Application of Lean Principles to IT Management

Damayanti Anand | 1737230

MBA Business Management
Dublin Business School
(School of Business and Law)

August 2014

Table of Contents

List of Tables and Figures

Acknowledgements

I would like to begin by thanking my Research Supervisor, Patrick O'Callaghan and my Research Methods Lecturer, P.J. Paul. Patrick guided me throughout the dissertation period and helped me achieve the direction that I needed to go ahead with the research. P.J. Paul guided me through the methodology and provided valuable feedback of my Research Proposal. I would also like to acknowledge my mother, Jaba Basu for her support and encouragement and my father, Anand Kumar Keshavan for his guidance in the subject area. I also have to acknowledge my former employer SpiderLogic India Pvt. Ltd., their Director, Gautam Kasturi and my colleague and team mate, Vinit Gulati, who had faith in me and taught me about the Software Development Life Cycle, without which I would not have been able to undertake this research.

Abstract

Lean is a management principle that originated in the company Toyota in the 1980s and since then, has been used by manufacturing companies throughout the world. It has had positive effects on the manufacturing industry: reduced costs and increased the speed of production. This Thesis aims to derive whether lean can also have similar positive effects on the IT industry, specifically software delivery. The study uses the philosophy of positivism and a deductive approach in order to conclude whether lean principles make software delivery efficient. Data has been collected using a case study as well as surveys and a quantitative method has been applied to analyse the data. In order to quantify efficiency, the parameters being studied are cost of the projects and time taken to deliver. The research findings have drawn a correlation between the use of lean principles and the cost and speed of projects. The results show that the use of lean principles decreases the cost of projects and enables teams to ship their software faster.

Chapter 1

Introduction

Chapter 1
Introduction

Introduction

Lean is a management principle that originated in the manufacturing industry. The basis of lean principles is that an organization should focus on the wants and needs of the customer. Any task or process that does not benefit the customer is considered waste, and waste should be eliminated as far as possible. Eliminating waste will help to reduce costs, while providing the best products and services to the customer. The use of lean principles has made the manufacturing industry more efficient by reducing costs, reducing the time taken for production and by improving the overall customer satisfaction (Reece and Antosiak, 2014). The aim of this thesis is to derive whether lean principles can have similar positive effects when applied to the IT industry.

The IT Sector is not an independent entity, but in fact, a backbone to every other industry in the economy. Every industry relies on Information Technology (IT) for smooth functioning. A logistics or supply chain company requires a very strong IT system without which the company will not be able to carry out its core functions. The same holds true for a finance company, or a manufacturing company (Russell, 2013). In the era of Digital Marketing, IT is an important aspect in the field of Marketing and Advertising as well. Each of these sectors depends on fast and reliable IT systems in order to achieve their goals, provide results and even measure their performance (Grensing-Pophal, 2014). Therefore, it can be derived that the IT Industry has a direct impact on the entire industry as well as the economy of a nation and the world. However, the IT sector faces some problems in the context of management and requires principles and methodologies that can improve the management and bring about positive results.

Before the problems faced in IT Management are discussed, it is necessary to discuss the set up of the IT sector. Schwalbe (2014) provides an introduction to the IT industry. When the term IT is used, it is a broad term that is a collection of software and hardware, including fields like software development, software testing, networking, infrastructure, data storage, etc. For this study, the research is focused on the Software Industry. The Software Industry is focused on developing new software systems and maintaining and upgrading currently used software systems. Usually, a software company develops custom software for an external client, or creates a software product that is sold in units to external clients. A team of Software Developers, Quality Assurance Analysts and Business Analysts, collectively referred to as the Development Team work together to develop the software and deliver it to the client. Each new software application or systems upgrade is undertaken by the development team in the form of a Project. Therefore, Project Management forms an important part of the management practices in the Software Industry.

1.1 Research Problem

In a world that has been affected by recession, companies strive to provide the best products or services to their customers quickly and at low costs. (Koury, 2014). Organizations spend large amounts of money on software systems for their business needs (Schwalbe, 2014). If a company spends a large amount on developing or outsourcing its software, it is vital for the company to get high quality software at the time that it is required for its business requirements. However, software delivery faces a lot of problems. Unlike manufacturing companies where processes are carried out by automated machines, the software industry is completely dependent on the skilled workforce. This makes standardization very difficult, results variable, software products and services unreliable, costs of projects high and software delivery not punctual (Elzamly and Hussin (2014).

The word 'inefficient' can be used to summarize the problems faced by the current software delivery processes. Companies have to strive to maintain high profitability even when the availability of resources is low. Thus, there is a need for a management principle that makes companies efficient.

1.2 Suggested Solution

When it comes to innovations in technology or business processes, the software industry has always been a pioneer. It is the one industry that has always been up to date with improving processes in order to work in the most efficient way possible (DeMarco, 2013). The suggested solution to the research problem is the use of Lean Principles to manage software delivery. The benefit of Lean principles to a company is that it can produce better quality products at lower prices by improving processes. Lean principles originated in Toyota in Japan after World War II when efficiency was extremely necessary. Lean principles benefited Toyota so clearly that manufacturing companies all over the world implemented them (Liker, 2004). Recently, the lean principles have been adapted to suit the software industry in order to make software delivery more efficient and to help companies gain a competitive advantage. Lean principles propagate process improvements and waste removal in order to provide high quality software at low prices (Bell and Orzen, 2010). However using lean principles in software delivery is a new area of study and research needs to be conducted to find out whether lean principles do have a positive effect on the software industry.

Applying lean principles to software delivery is relevant to the software industry worldwide, but specifically to Ireland where cost of employment of software development teams is higher than that of countries like India and China (Ireland: Brightwater, 2014; China: Robert Walters, 2014; India: Kelly Services, 2014). In order to compete with lower costs of development, Irish software companies require process improvements that help them to optimize their investments.

1.3 Research Question

Looking at the research problem, the researcher has identified that the software industry requires efficiency in order to gain competitive advantage. Competitive advantage can be gained if software of a high quality is delivered within the deadline and at lower costs. Companies in the manufacturing industry have used lean principles successfully in order to gain a competitive advantage, and therefore the researcher feels the need to study whether lean principles can have a similar effect in the software industry.

Research Question: Can lean principles be applied to the Software Industry for more efficient software delivery?

In order to define 'more efficient' in the research question, the researcher has identified three parameters on the basis of the writings of Elzamly and Hussin (2014) and Kerzner (2013):

1. Faster: Software delivery can be called efficient if the software is shipped within the given deadlines. In order to study whether lean principles can make software delivery more efficient than traditional methods, the duration of a release can be compared for companies that use lean principles with companies that do not use lean principles.

2. Lower Costs: For efficient software delivery, costs of the project need to be low. In order to study whether using lean principles can reduce costs of a project, the costs of the project can be compared for companies that use lean principles with companies that do not use lean principles.

3. Better quality: Software quality covers many aspects like meeting customer requirements, fewer defects, less waste and general customer satisfaction. Delivering better quality software faster and with lower costs is the epitome of efficiency for which companies strive.

For the research being conducted, the researcher will only be studying the first two parameters, i.e. faster software delivery and lower costs because they are tangible and can be quantified into concrete numerical data. Software quality increases the scope of the study and the time constraints of the research require a narrow research scope. While conducting the research, the assumption made will be that the software meets the quality standards, because well-reputed companies will be participating in the study. Also, the questions asked will be based on software deliveries that have already occurred, thus implying that the software has been accepted by the customer and can therefore be classified as good quality software.

1.4 Research Objectives

The research objectives have been derived from the research question. The aim of the research is to derive an answer to the research question. The objectives have been specified below:

1. To find out whether applying lean principles make software delivery faster.
2. To find out whether applying lean principles to a software project makes software delivery more cost effective.

When the comparative words are used, i.e., 'faster' and 'more cost effective', the comparison is being made to companies that do not use lean principles.

1.5 Hypothesis

For this study, the hypothesis is an answer to the research question.

Research Question: Can lean principles be applied to the Software Industry for more efficient software delivery?

Hypothesis: Yes, the use of lean principles in the software industry will help to make software delivery more efficient.

The hypothesis states that the answer to the research question will be positive. If the answer to the research question is 'Yes', then it implies that the hypothesis has been proven true. However, if the answer to the research question is 'No', it will lead to two possible research questions:

1. Does the use of lean principles not have any effect on the efficiency of software delivery? or
2. Does the use of lean principles have a negative effect on the efficiency of software delivery?

If the answer to the original research question is 'No', then further research needs to be carried out to determine which of the negative research questions is true.

The researcher will conduct the study keeping the hypothesis in mind and try to observe trends and map connections between the use of lean principles and the efficiency of software delivery.

Chapter 2

Literature Review

Chapter 2
Literature Review

2.1 History of Lean

Goddard (2009) writes about the history of Management Theory. Management in its present form dates back to the late 19th century, after the Industrial Revolution, where large manufacturing industries and factories were in place but they were not well managed. This led to lower efficiency and therefore required management practices to be introduced in manufacturing processes. F.W. Taylor focused on 'scientific management', which used scientific processes to improve manufacturing processes. Henri Fayol wrote about management in the form of tasks or activities that could be carried out and did not agree with the concept of Scientific Management; Fayol kept the scientific and managerial aspects separate. He created the 14 Principles of Management which is a guideline used by Managers to manage companies effectively. Goddard (2009) writes that Peter Drucker based his work on the work of both F.W. Taylor and Henry Fayol. Berson (2014) writes about Peter Drucker with respect to improving a company's effectiveness. According to Berson, Drucker propagated improving efficiency in industries by allocating human resources on the basis of skill and forte, and using managerial practices to improve workers' productivity. With respect to effectiveness of a company, it can be said that Druckers work was more based on Henry Fayol's teachings than on F.W. Taylors'.

The classical management practices were used in the manufacturing industry in most parts of the world. However, after World War II, Japan sustained heavy losses. Japanese manufacturers were in great need of efficiency; they required a management principle that allowed factories to effectively put their resources to use. They needed to produce high quality products in order to meet customer demands, but they also needed to keep their costs low and profitability high, so that they could recover from the losses of the War (Dominici & Palumbo, 2013). For this reason, innovative systems were introduced and implemented in Japan. The Toyota Way, or lean principles, emerged officially in the 1980s when Toyota began producing reliable cars quickly and at high profit margins. The customers were satisfied because the cars were of a high standard and at a reasonable price. However, this was a radical development for the manufacturing industry, as Toyota had discovered a way to produce good cars with very few defects and recalls, had loyal and happy customers and still managed to make high profit margins (Liker, 2004).

The manufacturing process applied by Toyota came to be known as the Toyota Production System (TPS) and the principles used by them became famous as Lean principles. Since the 1980s, manufacturing companies all over the world have adopted these principles. Lean principles have helped to improve manufacturing processes and improve the efficiency of manufacturing companies (Liker, 2004). In the twenty first century, Lean has spread across industries and the principles are being implemented to improve business processes in general. The software

industry, too, has begun implementing lean principles in order to improve software delivery (Bell and Orzen, 2010).

Today, Lean principles are applied through companies, across departments. One of the major companies to have adopted lean and been successful is Amazon. Amazon's customer orientation, focus on waste elimination and continuous learning are in keeping with Toyota's lean principles (Onetto, 2014). Duncan and Ritter (2014) discuss the adoption of lean principles in many other industries, like the service industry, marketing and finance sectors. Smith (2013) writes about how lean principles are successfully moving from the manufacturing industry to supporting services like accounting. Section 2.2 covers the philosophies and processes that form the lean principles in greater detail.

2.2 What is Lean?

According to Liker (2004), Lean is not merely a process that is implemented, but a management philosophy that has to be followed at every level of a company in order to provide positive results. Lean principles can be applied by following the Toyota Production System (TPS) that is based on the 4 Ps: Philosophy, Process, People and Partners and Problem solving. The 4 Ps are detailed in the following pyramid.

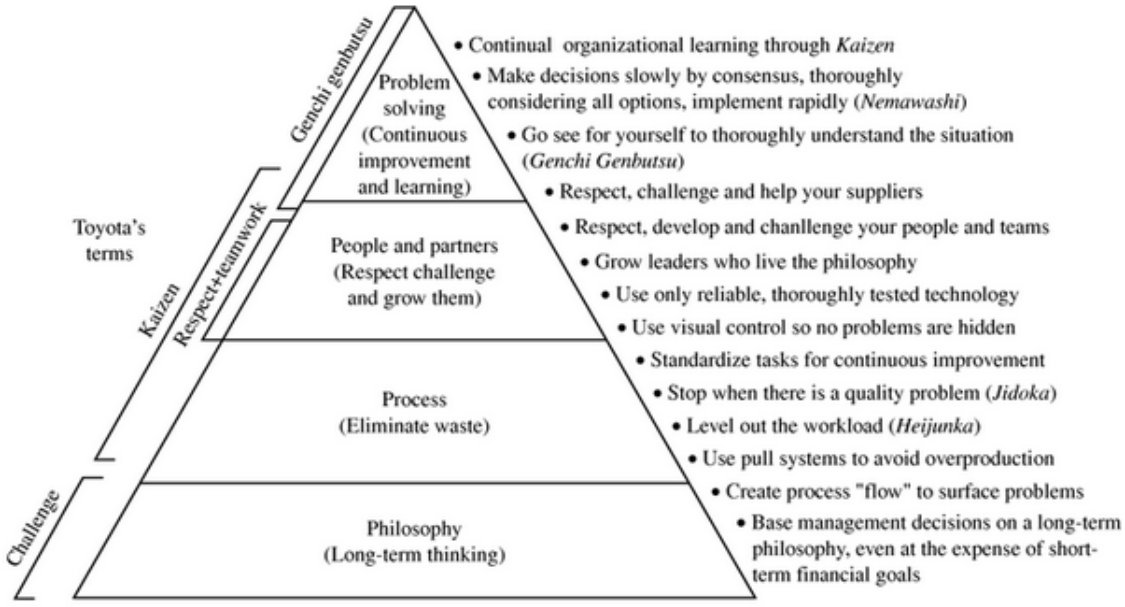

Figure 2.1: The 4 Ps and the 14 Principles of Toyota (Source: Liker, 2004)

According to Liker, the first P that needs to be tackled is the Philosophy. The lean transformation occurs once the philosophy of the company is altered or changed and the new focus or mission is to focus on the clients as opposed to profit making. The customer focus is the basis of the Toyota Way. Maurer (2013) reviews Womack's writing (cited by Maurier, 2013) about lean management theories,

bringing to light the difference between modern management theories and lean principles. Modern management lays stress on levels of hierarchy where the concentration of employees is on performing to the tune of the immediate superior in the hierarchy as opposed to lean principles, where employees concentrate on performing to the tune of the customers. When the outlook of a company changes to customer focus, it can be said that lean principles are being applied. Liebisch and Gruhs (2012) summarize the lean philosophy, saying that lean propagates that 'anything superfluous to the end result of customer satisfaction should be eliminated or reduced, thereby simplifying manufacturing, sales and distribution processes.'

As per Liker, once the philosophy has been transformed, the next P that requires a transformation is the Process. The focus of improving the process is to remove all wasteful processes and procedures and not spend time or money on any process that does not add value to the customer. This helps to provide the best products to the customers at the best prices. Partida (2014) suggests that when companies begin to improve their processes as part of the lean transformation, they should begin with improving internal process first on a trial basis. Once the employees are comfortable with the new philosophies and processes, the external processes like supply chain should be tackled. This will help to learn from problems faced in the internal processes and will help the company to implement the necessary changes in the external facing processes. Since the lean philosophy focuses on customers, customer-facing processes should be smooth and well managed at all times.

Implementing lean principles requires people who will maintain the lean philosophy and implement the lean principles. That is the third P as set forth by Liker: People and Partners. Employees on a company have to be trained in using lean principles, and they have to be able to carry forward the lean outlook long after the lean transformation has occurred. However, a company is not a separate and self-sufficient entity, all companies have suppliers and external stakeholders, all of whom are important in the business processes. It is important to bring everyone on board with the lean transformation and to ensure that the lean outlook is in keeping with the views of all the partners and suppliers (Liker, 2004). Swartling and Poksinska (2013) write about the human resource management aspect of lean principles. Lean principles are implemented by people and it is the people of the company who have to carry forward the lean legacy. Therefore, it is important to hire people whose skills match the job requirements, reduce the turnover of employees and keep employees motivated.

Liker's fourth P is Problem Solving, which implies that people must be able to solve problems by gaining a clear understanding of what is wrong and then make calculated decisions based on all alternatives available. The company must always learn from mistakes and keep improving itself in order to be successful (Liker, 2004). Atkinson and Nicholls (2013) discuss the risk of making a mistake in a business process, laying stress on the fact that one mistake could have serious implications on the entire workflow. Lean principles imply a continuous learning

process that uses all previous experience in order to avert serious dangers in the future. The culture of a lean organization focuses on understanding the system, understanding the mistakes and making well-informed decisions in order to minimize the risk of failure.

According to Liker, applying these 4 Ps (Philosophy, Processes, People and Partners and Problem Solving) will help a company to achieve the lean philosophy and implement lean processes. However, the question arises of whether lean principles that were created and used by one single company will be applicable to the industry in general. Monden (1998, cited by Dominici & Palumbo, 2013) expresses uncertainty about whether the Toyota Production System can be applicable to socio economic contexts outside Japan. Every culture and working environment is different and the professional set up in Japan may not be similar to other countries and continents. Dominici & Palumbo (2013) expel these doubts. They write that the Japanese were forced to reduce the use of their resources because of the after effects of World War II, and therefore developed an innovative system to remain efficient. The need for innovative systems due to a shortage of resources is not a scenario that is uncommon in the rest of the world. Especially in the current global context, working with limited resources is an integral part of every business organization. Every organization looks towards putting their resources to the best use in order to satisfy customers while also keeping their own profitability high. Every company strives to achieve efficiency in their business processes and operations. For this reason, the Lean principles will be applicable outside Japan as well.

Čiarnienė & Vienažindienė (2013, 'Lean Manufacturing Implementation: The Main Challenges And Barriers) discuss the processes involved in lean principles in detail. They write about a five-step process in which the first step is to define the customer's requirements. The next is to determine which are the most valuable processes and identify the processes that do not add value. The third step is to ensure that once the production process is in place, it should flow smoothly and continuously and the fourth step is to try to connect processes in case the flow is not smooth. The final step is to strive for a perfect process so that no time or resources are wasted dealing with faulty processes and wastages.

Čiarnienė & Vienažindienė (2013, 'Lean Manufacturing Implementation And Progress Measurement) also summarize the activities that are associated with implementing lean processes as 'five dimensions: elimination of waste, continuous improvement, continuous flow and pull-driven systems, multifunctional teams and information systems'. Compared to Liker's 4 Ps, Čiarnienė & Vienažindienė take a more narrow approach to the implementation of lean principles. Liker's approach is more holistic, covering the outlook of the entire organization, the way the organization should interact with stakeholders and even outlines how the organization should maintain the principles in the long term. The approach taken by Čiarnienė & Vienažindienė only discuss the physical implementation of lean principles at a given point of time, and do not mention ways to sustain the transformation over a period of time.

However, the writings of Čiarnienė & Vienažindienė do show evidence that they have followed the basic principles as laid down by Liker. They make a mention of the philosophy, people and problem solving aspects of lean principles, but they concentrate more on the process improvements that Liker recommended. Maurer (2013) emphasizes the importance of the application of the lean philosophy across the organization rather than merely changing processes. He writes that unless everybody at all levels of the organization adopts the lean philosophy, the process improvements will not show positive results. Bhasin (2013) and Atkinson & Nicholls (2013) write that for the lean philosophy to be implemented, it is necessary for the organization to bring about changes in the corporate culture. Both Bhasin and Atkinson & Nicholls agree that since people are an important part of a lean enterprise, the corporate culture is vital to the success of the organization. The skill of each worker is a valuable resource to a lean enterprise and the organization depends on the productivity of its employees. In order to be productive, employees need to be motivated and this leads to the discussion on the importance of corporate culture on lean enterprises. The culture of the company should encourage continuous learning and should not hinder innovation and growth.

All of the following: Maurer, Bhasin and Atkinson & Nicholls are of the same belief that lean principles are not only about the processes, but also about the philosophy, people and problem solving skills in the organization. All three are in agreement that implementing lean principles is not a narrow field and therefore they provide broader explanations of lean principles than Čiarnienė & Vienažindienė.

2.3 Lean and Six Sigma

While studying about manufacturing processes and their improvements, there are a many references to Six Sigma. Six Sigma is a combination of tools and procedures that are implemented in manufacturing processes in order to reduce the eight types of wastes that occur: ‘defects, overproduction, waiting, under utilized talent, transportation, inventory, motion, over processing’ (Bolte, 2014, p. 51). Bolte writes that Six Sigma goes hand in hand with lean principles as reduction of these eight types of waste can make the manufacturing process lean.

There are opposing views as to whether Lean and Six Sigma follow the same philosophies. Sunder (2013) writes about an integrated model that combines Lean and Six Sigma in order to have one common process that eliminates waste. Sunder compares the two methodologies and writes that they are complementary to one another. If combined, Lean Six Sigma can have a list of advantages to the company, which each methodology on its own cannot achieve. Smith (2013) takes an opposing view to Sunder, opining that Lean and Six Sigma are not complementary, but in fact Six Sigma is a sub set to Lean. Lean is the broad management principle that is used throughout an organization whereas Six Sigma is a tool that can be used in lean processes. It is not mandatory that Six Sigma is a part of lean, but since Lean Principles and Six Sigma focus on waste management, Six Sigma is a methodology

that can by used by Lean companies. Bell and Orzen (2010) agree that Six Sigma and Lean can work together in an organization, even though Six Sigma is not officially a part of lean.

Teschler (2014) who writes about the benefits of using lean principles argues against using Six Sigma. According to him, companies that use Six Sigma limit their ability to innovate, while lean principles propagate innovation. According to him, Six Sigma is a rigid methodology and does not resonate with what Lean principles Therefore, he believes that Six Sigma and Lean should be independent entities and companies that follow lean principles should not follow Six Sigma.

However, according to Liker (2004), waste minimization does form an important part of lean principles. Six Sigma is a waste reduction methodology that enables companies to reduce their wastage and therefore it can be concluded that Six Sigma can play a role in a lean approach to management, though it is not a compulsory tool that forms a part of Lean principles.

2.4 Effects of Lean Principles on the Manufacturing Industry

Camacho-Miñano et al (2013) record the results of a study on articles evaluating the financial performance of organisations using lean in the manufacturing industry. From the research, Camacho-Miñano et al write that there is no conclusive evidence linking financial improvements to the use of lean principles. However, Partida (2014) shows evidence of the use of lean principles reducing costs of business processes. The data below shows the results generated by Partida that show a reduction in costs of the procurement process and the materials order process.

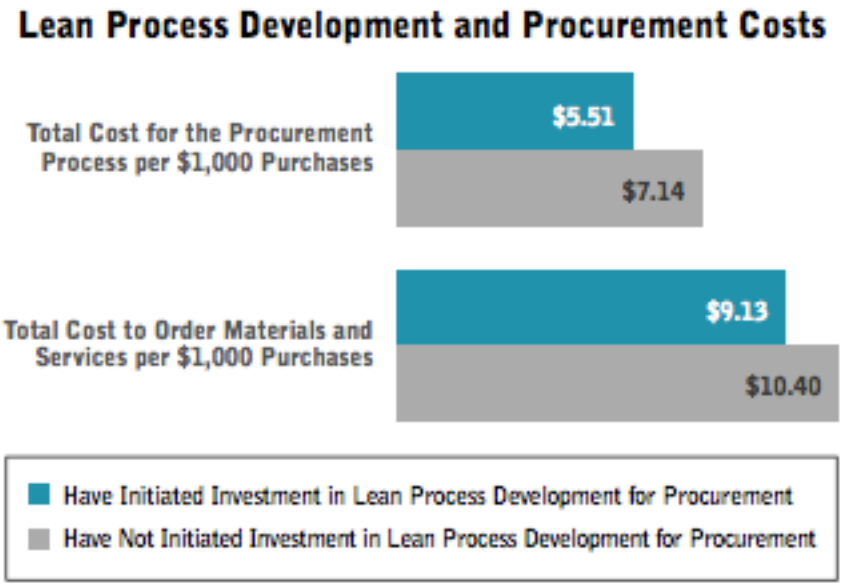

Figure 2.2: Cost reduction using Lean Principles (Source: Partida, 2014)

Filho and Uzsoy (2013) do not mention the cost benefits of using lean principles in production, but they write about the impact lean production has had on the set up time and repair time of the machinery during the manufacturing process. Their study shows that the duration of the setting up of machinery as well as the repair of machinery was reduced considerably after the implementation of lean principles. Thus, wastage in terms of time is reduced and the production process is made faster, providing general benefits to the manufacturing industry.

Sobral et al (2013) bring a new light to the topic of lean manufacturing. They study the connection between lean principles and the environment in the automobile industry. The use of lean principles in the automobile industry eliminates physical evidence of waste. Waste in the automobile manufacturing industry consists of materials that are hazardous to the environment. In the attempt to eliminate waste in order to reduce cost and save time, there is a positive side effect, i.e., less damage to the environment. Today, businesses and the environment strive to work together to achieve sustainable development, and such a positive side effect is beneficial to a manufacturing company. In terms of business too, less damage to the environment improves the goodwill of a manufacturing company. In a world where there is high competition for customer satisfaction and cost benefits, goodwill plays an important role for companies.

Keyes (2013) illustrates the reasons why Lean manufacturing is required in the industry. According to him, the reason for reducing wastage and having fewer steps is to improve the 'operational efficiency', which results in higher profit margins and a higher standing among the competition. With the increase in competition and reduction in the capacity of companies to hire a large number of employees, companies that can use lean principles will have a competitive advantage over companies that do not use these principles. Though Keyes is referring to the manufacturing industry, the same need for lean holds true in the software industry. Since the software industry is more dependent on human resources than the manufacturing industry, it may be of vital importance for companies to use lean principles in order to maintain a competitive advantage in the current economic conditions.

2.5 Software Project Management

Schwalbe (2014) gives an introduction to how the software industry works and why Project Management is an important field while talking about software delivery. The industry functions around a company developing and maintaining software for its own needs or for a client. Every task of creating new software or updating existing software is undertaken in the form of a project. For this reason, Project Management is an essential skill required in this industry. The quality of management practices used can determine the success or failure of a software project.

Ebert (2014) writes about the difference between Software Product Development and Software Project Management. Unlike what Schwalbe writes about the industry being based on building custom software for a company or client's needs, there is also a product aspect to the software industry. Creating a software product is different from developing a custom software application. The goals, objective and deadlines are different and the focus is different. While software products are standardized and requirements are based on the market, custom software projects vary from one client to another and are based on each client's requirements. However, for the purpose of the development team, the software development processes remain the same.

Schwalbe (2014, p. 4) discusses how a software project differs from an operations or manufacturing project. In a manufacturing company, the processes form a part of the operations, i.e., processes that occur regularly in order to maintain the running of a business. In a software company, software delivery takes place in the form of projects, which have a definite beginning and an end. However, like operations in manufacturing companies have defined processes, software projects too have phases and processes. However, manufacturing processes are based on automation with less human interaction whereas software development processes are people-oriented and therefore have more variables. Schwalbe also discusses the nature and characteristics of software projects. Unlike other industries, the software industry has many variations within itself. Projects can be of various kinds and across various domains. Software delivery is more dependent on people than the manufacturing industry. Employees have different backgrounds and skills and each project requires a customized set of skills. Therefore, software project management needs to be dynamic and flexible.

2.5.1 What makes a software project successful?

Kerzner (2013, p. 7) writes about what makes a project in any industry successful. Traditionally, project success was assessed as 'the completion of an activity within the constraints of time, cost, and performance.' However, Kerzner writes that nowadays, projects are judged by a larger number of factors, for example, client acceptance, fewer deviations from the original scope, conformity of the project to the processes and culture to those of the company that is undertaking the project. A successful project will maintain good relations with the customers as well as the organization that is carrying out the project.

Elzamly and Hussin (2014) write about the software industry specifically. Similar to what Kerzner discussed as the traditional approach to defining project success, Elzamly and Hussin write that the success of a software project can be assessed on the basis of three factors: budget and cost, schedules and meeting deadlines and the quality of the software.

Ghazi et al. (2014) give a more detailed evaluation of whether a software project can be deemed as successful or not. For each project, there are three main stakeholders,

viz., the development team, the customers and the management. Each stakeholder has a different definition of success. For the development team, the quality of the software is of utmost importance. The team will deem a project as successful if the software is well written and does not have many defects and errors. For clients, the quality of the code is not of any consequence. The quality is judged by the speed of the applications, how usable the software is and whether it meets the business requirements. When the management judges a software project, the success is evaluated by looking at the level of customer satisfaction, and most importantly, costs incurred and profits made. This is a more detailed analysis of success of a project, but in general, both Ghazi and Elzamly & Hussin agree on the fact that a software project can be deemed as successful, the software should be of good quality, the delivery should meet the deadlines and the project should be profitable.

Preimesberger (2014) lists a few project management tools for improving IT projects. They are: automation, collaboration, visibility, coverage, flexibility, scalability, usability, mobility, and deployment.

1. Automation refers to replacing monotonous or recurring activities that are done manually with automated processes. This will help to put skilled professionals' time to better use and also make their work less boring.
2. Collaboration can be achieved by encouraging communication and will help to locate and remove defects early and before the matter escalates.
3. Visibility helps to give a team an idea of what they are working towards so that the decisions they make conform to the future plans of the team and project.
4. Coverage refers to IT project management tools being up to date with the new innovations in the IT sector so that the area of new innovations is considered.
5. Flexibility is a very important aspect of an IT team. The team needs to be dynamic while dealing with clients and be able to adapt to different environments and business demands.
6. Scalability is also very important to development teams. The project should have the ability to grow with the growing market and demands without causing problems to the current implementations.
7. Usability is necessary for any software application to be successful. Pages should load fast and the functionality should be easy to understand and use.
8. Mobility: IT Project management tools should be mobile and should not be location based. Managers and teams should be able to communicate from anywhere in the world.
9. Deployment refers to launching the software, either to the test teams or to the clients. The processes of deployment should not be difficult and time consuming.

Preimesberger's list provides a guideline to how improve a project. While Ghazi et al say usability is one of the factors that can define success for a software project, Preimesberger says it is a project management tool that can be used to improve the

project. However, since both writers lay stress on usability, we can conclude that usability is very important to a successful software project.

Ghazi et al write about six factors that companies can consider in order to ensuring the success of software projects. Unlike Preimesberger, these factors focus more on the people than the tools that can be used. The factors are 'clearly stated requirements; involved users; engaged, competent project manager; project planned and scheduled; engaged, skilled team members; team work and communication encouraged.' All of these factors are on a higher level than Preimesberger's tools and are more difficult to put into practice. Therefore, Ghazi et al do not provide as clear a guideline to improving software projects the way Preimesberger does.

From the six factors, it is visible that Ghazi et al believe that the success of a project depends on the management and not the technology that is used. This is in keeping with what Schwalbe (2014) wrote about IT Project Management being able to determine the success of a project.

In contrast to Ghazi et al and Schwalbe about Project Management being the most important determinant of success, Stoica et al (2013) and Armour (2013) both write about the technical side of software project management and how they determine the success of failure of projects. Stoica brings to light the importance of quality assurance in a software project. According to him, a software project's success is determined by the effectiveness of the quality assurance and software testing conducted by the team. Armour on the other hand talks about why software projects are hard to estimate and how this has an effect on project management. The complexity of technological problems faced is hard to measure before the task is undertaken. Thus, as both Armour and Stoica et al conclude that the technical aspect of software development has to be considered while managing an IT Project.

It can be concluded that there is no one particular definition for a successful project. A software project needs to be technically strong and well managed to provide desired results. There are many factors to be kept in mind while trying to successfully complete a project. However, having this knowledge is not enough, implementing the tools and processes are also important. The next section discusses why so many software projects fail.

2.5.2 Why do software projects fail?

Kerzner writes about constraints that determine whether a project will be successful or not. The 6 constraints are: time, cost, scope, quality, value, image/reputation and risk. Kerzner compares traditional and complex projects in terms of constraints. In order to be successful, a traditional project has to independently achieve success in all the areas of constraint. The project should be delivered on time, at the decided cost, while maintaining quality, value and image and the risk of failure should be low. However, for a complex project, each

constraint competes with another, and depending on the customer's need, one or a few constraints have to be maintained and cannot be ceded. For example, value and cost may compete with each other. The quality of a project may not be maintained if the cost is reduced. However, if the customer wants high quality and low cost, the project's success is determined by competing constraints. This is where the problem arises and a project fails in one or many of the areas that determine success.

Elzamly and Hussin (2014, p. 411) write seven reasons traditional reasons why software projects fail. They are 'poor project planning, insufficient communication, lack of change, financial and performance management, failure to align with constituents and stakeholders, ineffective involvement of executive management, lack of qualified team members in the areas of soft skills, ability to adapt and missing methodology and tools.' Each of these has been discussed and analyzed below:

1. *Poor project planning:* Gupta (2013) writes that project planning is often not successful because project plans can be badly structured, incomplete and rigid. They may not account for every aspect of the tasks or for contingencies. Ferrucci et al (2013) discuss the problems caused due to bad project planning. Poor project planning leads to the development team working overtime. The software industry is dependent on the skills of the development team, and the development team may not be as productive while working overtime and will also be dissatisfied and stressed. The quality of the software suffers while the project cost rises, thus causing the project to not achieve success.

2. *Insufficient communication:* Cheung et al (2013) lay stress on the importance of communication in any project across industries. They write that communication builds trust within teams and between teams and stakeholders. Holzmann and Panizel (2013) write about how communication is vital to the success of software projects. While traditional models like waterfall rely on more formal communication like documentation, projects that use agile rely on more informal methods of communication within teams and with clients. However, in both cases, communication is a key factor in whether the project succeeds or fails.

3. *Lack of change, financial and performance management:* Heerkens (2014) writes about the cost of change for a project. With scope changes, the cost of the projects rises and causes the project to not be as successful in terms of budget. This is where both change and financial management come into question. Elzamly & Hussin, Kerzner and Ghazi et al measure one aspect of success by meeting budgets, a project with a changing scope or poor financial management can lead to higher costs and therefore, failure. Robbes et al write that although there is a cost associated with performance management, if the project manager does not know how tasks are being performed at each

stage, corrective measures cannot be taken in case of problems, and if such problems are not detected early, it may lead to project failure.

4. *Failure to align with constituents and stakeholders:* A study conducted by Holzmann and Panizel revealed that agile projects with better communication among the teams and with clients and the management showed more success than projects where communication was insufficient. Thus the converse was also proved true, i.e., projects that failed to communicate and align with the stakeholders were not very successful.

5. *Ineffective involvement of executive management:* Par (2014) writes about the importance of leadership in project management. If the executive management is not in tune with the regular working of the project and tries to manage the project without the right background to the ongoing activities, it could lead to dissatisfied team members or ineffective management. This will hinder the success of the project.

6. *Lack of qualified team members in the areas of soft skills and ability to adapt:* Fernando Capretz (2014) brings to light how software development is essentially an activity carried out by and dependent on the skill of people. Technical skills may form the basis of a good software application, but for the project to be successful, human elements like soft skills and the ability to adapt to a dynamic environment are essential. A project may fail if the team members are not strong in these areas. Kerzner (2013) argues that the skill of the project manager reflects in the way the team communicates and adapts to change. However, from both arguments, it is certain that the human element is vital to the success of the project.

7. *Missing methodology and tools:* In Section 2.5.1, Preimesberger (2014) enlisted a few tools that help to improve software projects. Elzamly and Hussin write that if there are missing tools and methodologies in software project management, it can lead to project failure.

After stating the seven traditional reasons for software project failure, Elzamly and Hussin go on to identify two more factors that lead to project failures: software sizing problems and ineffective risk mitigation.

While talking about software sizing problems, Elzamly and Hussin discuss the complexity of problems and therefore the added cost while sizing projects. Armour (2013) discusses software estimates in relation to development teams. Sizing a project and making estimates are often inaccurate; that is one of the reasons why software projects take longer than expect and why the projects are not 100% complete. While estimating tasks, the development teams may unintentionally magnify the progress made while not providing accurate estimates of the tasks that are still remaining. However, complexity of software projects is a real issue and it is genuinely difficult for the development team to estimate time taken for

contingencies like scope changes or defects and time taken to fix the defect and test the application.

This leads to the question of contingencies and risk. Elzamly and Hussin write in great detail about ineffective management of risk being a reason for project failure. Risk can occur in every stage of the project, but the problems that are detected later in the project are more difficult to mitigate and can have grave consequences. The risk in software projects is high because of the dependence on the human element and the number of unknowns when it comes to technical complexity. In contrast, Hijazi et al (2014 p. 52) write that risk is not uncertain and does not 'come out of nowhere'. A risk factor is directly or indirectly related to another factor and that risk factor in turn causes another factor to be at risk. Therefore, if the project manager understands the root cause of problems as well as potential problems, risk can be managed.

In conclusion, it can be summarized that a software project is successful if it meets clients needs, not just about the software system requirements but also regarding other important factors like cost and time. However, implementing software project management processes are not always successful and therefore sometimes lead to failure. While Software Project Management has a broader scope, the next section, Section 2.5.3 deals with the implementation of management methods in the form of software development methodologies.

2.5.3 Software Development Methodologies

Schwalbe (2014, p. 59-63) discusses the processes involved in software projects. There are 2 kinds of models that are used while delivering a project: the waterfall model and the spiral model. The waterfall model is a straight-line method where one process follows the other. The stages are: analysis, design, construction, testing and support. The assumption while using a waterfall model is that requirements remain the same over the development process. Jackson (2012) writes that the traditional waterfall model should not be written off in light of new methodologies as it is a model that has had proven results in the past and is still applicable to software development.

The spiral model takes into account dynamic client requirements and the processes are repeated at regular intervals. One of the recent spiral models is the Agile model which maintains flexibility while developing software (Schwalbe, 2014). Packeer Mohamed (2014) conducting a study about which are the most popular Agile methodologies used. Out of the following: Extreme Programming, Scrum, Adaptive Software Development, Crystal Methodologies and Feature Driven Development, Scrum and Extreme Programming (XP) are the most widely used methods. While XP focuses on the development processes, Scrum pays more attention to the management aspect of software development. Together, the two methodologies complement each other as they focus on the two main aspects of software delivery. Babb et al (2014) write about the implementation and learning from Agile. It

propagates iterative development, where requirements for the entire system are not fixed in the initial phases, but are flexible and can be changed on the clients' requests. The features with the most value to the customer should be deployed first and supporting features should be deployed later. The workflow is spiral, so it is a repetitive model.

Jackson (2012) considers the characteristics of both Waterfall and Agile models. Waterfall implies 'heavy upfront planning' and a 'sequential order of work' while Agile implies 'less upfront detailed planning' and 'very short work iterations, or sprints'. When a company uses waterfall, each stage of the Software Development Life Cycle (SDLC) is conducted in great detail and each step is specified, completed and then the team goes forward to the next stage. In Agile development, each stage repeats itself at regular intervals and each sprint or iteration is completed before going forward to the next. Introducing changes to the Waterfall model is more difficult as it is a more formal process whereas Agile development allows changes to be made more easily. Stoica et al. (2013) compare the agile model and the traditional, waterfall model. Stoica suggests the use of Agile models for small and medium sized projects and traditional models for larger projects. Stoica also touches upon the question of requirement changes. Agile models do not have fixed requirements at the beginning of the project and therefore provide more flexibility during the development process. Incorporating changes is easier through the development process. The traditional model is more formal and has well defined requirements at the beginning of the project. In the later stages of the process, it is more difficult to incorporate changes. Both Agile and traditional models have their advantages and disadvantages and Stoica et al conclude that regardless of which model is selected, the processes involved are complex and there are many chances of error. This is why there is a need for thorough testing in every development cycle.

2.6 Application of Lean Principles to the Software Industry

From section 2.5, it is visible that no matter which methodology of Software Development is used, there is room for improvement in software development. Souza et al (2013) discusses how the Pareto Principle can be applied to the software development, with only 20% of the totally system providing 80% of the value of the software product. Keeping this in mind, Bell and Orzen (2010, p. 174) write that 'Lean software development begins with a simple premise: identify the 20 percent of the code that provides 80 percent of the value, and deliver that just-in-time'.

Bell and Orzen (2010) write about implementing lean principles to software delivery, in terms of both software development and project management. They propose a model in which requirements are discussed at regular intervals so as to understand the customer's requirements. To keep in mind that customer's requirements change regularly, a spiral and iterative model should be used. Testing should be conducted regularly, not just towards the end of the SDLC. In order to prevent waste, Quality Assurance throughout the SDLC is a very important aspect of Lean principles.

Sheng et al (2014) discusses quality assurance in terms of cost of a defect. When a defect is discovered early, it can be fixed without having a huge impact on other parts of the system. Therefore, fixing it is quick and easy. It also might give an indication towards design flaws in the system, which in turn can be fixed. However, if the same defect is located towards the end of the SDLC, it will be time consuming and will have an impact on other areas of the system. Development and testing time will increase considerable. Thus, it is much cheaper to locate a defect early in the SDLC than late. This is the reason why Bell and Orzen (2010) propose undertaking software testing early in the SDLC.

Anderson (2012) discusses the principles that companies should follow while applying lean principles. Anderson has been quoted below:

"The Lean Software & Systems community seems to agree on a few principles that underpin Lean Software Development processes.

1. Follow a Systems Thinking & Design Approach
2. Emergent Outcomes can be Influenced by Architecting the Context of a Complex Adaptive System
3. Respect People (as part of the system)
4. Use the Scientific Method (to drive improvements)
5. Encourage Leadership
6. Generate Visibility (into work, workflow, and system operation)
7. Reduce Flow Time
8. Reduce Waste to Improve Efficiency"

Looking at the principles, it is clear that the principles have been applied from the lean principles originally laid down by Toyota (Liker, 2004). However, although there is one principle about waste, in general these principles lay more stress on the continuous flow of processes and the training of people rather than the waste elimination. According to Anderson (2012), the focus on the fact that process flow improvement is what constitutes Lean software delivery. In contrast, Bell and Orzen (2010) focus more towards reducing waste in a software process. Their main consideration is to eliminate waste from the SDLC. They identify waste in the form of features that are not used by clients and by requirements that are changed after implementation. They write about reducing such waste in order to make the software development processes lean.

Thus, there are differences in opinion on how to implement lean principles in the software industry.

2.7 Lean and Agile

Anderson (2012) writes that Alan Shalloway (2009, cited by Anderson 2012) compares lean principles and agile principles and categorises them together.

However, Anderson believes that Lean is a philosophy that does not form a part of the Agile methodology. Lean is an independent philosophy, which when applied to the field of IT has to be understood and modified in order to suit the requirements of the industry. Applying lean principles to IT management involved accepting the difference between the manufacturing industry and the software industry and paying regard to the fact that software delivery is dependent on people's skills, their understanding of complex situations and there will always be a certain degree of uncertainty.

Bell and Orzen (2011) write that while lean is a principle with a wide scope that applies throughout the IT sector, in cloud computing, Information Technology Infrastructure Library (ITIL), Project Management and Software Development, Agile is a narrow methodology that applies only to Software Development. However, Agile is a subset of Lean; it is the manifestation of lean principles in Software Development. While lean principles are the management philosophy that needs to be implemented throughout the organization, Agile is the way to implement lean principles in software development.

Thus, it is not conclusive whether Agile and Lean go hand in hand or whether they are independent and mutually exclusive methodologies.

2.8 Conclusions and Gap in the Literature

From the literature, it is visible that Lean principles have helped the manufacturing industry to become efficient, i.e., produce high quality products with minimum waste. The minimization of waste has led to a reduction in cost, faster production and therefore higher customer satisfaction. Lean principles are now being applied to the software industry in order to improve the efficiency of software delivery. When applied to the software industry, lean principles focus on eliminating waste by gathering domain knowledge, defining requirements and starting Quality Assurance procedures early in the development phase.

However, applying Lean principles to IT Management is a relatively new field of study and there has not been a vast range of academic literature written about it. As seen in Section 2.7, most of the literature discusses how to modify the principles that originated in the manufacturing industry and apply them to the software industry. There is no discussion about the results of applying lean principles to software delivery. This can be referred to as the Gap in the Literature. For this reason, the research being carried out will add to the literature in this field. The research aims to identify the effects of using lean principles to software delivery. When a management principle is applied from one sector to another, will the results and benefits be the same? The research question has been framed as: *Can lean principles be applied to the Software Industry for more efficient software delivery?* As opposed to the Literature, which deals with how to apply lean principles to software delivery, the research deals with whether using lean principles will have a positive effect on the software industry.

Chapter 3

Methodology

CHAPTER 3
Methodology

3.1 Methodology

The methodology selected for this research is based on the 'Research Onion' outlined by Saunders et al (2012, p. 128).

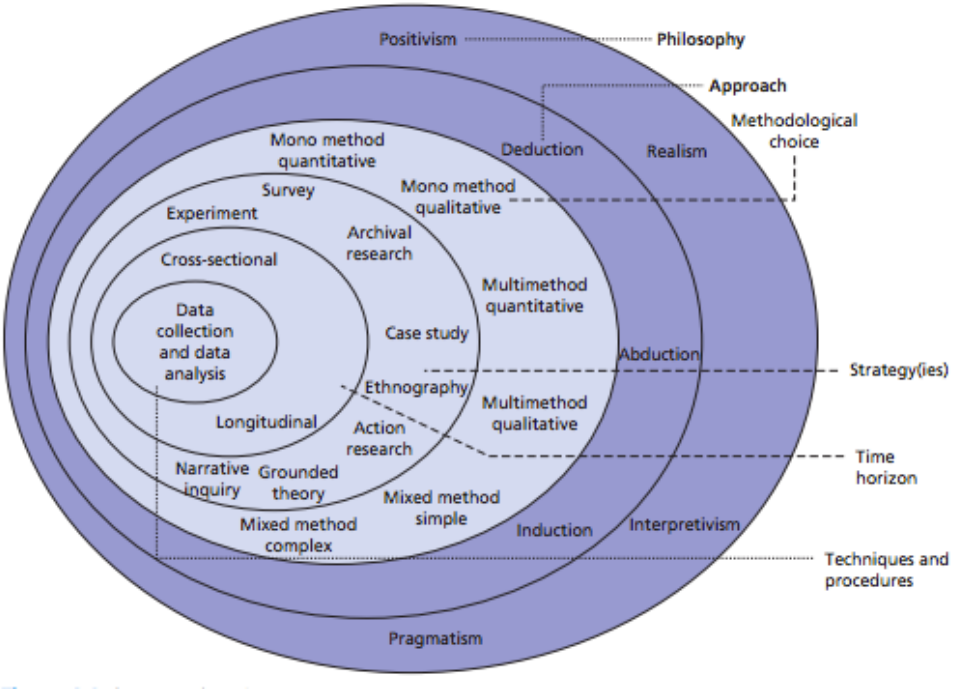

Figure 3.1: The Research Onion (Source: Saunders et al, 2012)

According to Saunders et al, research is carried out in steps, or layers that go from a wider, more philosophical outlook towards the research question to a more practical approach to solving the problem at hand. The philosophy and approach are the outer layers and are required to be decided first while probing into the problem in order to uncover the more practical aspects, i.e., the inner layers of the research. The way of thinking that is applied to the problem is the first step to solving a problem. Each philosophy, approach and methodology has its use and importance, and there is no set of methods that can be deemed better than another. The only concern is the relevancy of the research methods selected to the problem and question at hand. The research onion gives the various paths and combinations of research methods that are generally used for business research, but they are not exhaustive or rigid, but actually a mere guideline to how the research can be carried out. As per the guidelines given by the onion, the research philosophy can determine the approach, which in turn can determine the strategies and choices that are used.

The combination of research methods selected for this research at hand has been outlined below, along with justifications as to the choices.

3.2 Research Philosophy: Positivism

As per Saunders et al (2012, Chapter 4), there are three outlooks that determine the research philosophies: ontology, epistemology and axiology. Ontology deals with the matter of questioning reality and debating between the philosophies of subjectivism and objectivism. The researcher concludes that such a debate is invalid in the case of studying the data provided by software industry, as the data is concrete and the question of reality is clear and cannot be debated. Therefore these philosophies will not be used for the research being carried out. Axiology is an outlook that deals with 'judgements about value' (Saunders et al, 2012, p. 137), i.e., the researcher's sense of ethics while making decisions of selecting a research method. Since the study of efficient software delivery requires a more scientific approach than a value-based approach, this outlook, too, has not been selected for the purpose of this research.

Epistemology deals with 'acceptable knowledge' (Saunders et al, 2012, p. 132), what can be classified as a scientific approach, and whether a scientific approach can be applied to business and sociological environments or not. Positivism and realism are similar and apply a scientific approach to social environments, while interpretivism makes arguments against this and takes into account the feelings of the researcher while looking at a social problem. The study about lean principles in the software industry is of a scientific nature and attempts not to include a personal bias; therefore interpretivism has not been selected as a possible philosophy.

While undertaking this research, the three philosophies that were considered to be relevant were: positivism, realism and pragmatism. Positivism and realism both follow epistemological outlooks while pragmatism transcends epistemology, ontology and axiology and adopts any outlook and methodology suitable to the research question. The philosophy of pragmatism concentrates on selecting any outlook, or a combination of outlooks as long as the research problem has been kept in mind and the outlook is relevant and result oriented. Pragmatism was a choice for this study because it focuses on results and consequences. However, on further consideration, it was concluded that a scientific outlook is required to tackle the problem of inefficient software delivery. Realism and positivism are both scientific and deal with looking at causes and effects. Therefore, the choice of philosophies was narrowed down to two similar philosophies: positivism and realism. Both of them involve the researcher working in a scientific way to solve the research problem. Realism has two stances: direct realism and critical realism. Direct realism speaks about the reality of what is observed, whereas critical reality compares what is observed to what may be real (Saunders et al, 2012, Chapter 4). For the sake of this study, the researcher feels that there is only outlook, not two like Realism proposes, and that one outlook is closer to Positivism than Realism. Positivism deals with collecting data from 'observable reality' and applying the data collected to the general population, the way scientists conduct experiments (Gill and Johnson, 2010

cited by Saunders et al, 2012). The research at hand studies concrete data gathered about on-going projects in the software industry, it can be concluded that study deals with observable reality. For these reasons, **positivism** is the best choice of philosophy for the research being conducted.

3.3 Research Approach: Deductive

The possible choices of research approach are inductive and deductive approaches. According to Saunders et al (2012, Chapter 4), the deductive approach tests an existing theory whereas the inductive method builds a new theory. The study being conducted tests a hypothesis on the basis of existing theory.

The hypothesis is: Lean principles can make software more efficient.

The hypothesis needs to be tested scientifically and quantitatively. The variables in this hypothesis are the time and cost of software delivery. They will be studied by gathering data that link lean principles to efficient (fast and cost effective) software delivery. There will be 2 aspects studied: lean principles being linked to faster software delivery and lean principles being linked to lower costs of software delivery. If both aspects are proven to be true, the hypothesis will be deemed true.

Thus, from the above, it can be concluded that the approach is scientific and deductive. This is in keeping with the research philosophy of positivism. The research onion maps the philosophy of positivism to a deductive research approach (Saunders et al, 2012, Chapter 4). For these reasons, a **deductive** research approach has been selected for this study.

3.4 Data Collection: Quantitative

The data collection method being used is **quantitative.** The research question has been quantified in order to be able to study the data and provide numerical values.

Research Question: Can lean principles be applied to the IT sector for more efficient software delivery?

Efficiency has been defined using two parameters: speed and cost. Each of these parameters can be quantified as follows (Elzamly and Hussin, 2014):

Speed: The speed of software delivery can be measured by looking at time taken to deliver the software, i.e., length of a release cycle.

Cost: The cost of a software project is calculated in man-hours. Therefore, to calculate the cost of a project, many factors have to be kept in mind:

1. Size of the team: The size of the team is one of the factors that determine cost. The cost of hiring software developers, quality assurance analysts and

business analysts is high. The higher the number of people in the team, the higher the costs.

2. Time taken to deliver: If a team takes less time to complete a project, the project costs rise. However, team size and time taken are competing factors. A larger team may take less time to complete a task than a smaller team. But in order to achieve efficiency, the team size should be optimal to allow faster deliveries without the added cost of managing a large team. This aspect will be analysed on the basis of the data collected.
3. Amount of waste: A large contributor to costs in a development team is waste. The amount of waste is not always tangible and therefore is not easy to calculate. However, there are some aspects that can be considered while trying to quantify waste:

a) Late detection of defects: The cost of a defect is lowest at the beginning of the software development life cycle, and keeps escalating with each stage of the life cycle. A defect discovered in the later stages of the life cycle is very expensive to fix (Sheng et al, 2014).
b) Features in production that are not used by any users: If there is a feature in production that is not used by any user, it can be classified as a waste. The time and effort that was spent on developing and testing that feature could have been used more productively (Bell and Orzen, 2010).
c) Requirement changes after a feature has been developed: After a feature has been developed, if requirements are changed, it involves redoing work. Here too, the time and effort that was spent on developing and testing the feature the first time could have been used more productively (Bell and Orzen, 2010).

The data collection will be carried out using two strategies: a case study and surveys. All of the case study and survey questions have been attached in Appendix 1.

3.4.1 Case Study

While designing the case study, the research objectives have been kept in mind.

Research Objectives:

3. *To find out whether applying lean principles make software delivery faster.*
4. *To find out whether applying lean principles to a software project reduces the cost.*

The case study is of a company that has adopted lean principles over the last 5 years. The data being collected studies the changes that lean principles have brought about in the company since the implementation of lean. Since it is the study of the same company, the basis of comparison is credible and the cost and release cycle data will be directly relatable to the new management principle applied. The study is being conducted for two specific points in time: immediately before the

implementation of lean principles and the present day, after the implementation of lean.

Companies do not reveal data about costs; therefore the questions have been designed to compare costs without asking for sensitive data. The questions and their objectives have been discussed below.

1. What effect has the implementation of lean principles had on the length of a release cycle?

The length of a release cycle increased by
0-10% /11-20%/ 21-30%/ 31-40% /50%+
The length of a release cycle decreased by
0-10% /11-20%/ 21-30%/ 31-40% /50%+

This question is direct and can be related to the speed of software delivery. The answer to this question will provide a clear relation between lean principles and the speed of software delivery. This will be able to show whether lean principles were effective in making software delivery faster.

2. What effect has the implementation of lean principles had on the average team size?

The average team size has increased by 0-10%, 11-20%, 21-30%, 31-40% 50%+
The average team size has decreased by 0-10%, 11-20%, 21-30%, 31-40% 50%+

This question will provide an insight into the cost aspect. If the average team size has decreased, it can be concluded that the costs are lower.

3. What effect has the implementation of lean principles had on the cost of software projects?

The cost of projects has increased by 0-10%, 11-20%, 21-30%, 31-40%, 50%+
The cost of projects has decreased by 0-10%, 11-20%, 21-30%, 31-40%, 50%+

This is a direct question related to the cost of software delivery. The answer to this question will provide a clear relation between lean principles and cost of software delivery. This question will show whether lean principles are effective in reducing costs of software projects.

On receiving the answer to all three questions, the researcher will be able to conclude that the hypothesis is true. However, this is only the study of one company and therefore in order to make the research more credible and concrete, further research will be conducted.

3.4.2 Survey

The survey has been designed in order to compare companies that use lean principles with companies that do not. Since the size of the companies vary, projects in similar domains and of similar sizes will be compared.

The first few questions of the survey will determine which company, domain and what team size is in question. After that, questions related to speed and costs will be asked. These questions have been discussed and justified below.

a) Does the team use lean principles?

This question will divide the survey into two distinct sections for the purpose of data analysis. All further questions will be analysed on the basis of whether the company uses lean principles or not.

b) What methodology does the team use?
   Agile, waterfall, other

As seen in Chapter 2 (Section 2.2), lean principles have a wider definition and apply to all levels of management, whereas the methodologies of agile and waterfall apply specifically to software development. This question will help draw a relation between the principle and the methodology.

c) Length of the release cycle

For agile, the question will be divided into two parts, i.e., the length of an iteration and the number of iterations. For waterfall, the question will be direct. The question will reveal the speed of software delivery, comparable for projects that use lean and projects that do not. For the hypothesis to be proven true, the projects that use lean principles should deliver software faster that the projects that do not.

d) When are the most defects detected?
   - While defining requirements and designing a feature
   - During development
   - In the early stages of testing
   - During system testing

This question is meant to determine how high the cost of fixing defects is. The earlier in the software development life cycle a defect is detected, the cheaper it is to fix. Similarly, as mentioned earlier, the later a defect is discovered, the more expensive it is to fix (Sheng et al, 2014). For the hypothesis to be proven, the projects that use lean principles should detect defects earlier that the projects that do not use lean principles.

e) How much of the total system in production is used regularly by every user?
   81%+
   61-80%
   41-60%
   Less than 40%

f) How much of the total system in production is never used by any user?
   Less than 20%
   21-40%
   41-60%
   More than 61% of the system

These two questions are aimed at determining the waste. According to the hypothesis, a project that follows lean principles should have lower costs than a project that does not. As seen in Chapter 2 (Section 2.2), costs can be reduced if waste is reduced. If there is a feature in production that is never used, it is a waste of development time and effort and therefore cost. For the hypothesis to be proven true, projects that use lean principles should have a larger number for question e and a smaller number for question f.

g) For a given functionality, how much of the requirements are changed after the feature has been developed?
   0-10%
   11-30%
   31-50%
   51+%

This question has also been designed to determine how much development effort is wasted. Once a feature has been developed and the requirements are changed, it is a waste of developer time effort and therefore cost (Bell and Orzen, 2010). Projects that are lean will not have such wastage as part of the process. Therefore, for the hypothesis to be proven true, projects that use lean principles should provide a smaller value for this question than a project that does not use lean principles.

Therefore, the survey uses existing theory about software development and adapts the questions to determine whether lean principles are effective in making software development faster and more cost effective.

If the questions are answered as mentioned above, the hypothesis will be proven true and it will be concluded that projects that use lean principles are more efficient than projects that do not. If the questions are not answered as mentioned above, the hypothesis will be proven wrong and it will be concluded that lean principles do not have a positive effect on the efficiency of a project.

3.5 Sampling

The case study does not require sampling as one company is being studied at two different points in time. The company has been selected on the basis of the fact that it uses Lean principles and that is has implemented the change recently. The Project Manager has been part of the change and therefore can provide accurate details of the results that using lean principles has brought about in the company. The company has agreed to share details about the change process and this will help the researcher to draw conclusions about the adoption of lean principles and its benefits.

The surveys are being directed towards Project Managers in the Software Development field. This is a small population and therefore a non-probability sampling process has been selected. The survey will be sent to Project Managers, some of whom work in teams that use lean principles and some of whom work in teams that do not. The sampling method being used is **Critical Case Sampling** (Saunders, 2012, Chapter 7). The researcher feels that the cases being studied are relevant to the study being conducted. Each of the participants is very important to the study and the point made by each of them is relevant. The survey is aimed at a very specific population and therefore the sample size is small. There is a minimum requirement of 30 people to participate, half of which should work in companies that follow lean principles whereas the other half should work in companies that do not follow lean principles.

3.6 Time horizon: Cross sectional

As per Saunders et al (2012, Chapter 5) research can be conducted using one of the two time horizons: longitudinal and cross sectional. The longitudinal time horizon studies data over a large period of time and tracks the data through various stages. The cross sectional time horizon studies a given point of time. As we can see from the research design, the time horizon for the research is **cross sectional**. The surveys are being conducted for on going projects as of April-August 2014. The industry is being studied at a given point of time. The case study looks at data for the same company at 2 different points in time, one before lean principles were implemented and one after. However, the changes have not been tracked over the 5 years that it took to implement the principles, so it cannot be called a longitudinal time horizon. Therefore, the case study too is cross sectional.

Chapter 4

Research Findings and Data Analysis

Chapter 4
Research Findings and Data Analysis

In Chapter 4, the data collection findings have been discussed and analysed. The research findings have been divided into two sections: Case Study and Survey. Each section has been discussed individually and for each question or set of questions, the findings are followed by an analysis in order to gain a clear idea on what every result signifies. There is also a discussion on how the two studies are relevant to one another in Section 4.4.

4.1 Case Study Findings and Analysis

The Project Manager in the software company that was contacted has requested to keep the company name confidential because of the use of cost related data. The company is a Multinational Company with operations and clients in various parts of the world. The case study questions were designed to provide a clear relationship between the use of lean principles and the measures of efficiency. Since the same company is being studied at two different points in time, the assumption is that all factors except the management principle have been kept constant. The results to the questions are given below, followed by a short explanation provided by the Project Manager. Since this is a Quantitative research, the analysis focuses on the quantitative aspect of the case study.

A) What effect has the implementation of lean principles had on the length of a release cycle?

The length of a release cycle increased by
0-10% /11-20%/ 21-30%/ 31-40% /50%+
The length of a release cycle decreased by
0-10% /11-20%/ 21-30%/ **31-40%** /50%+

"The release cycle has reduced considerably. We have always used the Agile methodology and our sprint length is usually 2 weeks. Before we implemented lean principles, our main releases took place every 12-13 sprints. We also had one or two supporting releases after the main feature had been rolled out. The supporting releases were usually to fix any critical issues that came about. Now that we use lean principles, we release every 8 sprints on an average. The number of supporting releases has also reduced. Sometimes we don't need them at all."

The result states that the release duration has decreased by about 33% since the implementation of lean principles. From this result, it is clear that the software is being launched faster using lean principles. The manager mentions that there are fewer supporting releases, which can be an indication of less wastage. Waste reduction leads to a reduction in both time and cost. The effect of waste reduction

on time is clearly visible as per the results above. The effect of waste reduction on cost will be discussed in Question C.

B) What effect has the implementation of lean principles had on the average team size?

The average team size has increased by 0-10%/ 11-20%, 21-30%/ 31-40% /50%+
The average team size has decreased by **0-10%/** 11-20%/ 21-30%/ 31-40%/ 50%+

"I can't say the average team size has changed much. Resources are assigned on the basis of necessity and budget of the project, so every project has a different team size."

The change in the team size is negligible so this result does not allow the researcher to map the use of lean principles to the size of a software development team. The survey conducted also covers the relationship between team sizes and the use of lean principles, so the data collected from the survey will help to determine whether the use of lean principles has an effect on the size of the team. Since the cost of the development team contributes directly to the cost of a software development project, it is important to know if lean principles help to reduce team sizes.

C) What effect has the implementation of lean principles had on the cost of software projects?

The cost of projects has increased by 0-10%/ 11-20%/ 21-30% / 31-40%/ 50%+
The cost of projects has decreased by 0-10%/ 11-20%/ **21-30%/** 31-40%/ 50%+

"Giving an exact value for cost is tough because there have been a lot of changes in this field over the last five years. There has been inflation, but the cost of infrastructure and networking has also reduced considerably. Still, bearing all of this in mind I would have to say that the cost has reduced."

The result shows a distinct reduction in the cost of software projects as a result of using lean principles. As mentioned in section A, there are fewer supporting releases, and this can be attributed to a reduction in waste. Since a large aspect of the cost of software projects is calculated in terms of man-hours (considering the time taken by the development team and the cost of the development team), the reduction in time also leads to a reduction in cost. The time of each release has reduced by 33%, thus having an impact on the cost of the project.

The Case Study correlates the use of lean principles to the speed and cost of software delivery. It shows that the use of lean principles can make software delivery faster and more cost effective. However, the case study deals with the lean

transformation of only one company. Looking at the results of only one company does not allow the researcher to generalise the correlation between lean principles and the speed and cost of software delivery. The case study provides insights into the benefits that can be achieved if lean principles are applied, but it does not provide conclusive evidence. If further research findings back the results of the case study, only then can the findings be generalised. That leads to the next section of the study: the survey.

4.2 Survey Findings

The Survey was conducted using an online tool eSurv, which allows the user to create a survey and view the results. Once the results were received, they were exported to Microsoft Excel. Microsoft Excel was the tool was used to analyse the behaviour of the data.

The survey got 38 responses from Software Project Managers. The survey participants consisted of managers belonging to two countries: India and the Republic of Ireland. Although this question was not asked explicitly in the survey, the IP address was included in the data exported and this allowed the researcher to identify the location from which the survey was filled. On scrutinising the survey results, there are no trends visible that can be attributed to teams belonging to either country. Results from companies in both countries display trends on the basis of the management principle and methodology and not on the basis of location. Projects taking place in both nations display a vast range of trends when it comes to team sizes, detection of defects and requirement changes. On studying the websites of the companies that participated in the survey, it is clearly visible that companies of both countries have onshore and offshore clients, and therefore follow globally accepted management principles and methodologies (Appendix 2). The first conclusion drawn from the survey results is that the country to which the teams belong does not affect the results or efficiency of a software development team. This is an important conclusion, because while going ahead with the data analysis, all results will be used and treated on the same level, regardless of which country the participant belongs to. In order to form a fair basis of comparison, it is important to look at the domain and size of the project in question. This has been explained further in the next section.

4.2.1 Companies and Domains

The companies that participated in the surveys varied in size. For confidentiality reasons, the question regarding which company the manager worked with was left optional in the survey. 16% of the participants revealed the companies with which they worked. Those companies include IBM, SAP, Version 1, Extentia Information Technology and SpiderLogic India. Some of the companies are MNCs on a global scale whereas others are mid sized companies based out of two or three locations. The question arises of how an MNC and a small company can be compared. To make the comparison fair, the Project Domains and Team sizes have been considered. The

question of Project Domain has been discussed below while the question of Team Size has been discussed in section 4.2.3.

The pie chart below displays the number of responses for each domain.

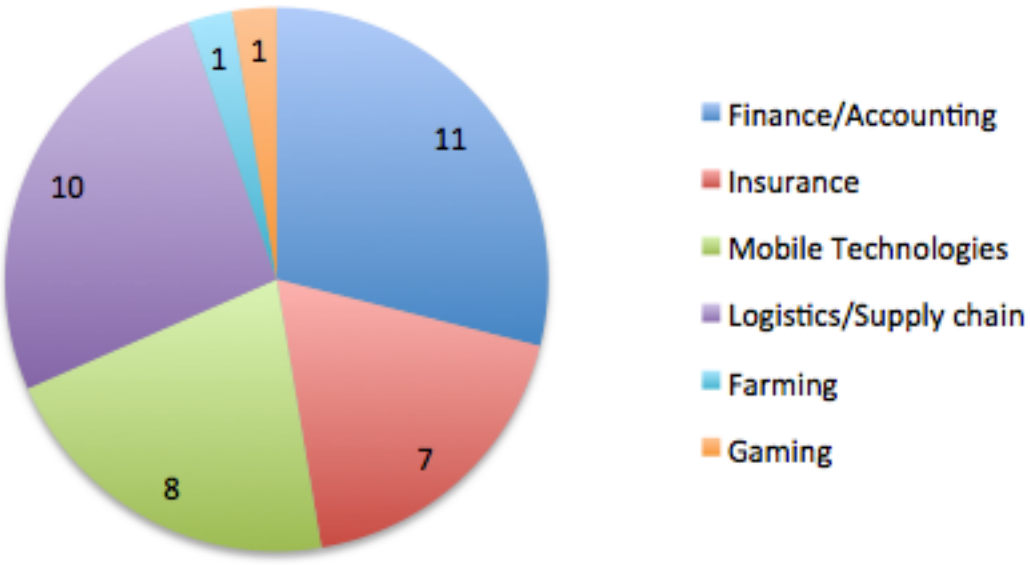

Figure 4.1: Chart showing number of responses for each domain

The project domains are mainly in the areas of finance, logistics, insurance and mobile technologies. However, there are two unique domains reported: gaming and farming. For the purpose of the study, the researcher is looking at the data received from all domains as most of the domains are recurring and can be used as a basis of comparison for other variables. Since two of the domains are not recurring, the variation is being neglected and all 38 responses are being used for the analysis and conclusions.

4.2.2 Usage of Lean Principles

The research being conducted aims to study whether using lean principles can make software delivery more efficient. Therefore, all results henceforth will be based on and divided on the basis of whether the companies use lean principles or not. In order to study if lean principles make software delivery more efficient, it is vital to compare parameters that measure efficiency of projects that use lean principles with companies that don't. The survey results reveal that 55% of the participants' projects follow lean principles while 45% of the participants' projects do not. While going forward, in order to make the presentation of the data easy, companies that do not use lean principles are being called 'non lean' companies.

The pie chart below shows the distribution of teams that use lean principles and teams that don't.

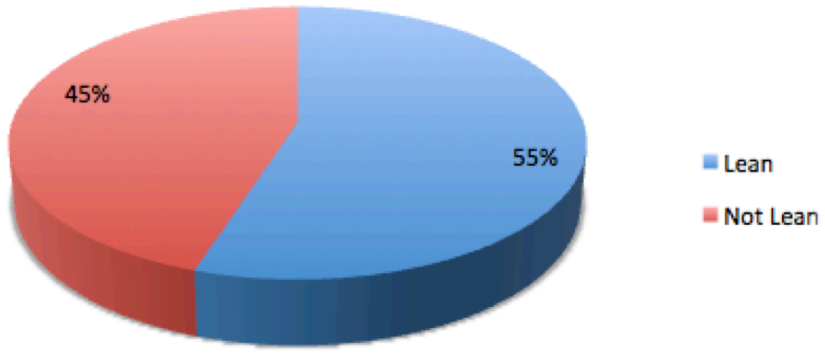

Figure 4.2: Pie Chart Showing the Usage of Lean Principles

4.2.3 Team Sizes

The survey results show that the team sizes vary from 4 to over 12 people. The maximum responses were between 7 to 9 people per team and the fewest responses were above 12 people per team.

The standard deviation of team sizes is 1.1. Therefore, it is clear that the variation of the data is not very high (Appendix 3). Because the standard deviation is not very high, it can be concluded that the team sizes are comparable and there is no need to leave out any of the extreme values while compiling results (Argyrous, 2009).

The responses for the size of the team can be viewed in the graph on the next page.

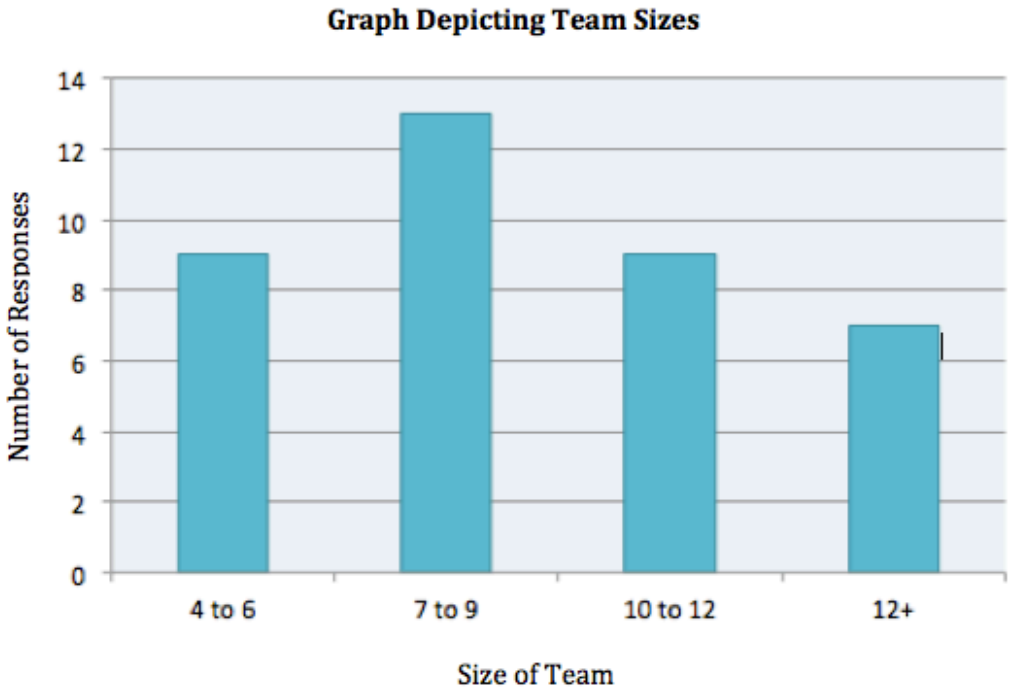

Figure 4.3: Graph depicting responses for team sizes

The team sizes have been divided on the basis of whether or not the team follows lean principles. The graphs below show the trends of team sizes on the basis of the management principle being followed.

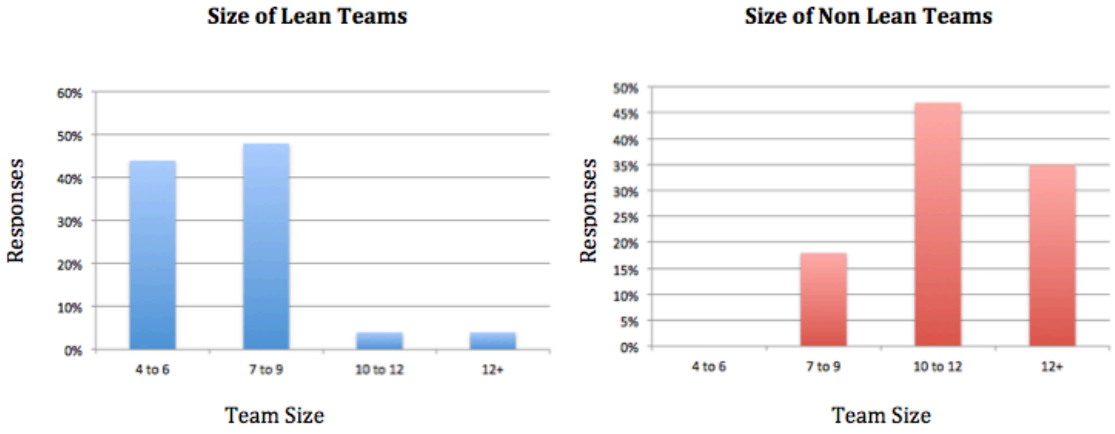

Figure 4.4: Graph depicting responses for Lean Team Sizes vs. Non Lean Team Sizes

On observing the graphs, it is visible that the number of team members can be correlated with whether the companies use lean principles or not. It is clear that the companies that use lean principles have smaller team sizes than companies that do not use lean principles. Companies that use lean principle have a high number of

responses for team sizes between 4 and 9 people. Companies that do not use lean principles have high responses for 10+ people in a team.

As mentioned in Chapter 2, Section 2.5, the software industry is dependent on the skilled workforce. One of the major costs contributing to the cost of a software project is the salary paid to the development team. If a smaller team takes the same amount of time to build an application, then it is more cost effective for a company to have smaller teams. As is visible, the teams that follow lean principles are smaller. However, it is also necessary to look at the time taken to deliver the software. For example, if a small team takes 10 weeks to build an application and a larger team takes 6 weeks, then it cannot be concluded that a smaller team implies lower costs. Therefore, it is required to study the two variables together: team size and release duration. Release duration has been discussed independently in Section 4.2.5 while a comparison of the two variables together has been discussed in Section 4.3.

4.2.4 Software Development Methodology Used

The question asked whether teams followed Agile or Waterfall and also allowed the participant to enter another methodology, 'Other' if applicable. All of the responses were either Agile or Waterfall. No participant selected the 'other' option. Agile was the more commonly used methodology, with 95 % using Agile and 5% using waterfall.

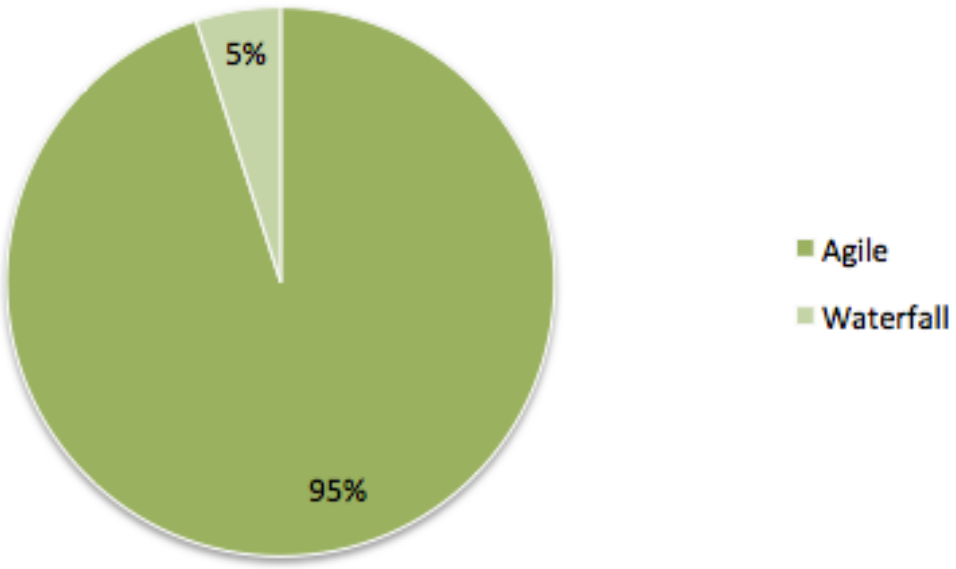

Figure 4.5: Pie Chart showing Agile and Waterfall Usage

100% of the companies using lean principles use the Agile Methodology. The only responses for the Waterfall model were from companies that do not use lean methodology.

4.2.5 Release Duration

Since there are only 2 results for Waterfall, there is not enough data to compare waterfall durations. The teams that used the Waterfall model had 3-4 year release durations and both teams did not use lean principles. In order to study how effective lean principles are, while going ahead with the research about release duration, the researcher only takes into consideration projects using Agile Methodology. The sample for studying the release duration consists of 36 companies, some of which follow lean principles and some of which do not. The release duration for an Agile project can be calculated using two measures: the length of each iteration or sprint and the number of sprints. For example, 8 sprints of 2 weeks each results in a 4 month release cycle. First, the length of a sprint will be considered. The graph below shows the range of responses as well as the number of responses for each option of sprint length.

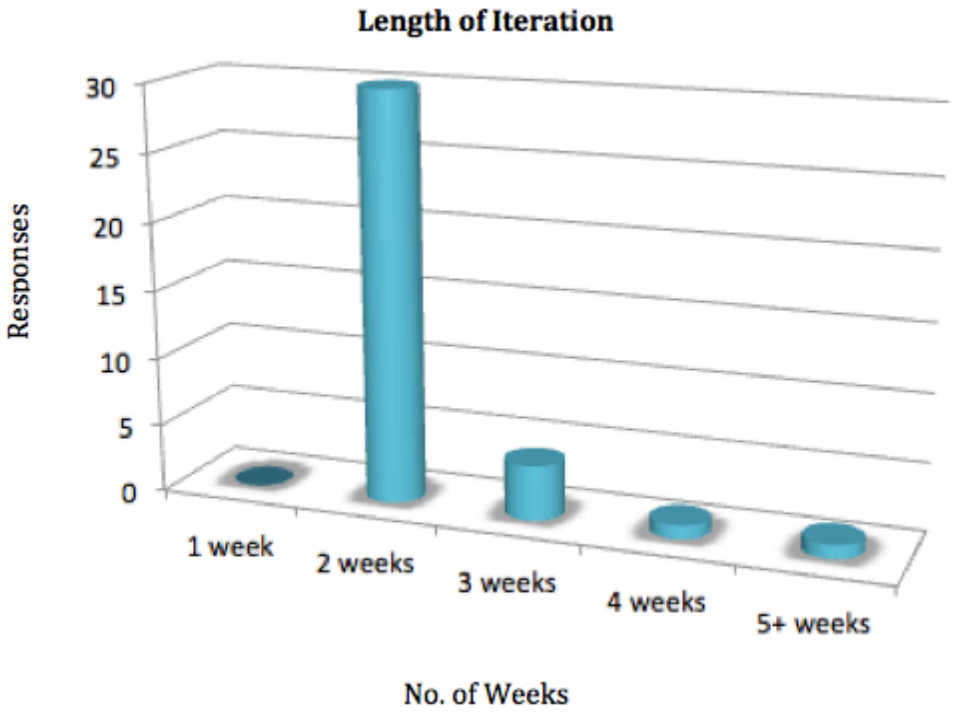

Figure 4.6: Graph showing the responses for length of an iteration

Since the highest responses for sprint length are 2-3 weeks, in order to form a clear conclusion about the duration of a release, only Agile teams with 2-3 week sprints will be considered. Thus, the sample size while studying the number of iterations reduces to 34 participants in order to reveal correct and accurate results.

The release lengths of the sample of 34 results range from less than 4 to over 10 iterations. There is a clear divide between the number of iterations needed by companies that use lean principles and companies that do not. Thus, the data has been divided into Lean and Non Lean projects. The graph below shows the trends of the length of release cycles for teams that use lean principles and teams that do not.

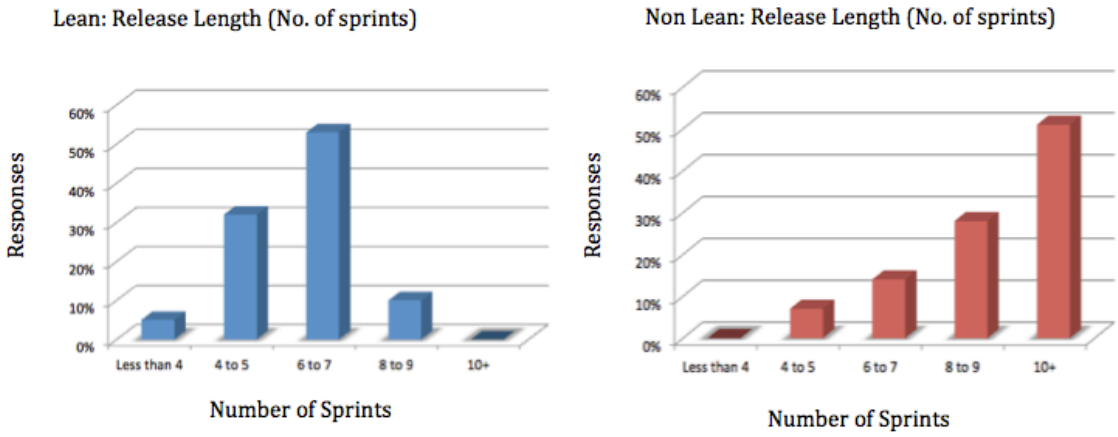

Figure 4.7: Comparison between number of iterations for Lean and Non Lean Projects

The graphs compare the number of iterations of teams that have the same iteration lengths. It is clearly visible that teams that use lean principles have a fewer number of sprints. This shows that teams that follow lean principles have shorter release cycles than teams that do not use lean principles. The release duration contributes to two parameters of measuring efficiency: time and cost. On its own, a shorter release cycle implies that software delivery is faster. However, in order to look at the cost, it is necessary to look at the team size and the release duration together. This has been covered in Section 4.3.

4.2.6 Detection of Defects in the Software Development Life Cycle

From this point onwards, the original sample of 38 participants will be considered. Regardless of the methodology that is used or the length of the release, all participants have answered the next set of questions. Defects are found in every project and need to be fixed in order to deliver the software. The question aims at determining at which point in the process of development are the defects are found.

From the point of view of detecting defects, the Software Development Life Cycle can be classified into four stages: Requirements definition and design, implementation or development (which includes unit testing), integration testing and system and user acceptance testing. Participants were asked to identify when most of the defects in the system were detected. There is a range of responses for

this question, which cover the entire Software Development Life Cycle. The responses can be seen in the graph below:

.

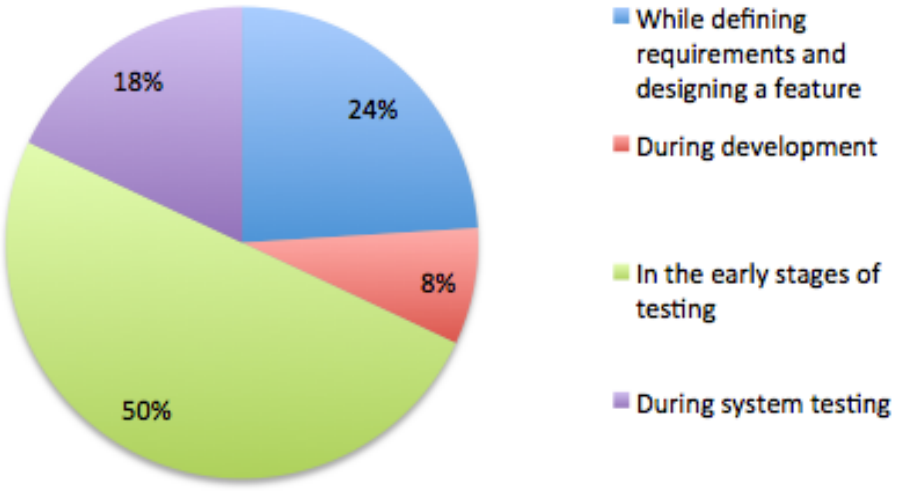

Figure 4.8: Chart showing when the most defects are detected in the SDLC

The study aims to find a correlation between lean principles and efficiency. In order to draw conclusions, it is necessary study when defects are located in projects that use lean principles and projects that do not.

The graph on the following page shows when the most defects are detected for companies that follow lean principles.

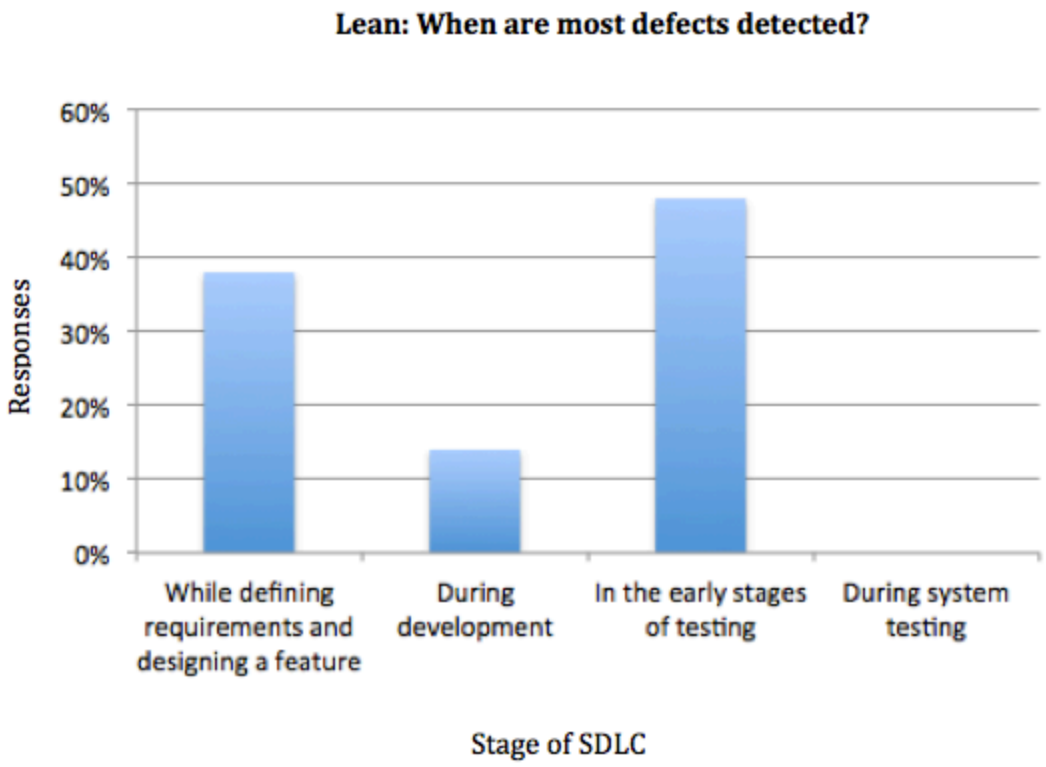

Figure 4.9: Chart showing when most defects are detected for projects using Lean principles

It is visible that most of the defects are located during the early stages of testing. However, a large number of responses say that defects are detected even earlier, i.e., in the requirements definition and design phase and during development. None of the participants said that the most number of defects is found during system testing. On the other hand, the graph below shows the detection of defects for companies that do no follow lean principles.

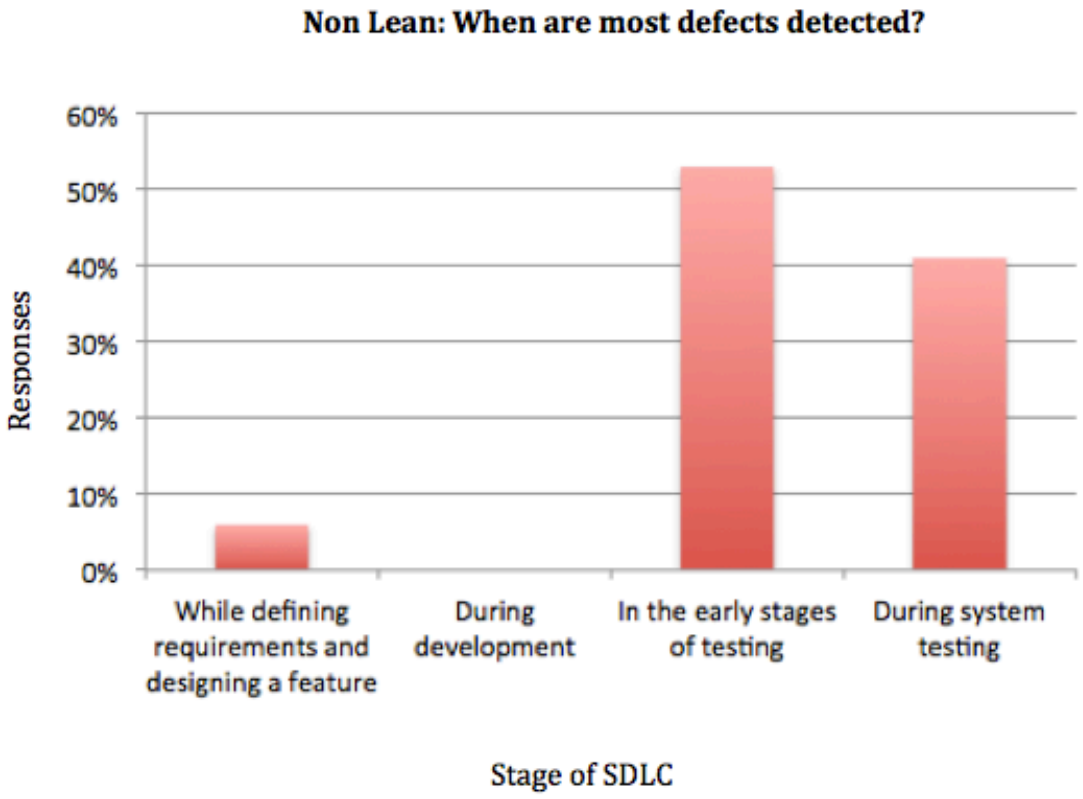

Figure 4.10: Chart showing when most defects are detected for projects not using Lean principles

From this graph, it is clear that teams that do not use lean principles discover defects later than teams that use lean principles. Like teams using lean principles, most of the participants answered that most defects are found during the early stages of testing. However, unlike teams using lean principles, teams that do not use lean principles discover a large number of defects in the last stage, i.e., during system testing. Very few participants locate defects during requirements definition and design while none responded that defects were detecting during development.

4.2.7 Regular Usage of the Total System in Production

These questions have been worded as:

'How much of the total system in production is used regularly by every user?'

'How much of the total system in production is never used by any user?'

The questions aim to understand how much of the total system is used by users on a regular basis and how much of the system is not used by any user. This allows the researcher to determine if any extra or wasteful work has been done. If the results for these questions have to be studied, the sample has to be divided into two groups:

projects using Lean principles and projects not using lean principles. The raw data on its own does not reveal anything concrete.

The graph below reveals the results of how much of the total system is used as answered by teams that use lean principles.

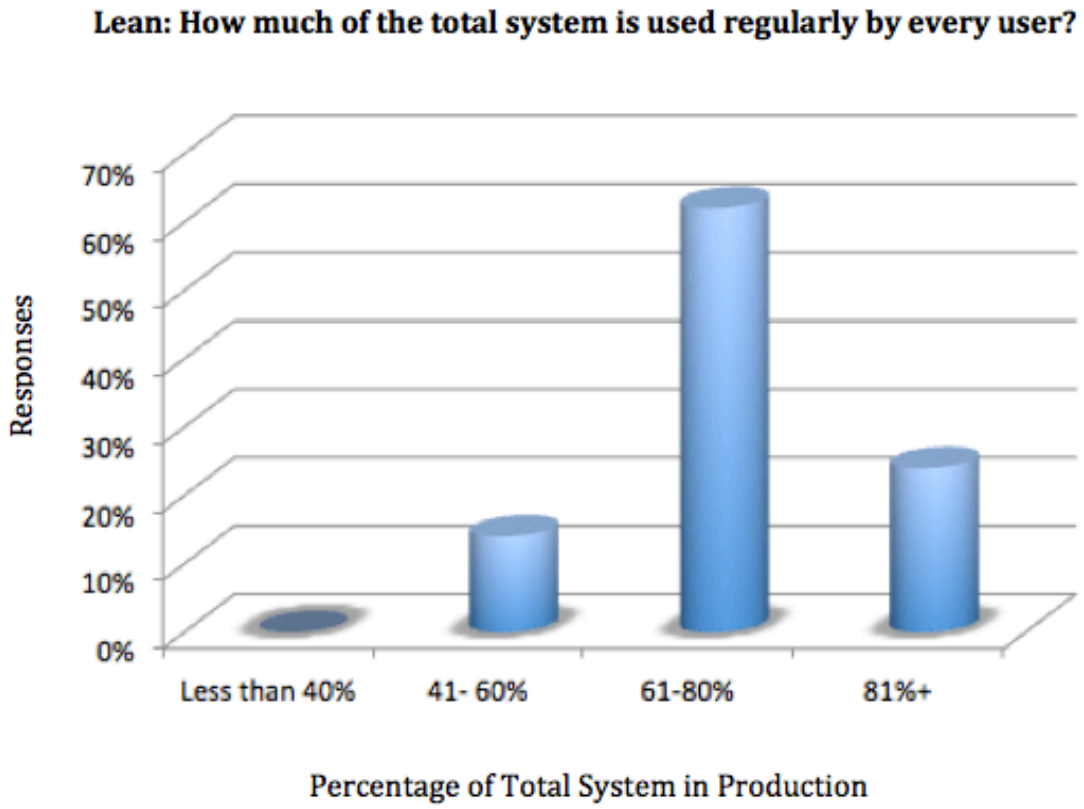

Figure 4.11: Chart showing the usage of the system for projects using Lean principles

For projects using lean principles, the average system in production that is used regularly by every user is higher than 41%. No participants have answered less than 40% while only 14% have answered 41-60%. 86% of the results exceed 61%. This shows that a large part of the systems developed by teams using lean principles are in regular use.

The graph below shows the usage of the system in production on a regular basis for teams not using lean principles.

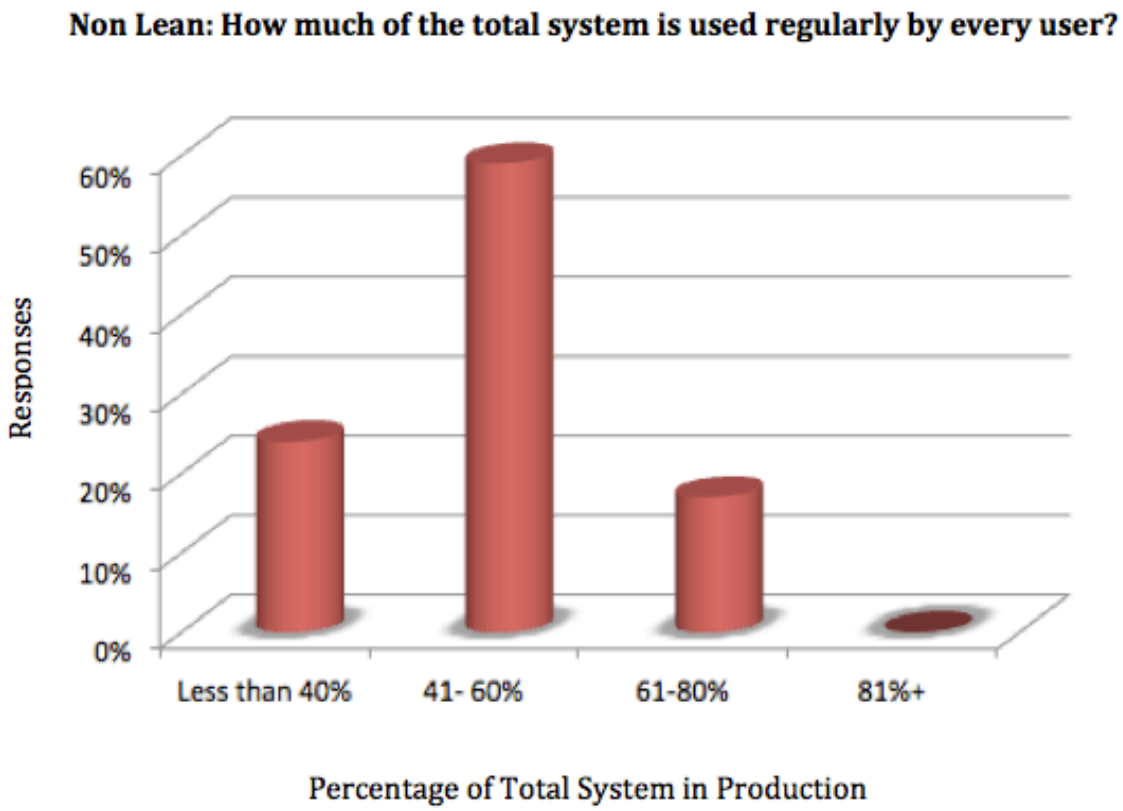

Figure 4.12: Chart showing the usage of the system for projects not using Lean principles

For projects not using lean principles, the average system in production that is used regularly by every user was lower than 80%. None of the participants have answered that over 81% of the total system is used.

This leads to the next question about how much of the total system in production is never used. The graph below shows how much of the total system is unused when the team that develops it uses lean principles.

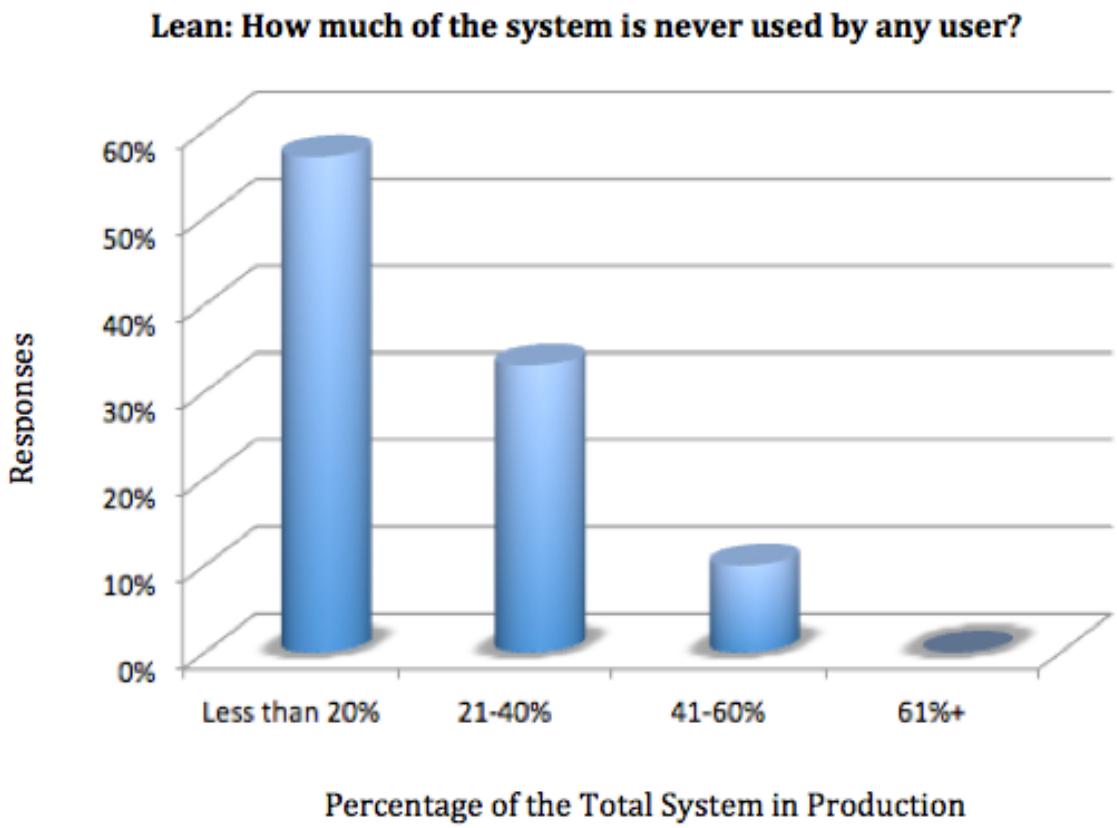

Figure 4.13: Chart showing the amount of the system unused when developed using lean principles

From the graph it is visible that when lean principles are used, a smaller percentage of the system is left unused. The highest percentage of responses says that less than 40% of the system is unused. This leads to the next question of how much of the system is not used when the system has not been developed using lean principles. The graph below shows this data.

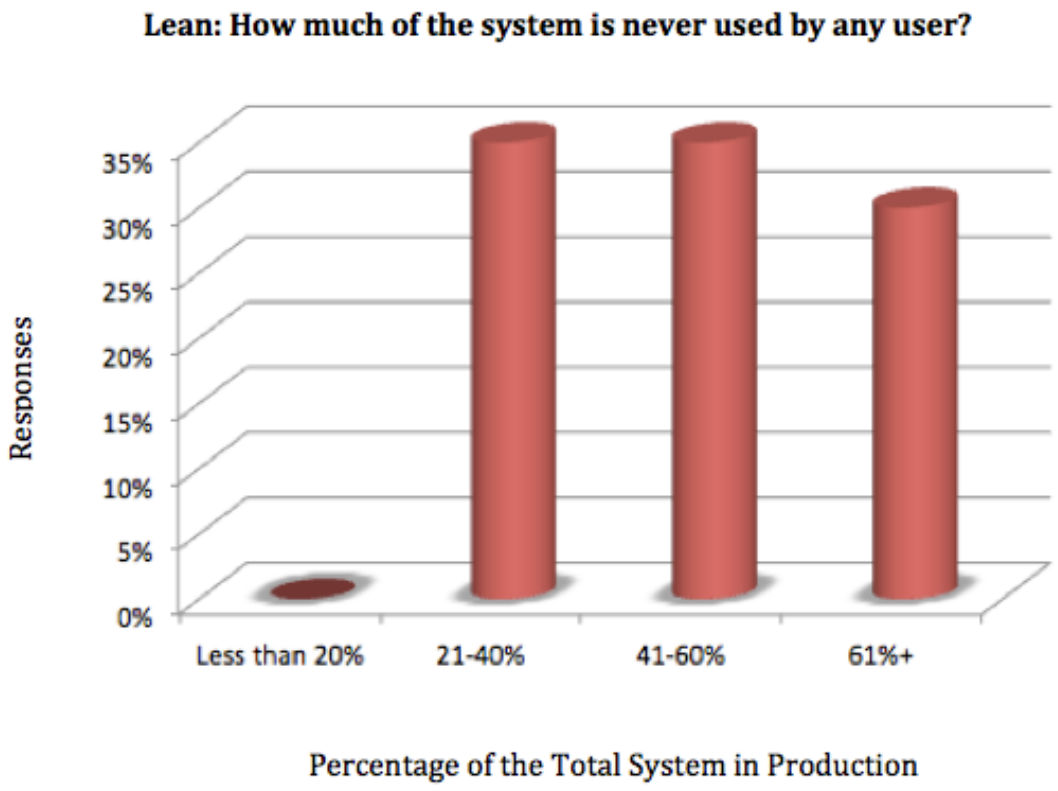

Figure 4.14: Chart showing the amount of the system unused when developed without using lean principles

From this graph, it is clear that the teams that do not use lean principles have a lot of their system unused by the users and clients. 35% of the participants said 21-40% of the system was unused and another 35% said that 41-60% of the system was unused. 30% of the participants said that 61%+ of the system was unused, which is a very high percentage of unused features.

4.2.8 Requirement changes

This question records how much of the requirements are changed after a functionality has been developed, causing the development team to work on the same feature again. Like section 4.2.7, this question also aims to determine the amount of waste that occurs in a development team. In all, there was a wide range of responses to this question.

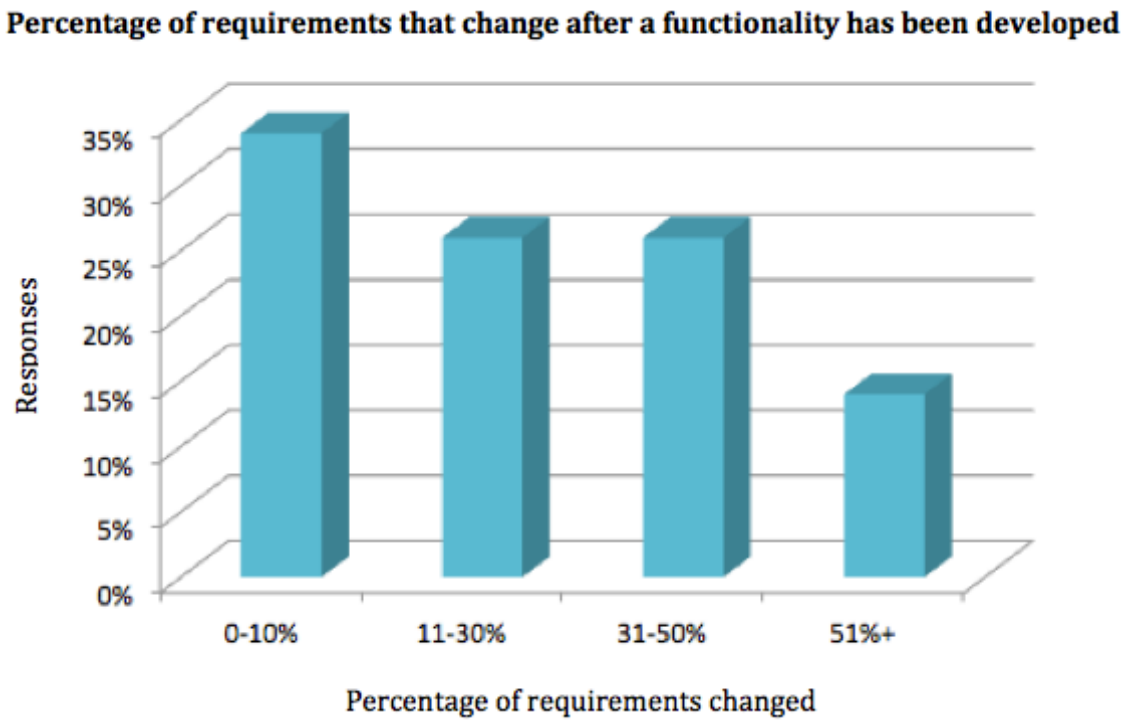

Figure 4.15: Chart depicting the changes in requirements of a functionality after development

Since the objective of the research is to study the effects lean principles has on software delivery, it is necessary to compare the results of the teams that use lean principles with the teams that do not. The results have been displayed below on the basis of whether the projects use lean or not.

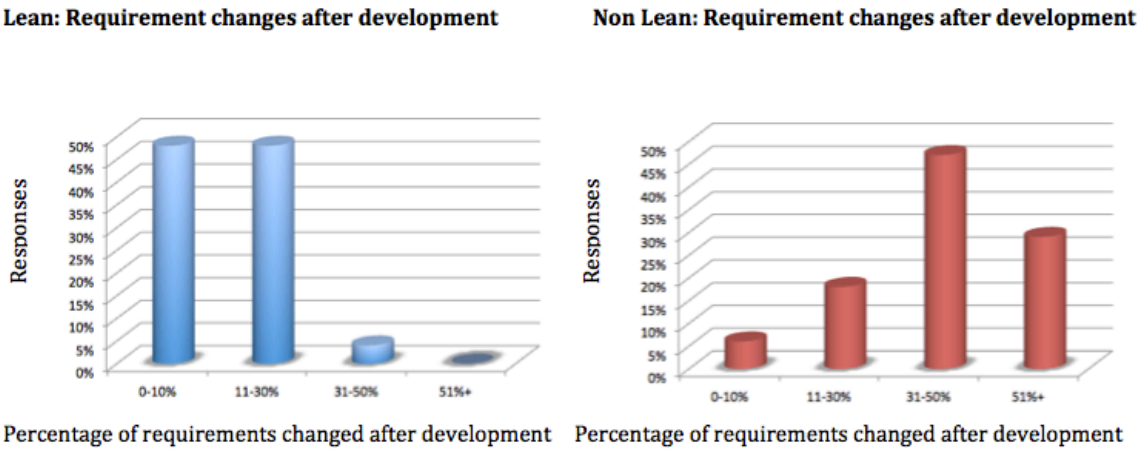

Figure 4.16: Comparison between requirement changes for Lean and Non Lean Projects

It is visible from the graphs that projects that use lean principles have distinctly lower requirement changes than projects that do not use lean principles. Projects that use lean principles have less than 30% requirement changes on an average,

whereas projects that do not use lean principles have over 30% requirement changes on an average.

4.3 Analysis of the Survey

4.3.1 Agile as the implementation of lean

As Bell and Orzen (2010) wrote, the use of Lean principles is a philosophy that has to be followed throughout an organisation, but while implementing lean principles in software development specifically, Agile is the methodology used. This is visible in the data provided. 100% of the companies that use lean principles follow agile methodology, whereas 88% of the companies that do not use lean, use Agile. The results drawn from this section of the survey do not reveal any new information or findings, but instead resonate with what Bell and Orzen write about Lean and Agile. The data about Agile usage is not being used to draw conclusions about the efficiency of Lean principles, but it is important for the rest of the research as the methodology being used determines the team sizes and release durations. If there were higher values of Waterfall models, it would be difficult to compare team sizes and release durations of Agile and Waterfall teams. Therefore, it is important to have found this information.

4.3.2 Team Size vs. Release Length

There are two parameters being used to determine the efficiency of software delivery: cost and time. While determining the time taken to release software, the results are clear: Teams that use lean principles deliver software faster than companies that do not. However, while measuring cost, there are many aspects that need to be kept in mind.

Chapter 2, Section 2.5 mentions that in the software industry, a large portion of the cost is attributed to the development team. For a software project, the clients are charged on the basis of man-hours. If the team is large, it means the salaries paid will be higher. The longer the team works, the more the cost attributed to the project. However, the two factors compete with each other (Kerzner, 2013). If the time is kept constant, a larger team will incur higher cost than a smaller team. However, if the team size is kept constant, the longer it takes to complete the project, the higher the costs. Therefore, to study the team size and release duration, the data has to be broken down into two parts:

A) Study Release Duration by Keeping the Team Size Constant
B) Study Team Size by Keeping the Release Duration Constant

Each of these studies has been conducted individually below:

A) Study Release Duration by Keeping the Team Size Constant

The average team size is 7-9. This is also the most recurring team size. In order to study release durations, it is necessary to keep the methodology and sprint length constant. Therefore the release durations of teams of 7-9 people, who use agile Methodology and two-week sprints, are being studied.

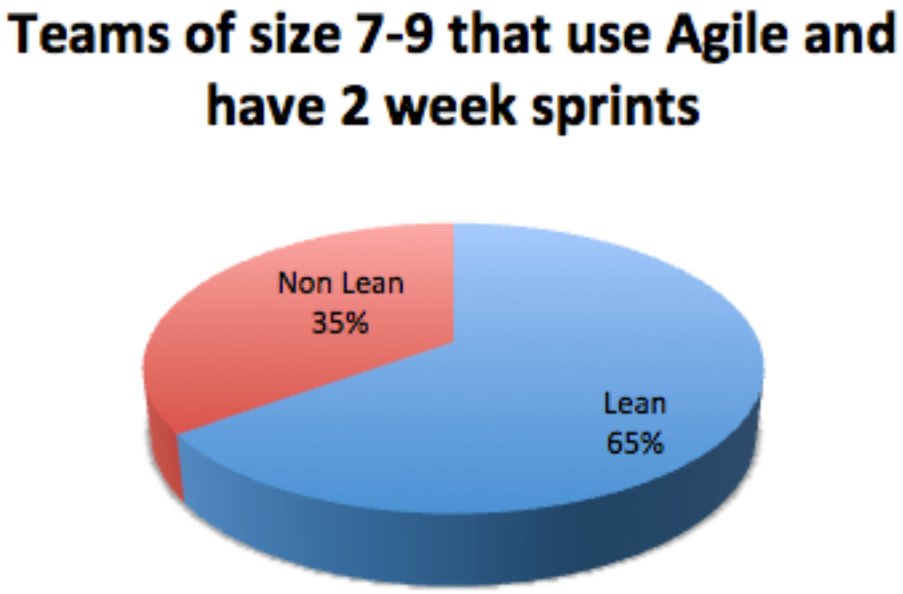

Figure 4.17: Usage of Lean principles for teams of 7-9 people

Out of the above mentioned data samples, the release duration is being mapped to whether teams follow lean principles or not. The graph on the next page show the results:

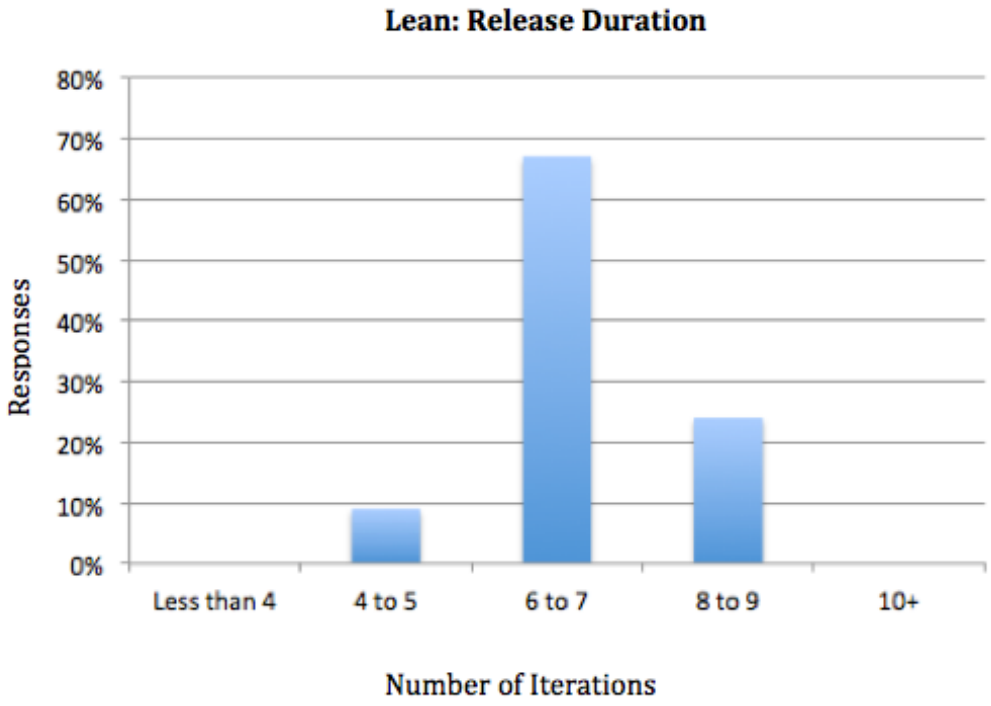

Figure 4.18: Lean: Release durations for teams with the same size, methodology and sprint length

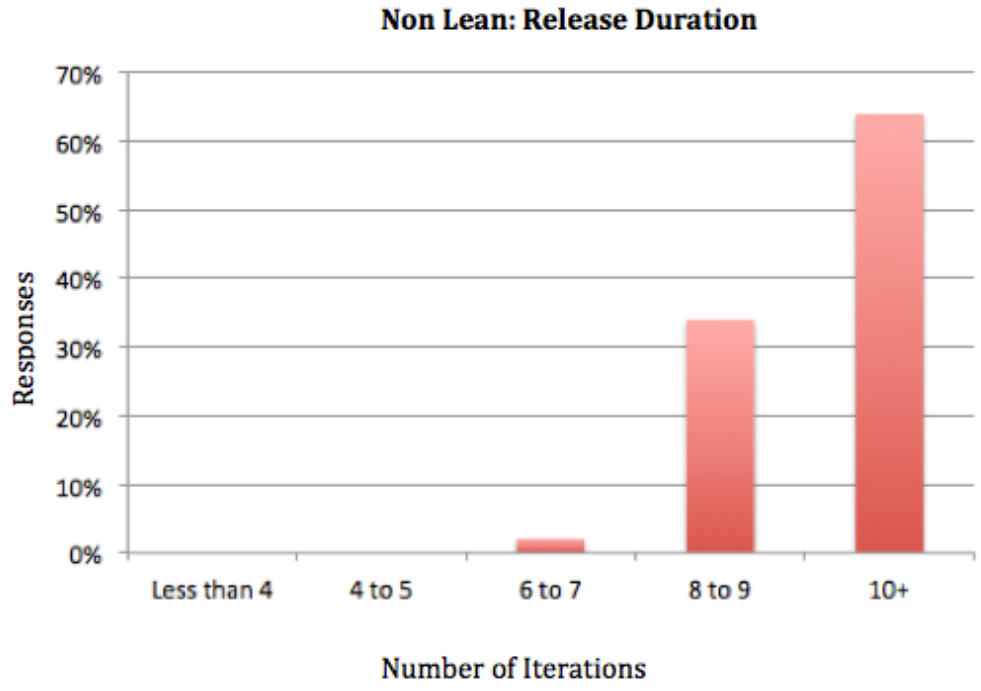

Figure 4.19: Non Lean: Release durations for teams with the same size, methodology and sprint length

The two graphs reveal the relation between release lengths and lean principles. Agile teams with two-week sprints and 7 to 9 people take less time if lean principles are followed than if lean principles are not followed. On an average, teams that use lean principles released their software 30% sooner than teams that did not use lean principles. This provides evidence that teams using lean principles release software faster than teams that do not use lean principles.

B) Study Team Size by Keeping the Release Duration Constant

While studying release length, the projects that take 6-7 iterations to release are mainly projects that follow lean, and projects that take over 10 iterations are projects that do not follow lean. There is only one response that has a fair division of lean and non- lean responses and that is 8 to 9 iterations. Therefore, though the sample is small, the value that will be kept constant is 8 to 9 iterations. Once again, the methodology being considered is Agile.

Figure 4.20: Usage of lean principles for agile teams that release every 8 to 9 iterations

Using the data samples as mentioned above, the team sizes are being mapped to the use of lean principles.

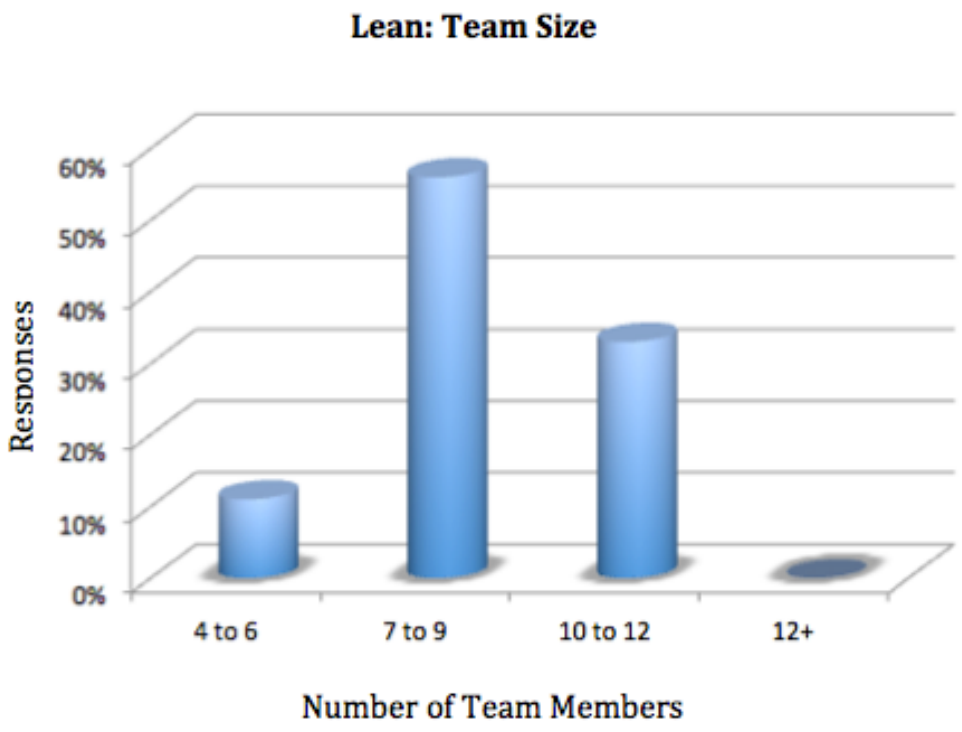

Figure 4.21: Lean: Agile team sizes when release length is 8-9 iterations

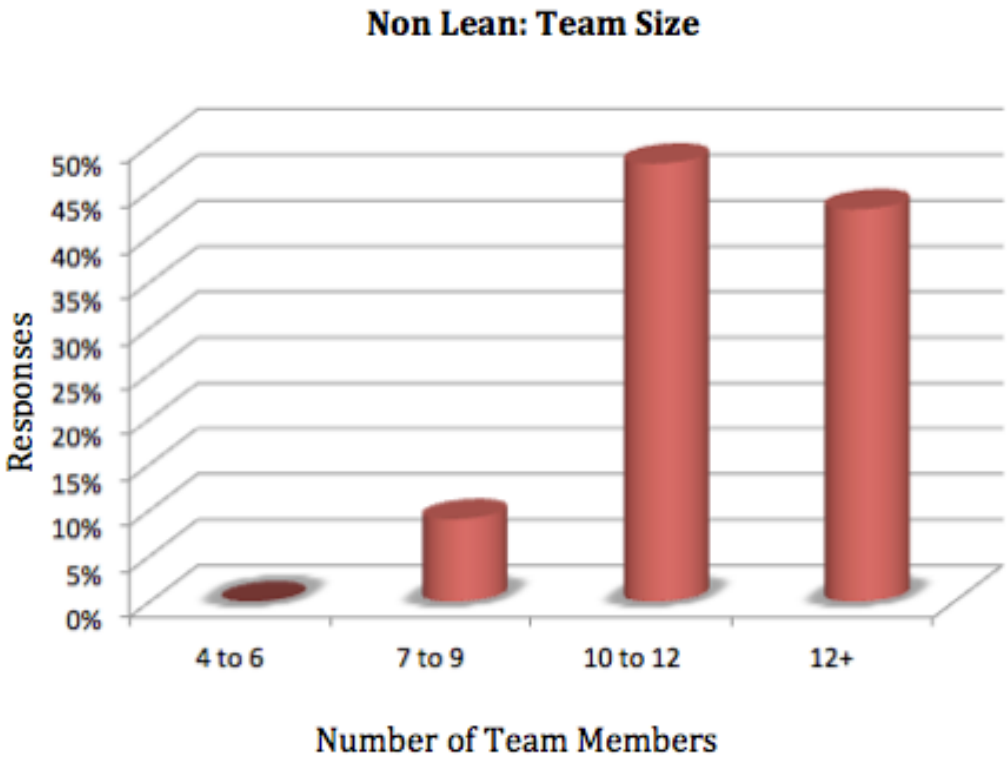

Figure 4.22: Non-Lean: Agile team sizes when release length is 8-9 iterations

From the graphs above, it is seen that on an average, teams that use lean principles are 33% smaller than teams that do not. This gives conclusive evidence that lean principles help to reduce team sizes.

4.3.3 Early Detection of Defects

As mentioned previously, the earlier a defect is detected, the cheaper it is to fix. The data shows that the most common stage where defects are located is in the early stages of testing. However, for teams using lean principles, there are a large number of responses for locating defects even earlier, i.e., in the requirements definition and implementation phase. None of the teams using lean principles responded that the maximum number of defects were located during system testing. For teams not using lean principles, although most of the responses showed early stages of testing, a large number of responses also showed that defects were detected during system testing. The option of discovering defects during the development phase has no responses from teams that do not use lean principles. On an average, for teams that use lean principles, over 50% of defects are located before the team goes into the testing phase, whereas for teams that do not use lean principles, over 90% of the defects are found in the early testing and system testing phases. From this, it can be concluded that unit test are not being conducted and defects are being discovered very late in the Software Development Life Cycle. There is a high cost associated with this and it can also be a reason that the release durations are longer. If defects are detected later, the time and effort to fix them will be higher, as well as testing time will increase. A bug fix later in the cycle might cause existing functionality to break, thus increasing the need for re testing and regression testing and a possibility of a regression defect. This will add to the time taken to release the software as well as the cost of development.

4.3.4 Waste in Software Development

Although waste is not easily tangible while talking about software delivery, there are a few areas where waste can be identified. It can be said that the work done was a waste either if the feature is in production but is not being used, or if a feature was developed and then requirements were changed, leading to the development team having to rebuild a feature.

As seen is Chapter 2 Section 2.6, Souza et al (2013) wrote about how the Pareto Principle could be applied to the software development, resulting in only 20% of the totally system providing 80% of the value of the software product. This shows that there is a large portion of the software that goes unused. Any feature that has been developed and released to production and is not being used can be considered waste. Since lean principles stress on waste reduction, this is a very important section of the study. The objective of the study is to reveal whether using lean principles has had effect on the amount of waste in software development. The results show a clear distinction between teams that use lean principles and teams

that do not. Projects that use lean principles have a larger percentage of system in production that is in use regularly. Also, projects that use lean principles have a lower percentage of the total system in production that is not used by any user. This is a positive result as the foundations of lean principles are waste management. Waste adds to both parameters of efficiency in software delivery: time and cost. Developing a feature that nobody uses adds to the cost of the project and takes time away from developing a feature that is of more value to the customer. From the results, it is clear that teams that use lean principles spend more time on features that do add value to the customer and spend less time and effort on wasteful tasks. In all, the results show that on an average, more than 70% of a system is used regularly and less than 20% is never used when lean principles are applied. When lean principles are not applied, only about 50% of the system is used regularly whereas over 50% of the system is never used by any user.

As derived from the results of the question regarding requirement changes, teams using lean principles tend to have fewer requirements changed after a feature has been developed. The results showed that over 90% of the teams that used lean principles had less than 30% requirement changes whereas over 80% of the teams that did not use lean had more than 30% requirement changes after a feature was in production. Fewer requirement changes cut down the waste of time and cost that goes into rebuilding a feature and disposing of features that were already developed. Teams that do not use lean principles record a higher percentage of requirements changes, thus making it clear that the use of lean principles have a direct impact on the amount of wastage during software delivery.

4.4. Analysis of Research: Case Study and Survey

The Case Study and Survey when studied together provide similar results. Both the studies show that lean principles have an effect of the efficiency software projects. The Case Study did not provide and link between team sizes and the use of lean principles, but the Survey allowed the researcher to compare team sizes with release duration in order to draw a connection between the variables.

While the Case study was for a company that followed Agile, the Survey covered companies that followed both Agile and Waterfall. To have a fair basis of comparison, while comparing release lengths, only Agile teams using two week sprints were considered among the Survey participants. The case study directly revealed that the release duration was decreased after the implementation of Lean principles. In the survey, this result had to be derived by comparing the iteration length of teams that use lean principles and teams that do not.

In the Survey, the questions about waste have been added because the question of cost cannot be asked directly. These questions are unnecessary in the Case Study because the changes in cost have been asked directly. The questions about waste have been designed in order to draw a conclusion about cost. On studying the cost aspect, the participant in the survey show low waste, which can be attributed to

lower costs. The case study directly answers the question about a reduction in cost of projects. Both the results together show the same trends, and it can be derived that lean principles help to reduce costs of projects.

The Case Study independently does not provide results that can be generalised, but when the results of the Survey resonate with those of the Case Study, the findings can be generalised and applied to the population at large.

Chapter 5

Conclusions and Recommendations

Chapter 5
Conclusions and Recommendations

5.1 Conclusions

Before the conclusions are drawn, the research question needs to be brought to focus.

Research Question: Can lean principles be applied to the Software Industry for more efficient software delivery?

The parameters to measure efficiency for this study were chosen to be time taken to release the software and cost of the project. The conclusions drawn relate to each of these parameters individually. They have been discussed below:

5.1.1. When Applied to the Software Industry, Lean Principles Make Software Delivery Faster

From the results of the case study and the survey, it is clear that the use of lean principles makes software delivery faster than if lean principles are not used. The same company recorded clear reductions in the time taken to deliver software from the time that they hadn't implemented lean principles to the time when they had implemented them. On comparing projects of the same size and iteration lengths, the survey proved that the use of lean principles helped to reduce the duration of a release. Therefore, it is evident from both studies that lean principles make software delivery faster. Faster software is undoubtedly important for customer satisfaction. However, it also has an impact on the cost of the software project. This has been discussed below, in Section 5.1.2.

When clients hire a company to develop an application for them, or if a company has an IT department that develops software for them, they require the software in order to fulfil a business need. The companies that participated in the survey were developing software for clients in sectors like finance, insurance and supply chain. Their clients would have their own timelines in which they would need to implement their process changes and introduce or upgrade their software systems. If the software systems were delivered later than expected, it would cause further processes to be disrupted, plans to be changed and would set off a chain of events affecting both the software company and the client company. Focus on customers in one of the most vital lessons that companies can learn from using lean principles. In order to provide better service to clients, it is vital to keep their requirements in mind. Companies that follow lean principles are trained to keep abreast of customer requirements, and therefore quick software delivery is an integral part of the lean principles when applied to software industry. The results of the research show that the use of lean principles will provide faster software delivery, thus fulfilling a basic need of the customer.

5.1.2. When Applied to the Software Industry, Lean Principles Make Software Delivery More Cost Effective

In every business, the final result that is looked at by all stakeholders is the profit (or loss) made by the company, the project or the operations. Profitability will be high if the revenue from a product or service is greater than the cost. If the cost of a software project is low, the companies that undertake the project as well the client companies will have a high profitability. Lean principles focus on cutting out all wasteful processes, procedures and expenditures. From the results of the case study, it is clear that the costs of a project reduce of lean principles are followed. Cost is calculated taking factors things into account:

1. *Size of the team:* Since the cost of a software project is directly related to the size of the team, the smaller the team, the lower the cost. Although the case study did not reveal a relationship between the use of lean principles and the size of the team, a majority of the responses of the survey proved that there is a connection between the two variables. The teams that follow lean principles had a larger number of responses for smaller teams and teams that do not follow lean principles had a larger number of responses for larger teams. Thus, it can be concluded that lean principles help to reduce the team sizes and therefore help to reduce the costs of a software project.

2. *Time taken to deliver:* The release duration has been discussed above in Section 5.1.1. Quick releases are an important aspect of efficient software delivery, but the time taken to develop the software also relates directly to the costs of the project. The longer a project takes, the higher the cost associated with it. Therefore, the speed and cost of a project are very closely related. Since the study has shown that there is a distinct difference between the speeds of teams that use lean principles and teams that do not, the research provides conclusive evidence that the cost is lower for teams that follow lean principles than teams that do not.

3. *Amount of waste*: Lean principles lay stress on waste management. The amount of waste can increase both the time taken for the project as well as the cost of the project. For the purpose of the research, waste was quantified in three ways: how long it took for defects to be discovered, how much of the total system in production was used and how much was not used, and how much of the requirements were changed after a feature was developed. The survey covered all of these aspects and revealed that the projects that used lean principles had considerably less waste than projects that did not use lean principles

Since each of the factors used to measure areas of cost in a software project reveal positive results with respect to cost reduction, it can be concluded that the cost of software projects is lower when lean principles are used.

Time, Cost and Quality are three important measures against which the success of a project is measured. If high quality software is delivered on time and at low cost, it can be classified as successful (Elzalmy and Hussin, 2014). For the purpose of the study, quality of the software has not been studied. However, the fact that all the participants were part of reputed software companies that released software that was accepted by the clients, it can be concluded that the quality of all the projects met the required standards. No comment can be made about whether one project released better software than other. However, from the point of view of the stakeholders like the Management and the clients, the time that the software was shipped and the cost of the project are also vital in determining the success of a project. From the results of the study conducted, it is evident that when implemented, lean principles help to reduce both the time taken and the cost of a project. Thus, it can be said that projects that use lean principles are more successful than projects that do not.

If a software project is deemed as successful, it means that the clients will be satisfied. The basis of lean principles is to make sure that the clients' requirements are met, not only with respect to the software application, but the all round product and service. However, while using lean principles, the clients' needs are met but not at the expense of the development team. Lean principles help to improve processes and remove wasteful work, and these are important to the general satisfaction of the development team as well. If the development team does not have to work overtime, redo large portions of their work or work on a functionality that they feel is of no use, they will be more motivated to work and the morale of the team will be high. From this, it can be concluded that lean principles provide a 360° improvement to a software project.

5.2 Discussion

Going back to the beginning of the research, the hypothesis stated that, "Yes, *the use of lean principles in the software industry will help to make software delivery more efficient."*

The conclusions of the research are positive and prove that the hypothesis is true. The results of the study were positive and for this reason, further research in manner of negative research questions is not required *(Negative questions if the hypothesis was proven false: Does the use of lean principles have no effect/a negative effect on the efficiency of software delivery?)*

If a further study needs to be taken up, it should be about the effect of lean principles on the quality of the software. Since this research was of a limited time, it had to be narrowed down in order to maintain the scope. However, if the quality of software can be traced to the use of lean principles, it will make this management principle highly sought after by the software industry. Already, the use of lean principles has grown in the field of software development. If it is possible to prove that the use of lean principles leads to faster, cheaper and better software, both

clients and development teams will be interested in implementing the principles immediately. However, as Liker (2004) wrote, lean principles are not a process that can be implemented, they are a philosophy that needs to be adopted by every level of the organisation, and put into practice by continuous learning.

5.3 Recommendations

After conducting the study, there are a few recommendations that can be made on the basis of the findings and analysis. The recommendations are directed at various different parties, i.e., companies that use lean, companies that don't use lean, companies that use agile, clients who need custom software products or services and researchers in the field of Lean. Each of the cases has been discussed separately.

5.3.1 Software companies that do not use lean principles

Since the software industry is very dynamic, it is important for software companies to keep track of developments and innovations in this field, both in the area of technology as well as management. Companies should be aware of their competitors' management principles and processes and be updated about changes taking place in the industry. If a company decides to implement lean principles, they should implement it in the entire organisation, as it is a philosophy and not merely a process. The company should not expect immediate results, as it is not a change that can happen overnight. The company that was studied in the case study took over 5 years to implement lean principles throughout the organisation. However, once implemented, the results showed marked improvements in efficiency. If a software company wants to use lean principles, they should understand that though Agile is the manifestation of lean principles in software development, it is not the only step in the lean transformation. There are many companies that follow Agile but are not lean. Therefore, the important thing to keep in mind is that it is the philosophy of lean that is important while implementing lean principles.

5.3.2 Software companies that use lean principles

Continuous learning is an integral part of lean principles. If a company uses lean principles, the recommendation is that it should be acknowledged that implementing lean is not something that happens once and will automatically mean that the company is lean. For a company to remain lean, the employees of the company have to carry on the legacy by actively implementing the principles at every step along the way. Nobody is perfect and there will be mistakes made along the way, but the spirit of lean is that it is a learning process and one has to learn from one's mistakes. There is always scope for improvement and if the study says that lean principles can make software delivery faster and more cost effective, there is still scope for reducing the development time and managing waste to reduce costs. If everybody on the team adheres to this outlook, there will be more improvement in efficiency.

5.3.3 Software companies that use Agile methodology

From the results of the survey, it is clear that all teams that use agile are not lean, but all teams that are lean use agile. That means teams that use agile are not necessarily lean, but they are implementing processes that lean propagates. Therefore, if such a company decides to make a lean transformation, the transformation will be easier than if the company followed ad hoc methods or the Waterfall methodology. Some of their processes will already be in place, and it will be the outlook and philosophy that will need to be changed, and not the entire software development methodology. Like the company that was studied in the case study, when a company using Agile methods decides to use lean principles, the transformation can be smooth and deliver concrete results. The recommendation to teams that use Agile is to get a clear understanding of the clients' requirements before implementing to avoid having to rebuild a feature. Since the Agile methodology stresses on changes being easy to implement, the changes should be made in the requirements or development stages and not in the phase of system testing. This is where having small sprints helps; if changes are required to be made, they can be implemented before that part of the system has been integrated with a large part of the system. Thus, making the change will not have an effect on many other parts of the system. This helps to reduce waste and will help an Agile team to be lean.

5.3.4 Companies that need custom software developed

If a company needs a software system for its business requirements, it should consider the quality of the software as well as the cost and timeline before selecting a software company. If feasible, the client should consider working with a company that uses lean principles. Since lean principles focus on customer requirements, their needs will be met while being cost effective and delivered quickly. If the clients have a clear understanding of their business requirements, it will make the software delivery process more efficient and reduce the time and the cost of the software development project.

5.3.5 Researchers in the field of IT Project Management

As mentioned in section 5.2, a researcher in the field of IT Project Management should consider tracing the effects of using lean principles to the quality of the software that is developed. The study will have a wide and interesting scope it can use this research that was conducted as a basis for further study. Quality Assurance in terms of software development is very important and if philosophies and processes around software testing and quality assurance can be implemented, it will improve the overall experience of delivering software products and services.

5.4 Future Directions

Duncan and Ritter (2014) write about how lean principles will be adopted by various other industries like the service industries like hospitals and restaurants and fields like marketing and finance. According to them, there will be major improvements in the approach and execution of lean principles and over the next 20 years and concepts like waste management and customer focus will be the prime values in all industries. Onetto (2014) writes about the adoption of lean principles in the e commerce sector, led by the company Amazon. While Amazon use lean principles throughout the organization, their IT management is also being transformed into a system that uses lean principles.

For the software industry, there will be changes in the implementation of lean principles as this field grows and develops and there will be continuous improvements in the processes and methodologies. Companies that follow lean principles effectively will be trained to accept the changes, as continuous learning is one of the main features of using lean principles. Therefore, companies will learn the correct implementation of lean principles. Since it has been proved that lean principles make software delivery more efficient, the efficiency of software delivery will rise incrementally in the future as more companies adopt lean principles and gain experience in how to make it work for their benefit.

Chapter 6

Self Reflection

Chapter 6
Self Reflection

When I signed up for the MBA programme at Dublin Business School, I was aware that the course required students to write a Master's Thesis. Still, if someone were to tell me that by the end of a year I would have a Master's Thesis complete, I would never have believed them. Looking back I would not have thought myself capable of achieving something like this. Now as I write the last chapter, I am proud and honestly, a little relieved to say that I exceeded my own expectations.

When the course commenced, I didn't give much thought to what topic I would choose, or even which broad subject area I would select for the Thesis. I wanted to attend classes and study a few modules before making a decision. However, on the first day of the first semester, I attended a Research Methods class and was struck with the realisation that within a couple of months, I had to submit not only a topic, but also a rough idea how to go about my research. I panicked and immediately began to try to reach a decision about which subject area I would choose for the Thesis.

The first thing I learnt about myself was that I panic often, and easily.

As the semester continued, I decided that I wanted to conduct research in the field of IT Management because of my professional background in the field and a desire to pursue a career in the same field. I wanted to select a new area of study, something that had not been studied in great detail before. With the help of my father, I identified Lean Principles as a new and upcoming area in the field of IT Management. As I read about the topic, my interest in it grew and I chalked out my plans for the research. Everything went smoothly, until after the second semester, when I received my result for the Dissertation Proposal and I thought it was lower than expected.

Once again, I panicked.

It was at this time that my Supervisor, Patrick O'Callaghan helped to get me back on track with the research. With his help, I learnt that the marks were not as important as the feedback. I tried to incorporate the feedback into my dissertation and use it as a way to improve my work rather than be upset about a low mark and not take any corrective action.

At the beginning of the academic year, I had an abject fear of public speaking. With each module in the MBA including Research Methods, I gradually overcame this fear. I realised that it was all about practice; the more presentations I gave, the more confident I became. Initially, I was very anxious before presentations and I panicked every time I heard that an assignment included a presentation. But by the end of the two semesters, I was more calm and composed and I was able to present my work

confidently. Presentations skills are very important in one's career, and I will always use this skill in the future to get ahead in my career.

While working on the dissertation, I not only gained knowledge about the topic, but also about research methods. While doing the literature review, I realised how important it was to use reliable sources. Since my topic is in a relatively new field, there are a lot of websites and blogs that publish articles in the field. However, when compared to journal articles, some of the blogs and websites do not show enough evidence of research. Going forward in the dissertation, I was surprised to find out that studying any topic required the knowledge of research methodology and that the methods chosen provided a guideline on how to go about the research, from the philosophy selected to look at the problem, till the data gathering and analysis section. A large section of the study involved designing the research methods to be adopted. While conducting the research, an important skill that I picked up was data gathering and analysis. I feel that I will be able to apply this to my career.

The dissertation phase of the course taught also me about how to manage a project myself. I have never worked without a boss or had a submission without an impending due date. This was the first time that I was my own boss and had to manage myself. I learnt about time management the hard way, when I had lost some time and I was lagging behind in my work. However, I also learnt that if I put my mind to it, I could catch up and get back on track. I realised the importance of having a project plan and following it, but I also learnt that it is important to be flexible and be able to change the plans in case of contingencies.

Due to the assignments and the dissertation, I had to give up one of my bad habits: procrastination. In the first semester, I had procrastinated and left some essays incomplete until the last minute. I learnt from experience that this was not feasible in the second semester, and I improved my focus and concentration in the second semester. When it was time for the thesis, I realised that it could not be left till the last minute under any circumstance. I realised that it would be physically impossible to complete the Thesis in the last fortnight, so I made a conscious effort not to succumb to temptation. I was distracted from time to time and I did divert my mind from the Thesis, but I tried to maintain a constant flow of work.

Living away from home during the course taught me a few important life lessons. If I had stayed with my family, I would have been irresponsible and not had to worry about anything apart from work for the dissertation period of the course. However, living in a foreign country made me more responsible, teaching me that just because I had a deadline, I couldn't stop routine activities like cooking and cleaning. I had to learn how to balance work and life and I think it has made me more responsible as a person. Every time I panicked, I learnt that although I did have the support of my professors, parents and friends, I was largely independent and I had to learn to work through it on my own.

The Thesis had a very direct and relevant impact on my career. One of the companies that participated in my survey has invited me to interview with them for a prospect of a job. The future looks positive and after one year of my MBA, I feel that I have developed from a carefree child to a responsible adult.

References

References

1. Abandah, H. and Alsmadi, I. (2013) **'Call Graph Based Metrics To Evaluate Software Design Quality'**, *International Journal Of Software Engineering & Its Applications*, 7, 1, pp. 1-12, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 22 July 2014]

2. Anderson, D. (2012) *Lean Software Development* [online] Msdn.microsoft.com. Available at: http://msdn.microsoft.com/en-us/library/hh533841.aspx [Accessed: 29 June 2014]

3. Armour, P. G. (2013) '**The Business of Software: How We Build Things...and why things are 90% complete'**, *Communications Of The ACM*, 56, 1, pp. 32-33, Business Source Complete, EBSCO*host* [Accessed: 21 July 2014]

4. Armour, P. G. (2014) **'The Business of Software: Estimation Is Not Evil',** *Communications Of The ACM*, 57, 1, pp. 42-43, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

5. Argyrous, G (2009) *Statistics for Research,* 2nd Edn, London: SAGE Publications

6. Atkinson, P. & Nicholls, L. (2013) **'Demystifying Lean Culture Change and continuous improvement (cover story)'**, *Management Services*, 57, 3, pp. 10-15, Business Source Complete, EBSCO*host* [Accessed: 29 June 2014]

7. Babb, J., Hoda, R., & Norbjerg, J (2014) **'Embedding Reflection and Learning into Agile Software Development'**, *IEEE Software*, 31, 4, pp. 51-57, Business Source Complete, EBSCO*host* [Accessed: August 2014]

8. Bell, S. and Orzen, M. (2010) *Lean IT: Enabling and Sustaining Your Lean Transformation.* 1st ed. Boca Raton: Productivity Press.

9. Berson, B.L. (2014), **'CAN YOU IMPROVE YOUR ORGANIZATION'S EFFECTIVENESS?'**, *Quality*, 53, 6, p. 23, Business Source Complete, EBSCO*host* [Accessed: 25 July 2014]

10. Bhasin, S (2013) '**Impact of corporate culture on the adoption of the Lean principles'**, International Journal of Lean Six Sigma, Vol. 4 Iss: 2, pp.118 – 140 Emerald Insight [Accessed: 29 June 2014]

11. Bolte, J.L. (2014) **'The big question about LEAN SIX SIGMA'**, *Industrial Engineer: IE*, 46, 4, pp. 50-53, Business Source Complete, EBSCO*host* [Accessed: 25 July 2014]

12. Brightwater Recruitment Specialists (2014), '**Salary Survey 2014**' (Ireland) Available at: http://www.brightwater.ie/documents/brightwater_salary_surveys_ROI_2014.pdf [Accessed: 20 June 2014]

13. Camacho-Miñano, M., Moyano-Fuentes, J. and Sacristán-Díaz, M. (2013), **'What can we learn from the evolution of research on lean management assessment?'**, *International Journal Of Production Research*, 51, 4, pp. 1098-1116, Business Source Complete, EBSCO*host* [Accessed: 21 July 2014]

14. Cheung, S., Yiu, T., & Lam, M. (2013), **'Interweaving Trust and Communication with Project Performance'**, *Journal Of Construction Engineering & Management*, 139, 8, pp. 941-950, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 21 July 2014]

15. Čiarnienė, R. & Vienažindienė, M. (2013) **'Lean Manufacturing Implementation: The Main Challenges And Barriers'**, *Management Theory & Studies For Rural Business & Infrastructure Development*, 35, 1, pp. 41-47, Business Source Complete, EBSCO*host* [Accessed: 24 July 2014]

16. Ciarniene, R. & Vienazindiene, M. (2013) **'Lean Manufacturing Implementation And Progress Measurement'**, *Economics & Management*, 18, 2, pp. 366-373, Business Source Complete, EBSCO*host* [Accessed: 21 June 2014]

17. DeMarco, T. (2013), **'Bells, Whistles, Power, and the Requirements Process'**, *IEEE Software*, 30, 4, p. 104, Business Source Complete, EBSCO*host* [Accessed: 16 July 2014]

18. Dominici, G., and Palumbo, F. (2013) **'Decoding the Japanese Lean Production System According to a Viable Systems Perspective'**, *Systemic Practice & Action Research*, 26, 2, pp. 153-171, Business Source Complete, EBSCO*host* [Accessed: 15 June 2014]

19. Duncan, E., and Ritter, R. (2014), **'Next frontiers for lean'**, *Mckinsey Quarterly*, 2, pp. 82-89, Business Source Complete, EBSCO*host* [Accessed: 23 July 2014]

20. Ebert, C. (2014) **'Software Product Management'**, *IEEE Software*, 31, 3, pp. 21-24, Business Source Complete, EBSCO*host* [Accessed: 17 July 2014]

21. Elzamly, A., and Hussin, B. (2014) **'AN ENHANCEMENT OF FRAMEWORK SOFTWARE RISK MANAGEMENT METHODOLOGY FOR SUCCESSFUL SOFTWARE DEVELOPMENT'**, *Journal Of Theoretical & Applied Information Technology*, 62, 2, pp. 410-423, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 10 July 2014]

22. Fernando Capretz, L. (2014) **'Bringing the Human Factor to Software Engineering'**, *IEEE Software*, 31, 2, p. 104, Business Source Complete, EBSCO*host* [Accessed: 10 July 2014]

23. Ferrucci, F., Harman, M., Ren, J., and Sarro, F. (2013), **'Not Going to Take This Anymore: Multi-objective Overtime Planning for Software Engineering Projects'**, *ICSE: International Conference On Software Engineering*, pp. 462-471, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 11 July 2014]

24. Filho, M., and Uzsoy, R. (2013) **'The impact of simultaneous continuous improvement in setup time and repair time on manufacturing cycle times under uncertain conditions'**, *International Journal Of Production Research*, 51, 2, pp. 447-464, Business Source Complete, EBSCO*host* [Accessed: 29 June 2014]

25. Ghazi, P., Moreno, A., and Peters, L. (2014) **'Looking for the Holy Grail of Software Development'**, *IEEE Software*, 31, 1, p. 96, Business Source Complete, EBSCO*host* [Accessed: 20 June 2014]

26. Grensing-Pophal, L. (2014) **'DIGITAL ADVERTISING TRENDS YOU NEED TO KNOW'**, *Econtent*, 37, 5, pp. 14-19, Business Source Complete, EBSCO*host* [Accessed: 2 August 2014]

27. Goddard, J. (2009) 'A Brief History of Management Theory' *Rethinking Management and Employee Engagement* pp.1-13

28. GUPTA, S. (2013), **'Turn planning art into planning science'**, *Industrial Management*, 55, 5, pp. 16-19, Business Source Complete, EBSCO*host* [Accessed: 17 July 2014]

29. Heerkens, G.R. (2014) **'THE COST OF CHANGE'**, *PM Network*, 28, 1, p. 22, Business Source Complete, EBSCO*host* [Accessed: 24 July 2014]

30. Hijazi, H., Alqrainy, S., Muaidi, H., and Khdour, T. (2014) **'Identifying Causality Relation between Software Projects Risk Factors'**, *International Journal Of Software Engineering & Its Applications*, 8, 2, pp. 51-58, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 16 August 2014]

31. Holzmann, V. and Panizel, I. (2013) **'Communications Management in Scrum Projects'**, *Proceedings Of The European Conference On Information Management & Evaluation*, pp. 67-74, Business Source Complete, EBSCO*host* [Accessed: 29 July 2014]

32. Jackson, M. (2012), **'STEP BY STEP'**, *PM Network*, 26, 6, pp. 56-61, Business Source Complete, EBSCO*host* [Accessed: 29 June 2014]

33. Kelly Services India (2014) **'2013 Salary Guide'**. (India) Available at: http://kellyservices.co.in/uploadedFiles/India_-_Kelly_Services/Documents/India%20salary%202013%2014.pdf [Accessed: 20 June 2014]

34. Kerzner, H. (2013) *Project Management*, 11th Edn, New Jersey: Wiley

35. Keyes, J. (2013) **'The Need for Lean Training', *Journal Of Management Policy & Practice*,** 14, 3, pp. 78-83, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

36. Koury, F. (2014), **'The search for a better way'**, *Smart Business Chicago*, 11, 4, p. 4, Business Source Complete, EBSCO*host* [Accessed: 21 July 2014]

37. Liebisch, W. and Gruhs, R. (2012) **'Lean Training Linking theory and practice in Siemens'**, *Global Focus: The EFMD Business Magazine*, 6, pp. 7-10, Business Source Complete, EBSCO*host* [Accessed: 23 July 2014]

38. Liker, J. (2004) *The Toyota Way: 14 management principles from the world's greatest manufacturer,* New York: McGraw-Hill

39. Maurer, R (2013) **'Will Lean Work in Your Organization?'** *Journal For Quality & Participation*, 36, 2, pp. 27-31, Business Source Complete, EBSCO*host* [Accessed: 23 July 2014]

40. Onetto, M. (2014) **'When Toyota met e-commerce: Lean at Amazon'**, *Mckinsey Quarterly*, 2, pp. 90-96, Business Source Complete, EBSCO*host* [Accessed: 2 August 2014]

41. Packeer Mohamed, S., Baharom, F., and Deraman, A. (2014) **'An Exploratory Study on Agile based Software Development Practices'**, *International Journal Of Software Engineering & Its Applications*, 8, 5, pp. 85-113, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 16 August 2014]

42. Par, N. (2014), **'Leadership and Adaptability Take Center Stage (cover story)'**, *PM Network*, pp. 1-11, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

43. Partida, B. (2014) **'Adopting Lean Processes in Procurement Can Lead to Efficiency and Lower Costs'**, *Supply Chain Management Review*, 18, 4, pp. 78-80, Business Source Complete, EBSCO*host* [Accessed: 16 August 2014]

44. Preimesberger, C. (2014), **'10 Essential Attributes of the Latest IT Management Tools'**, *Eweek*, p. 1, Business Source Complete, EBSCO*host* [Accessed: 2 August 2014]

45. Reece, P. and Antosiak, L. (2014) **'Outsourcing & Lean Manufacturing: Product Quality & Waste Elimination'**, *Nutraceuticals World*, 17, 2, pp. S-18-S-21, Business Source Complete, EBSCO*host* [Accessed: 29 June 2014]

46. Robbes, R., Vidal, R., and Bastarrica, M. (2013), **'Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft'**, *IEEE Software*, 30, 5, pp. 46-53, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

47. Robert Walters (2014) '**Salary Survey'**. (China) Available at: http://www.robertwalters.cn/wwwmedialibrary/WWW2/country/china/content/salary-survey/2014%20Robert%20Walters%20Salary%20Survey%20-%20China%20%28EN%29.pdf [Accessed: 22 July 2014]

48. Russell, M.T. (2013) '**Marketing IT to BI Users In-House: The Importance of Small Talk',** *Business Intelligence Journal*, 18, 4, pp. 19-21, Business Source Complete, EBSCO*host* [Accessed: 28 June 2014]

49. Saunders, M., Lewis, P. and Thornhill, A. (2012) *Research Methods for Business Students,* 6th Edn, Essex: Pearson Education Limited

50. Schwalbe, K (2014) *Information Technology Project Management,* 7th Edn, Boston: Course Technology

51. Sheng, V., Gu, B., Fang, W., and Wu, J. (2014) **'Cost-sensitive learning for defect escalation'**, *Knowledge-Based Systems*, 66, pp. 146-155, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 16 August 2014]

52. Smith, S. (2013), 'Revolutionizing support services with lean', *Industrial Engineer: IE*, 45, 7, pp. 42-46, Business Source Complete, EBSCO*host* [Accessed: 24 July 2014]

53. Stoica, M., Mircea, M., and Ghilic-Micu, B. (2013) **'Software Development: Agile vs. Traditional'**, *Informatica Economica*, 17, 4, pp. 64-76, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

54. Sobral, M., Sousa Jabbour, A., and Chiappetta Jabbour, C. (2013), **'Green Benefits From Adopting Lean Manufacturing: A Case Study From the Automotive Sector',** *Environmental Quality Management*, 22, 3, pp. 65-72, Business Source Complete, EBSCO*host* [Accessed: 2 August 2014]

55. Souza, I., da Silva Gomes, G., da Mota Silveira Neto, P., do Carmo Machado, I., de Almeida, E., and de Lemos Meira, S. (2013) **'Evidence of software inspection on feature specification for software product lines'**, *Journal Of Systems & Software*, 86, 5, pp. 1172-1190, Computers & Applied Sciences Complete, EBSCO*host* [Accessed: 29 June 2014]

56. Sunder, V. (2013) **'Synergies of Lean Six Sigma'**, *IUP Journal Of Operations Management*, 12, 1, pp. 21-31, Business Source Complete, EBSCO*host* [Accessed: 16 June 2014]

57. Swartling, D., and Poksinska, B. (2013) **'Management Initiation of Continuous Improvement from a Motivational Perspective'**, *Journal Of Applied Economics & Business Research*, 3, 2, pp. 81-94, Business Source Complete, EBSCO*host* [Accessed: 22 July 2014]

58. Teschler, L. (2014) **'Why big companies can't innovate**', *Machine Design*, 13 February, Business Source Complete, EBSCO*host* [Accessed: 24 July 2014]

Appendix

Appendix

1. Primary Data Collection Questions

1.1 Case Study Questions

A) Length of release cycle
Increased by 0-10%, 11-20%, 21-30%, 31-40% 50%+
Decreased by 0-10%, 11-20%, 21-30%, 31-40%, 50%+

B) Team size
Increased by 0-10%, 11-20%, 21-30%, 31-40% 50%+
Decreased by 0-10%, 11-20%, 21-30%, 31-40% 50%+

C) Cost
Increased by 0-10%, 11-20%, 21-30%, 31-40%, 50%+
Decreased by 0-10%, 11-20%, 21-30%, 31-40%, 50%+

1.2 Survey Questions

A) Domain: (Free Text)

B) Team size: 4-6, 7-9, 10-12, 12+

Lean/Agile:

D) Length of an iteration
1 week
2 weeks
3 weeks
4 weeks
More than 4 weeks

E) Duration of a release cycle
Less than 4 iterations
4-5
6-7
7-8
9-10
10+

Waterfall:
F) Duration of a release cycle: (Free Text)

G) When are the maximum defects detected?
While defining requirements and designing the feature
While developing
Early stages of testing
During system testing

H) How much of the total system in production is used regularly by every user?
81%+
61-80%
41-60%
Less than 40%

I) How much of the total system in production is never used by any user?
Less than 20%
21-40%
41-60%
More than 61% of the system

J) What percentage of requirements is changed after a feature has been developed?
0-10%
11-30%
31-50%
51+%

2. Companies Participating in the Survey

Some of the companies allowed their names to be used in the study. The links to the company websites have been given below:

http://www.ibm.com
http://www.version1.com/Home
http://www.sap.com/index.html
http://www.extentia.com/
http://www.spiderlogic.com/

3. Calculation of Standard Deviation for team sizes

Data	Equal Class Intervals	Frequency (f)	Mid Point (x)	(fx)
4-6	3.5-6.5	9	5	45
7-9	6.5-9.5	13	8	104
10-12	9.5-12.5	9	11	99
12+	12.5+	7	14 (assumed mid point)	98
		∑(f)= 38		∑(fx)= 346

Formula used for Arithmetic Mean

$$\bar{x} = \frac{\sum(f\,x^1)}{\sum(f)}$$

Mean= 9

Formula used for Standard Deviation

$$\sigma = \sqrt{\frac{\sum f\left(x^1 - \bar{x}\right)^2}{\sum(f)}}$$

Standard Deviation= 1.1