5 for internal releases / Number of requirements for internal releases	Interview
17	Sprint-level effort burndown	Effort burndown Chart developed for relevant sprint		Interview
18	Targeted Value Increase (TVI+)	Current Sprint's Velocity ÷ Original Velocity	Current Sprint's Velocity ÷ Original Velocity	(Downey and Sutherland, 2013)
19	Technical debt	Short-cuts implemented and left to the code that require later re-factoring for getting the code working well.		(Mannila, 2013)/Interview
20	Unit test coverage for the developed code			(Mannila, 2013)/Interview
21	Velocity	∑ of original estimates of all accepted work	∑ of original estimates of all Completed work in the sprint	(Downey and Sutherland, 2013) / Interview
22	Work Capacity	The sum of all actual work reported during the Sprint, whether the SBI toward which the work was applied finished or not.	∑ of all actual work committed during the Sprint	(Downey and Sutherland, 2013) / Interview