EFFECTIVENESS OF AGILE DEVELOPMENT FRAMEWORKS WITH RESPECT TO TESTING

SUBMITTED BY:

MUHAMMAD AHMAD NAWAZ UL GHANI 15007114021

SUPERVISED BY:

DR. MUHAMMAD SHOAIB FAROOQ

CO-SUPERVISED BY:

MR. AMJAD HUSSAIN ZAHID

SCHOOL OF SYSTEMS AND TECHNOLOGY
UNIVERSITY OF MANAGEMENT AND TECHNOLOGY
2015-2017

Thesis Report

EFFECTIVENESS OF AGILE DEVELOPMENT FRAMEWORKS WITH RESPECT TO TESTING

Submitted To:

School of Systems and Technology

In Partial Fulfilment of the Requirements

For the Degree of

MASTER OF SCIENCE (SOFTWARE ENGINEERING)

Submitted By:

MUHAMMAD AHMAD NAWAZ UL GHANI 15007114021

Session 2015-2017

Supervised By:

Dr. Muhammad Shoaib Farooq

Co-Supervised By:

Mr. Amjad Hussain Zahid


بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

FINAL APPROVAL

It is certified that the research work presented in this thesis entitled “Effectiveness of Agile development frameworks with respect to testing” was conducted by Muhammad Ahmad Nawaz ulGhani under the supervision of Dr. Muhammad ShoaibFarooq, and co-supervised by Mr.AmjadHussainZahid at the University of Management and Technology, Lahore, Pakistan in September 2017 for completing the requirement of the degree of the MS Software Engineering.

1. **Supervisor**
   Dr. Muhammad ShoaibFarooq _____
   Director Graduate Studies
   HEC approved PhD Supervisor,
   Associate Processor
   School of Systems and Technology,
   University of Management and Technology, Lahore

2. **Director Graduate Studies** _____
   School of Systems and Technology,
   University of Management and Technology, Lahore

DECLARATION

I, Muhammad Ahmad Nawaz ul Ghani, student ID# 15007114021, Session 2015-2017, here by certify that this thesis is being submitted in partial fulfillment of the requirement for the MS degree in Software Engineering. This thesis is my original work and the data/ material presented here in has not been used for the acquisition of any other degree from any institution.

Muhammad Ahmad Nawaz ul Ghani

Signature: _____

Date: _____

DEDICATION

With due respect, gratification & sincere devotion

I dedicate this research to my

Parents, Teachers & Friends

Whose support always strengthened

the way of my success to

'love and live without status'*

**Muhammad Ahmad Nawaz ul Ghani*

ACKNOWLEGEMENT

Alhamdulillah! With the grace of Almighty Allah (SWT) and the blessings of Prophet Muhammad (PBUH), I've been able to complete this research. The work accomplished is a feeling of joy and contentment. However, I want to relate this open door on account of all who have straightforwardly or in an unforeseen way helped me to perform this task.

Foremost, I would like to express my sincere gratitude to my Supervisor, HEC approved PhD supervisor, Director Graduate Studies, Dr. Muhammad Shoaib Farooq and co-supervisor Mr. Amjad Husain Zahid for their continuous support in my MSSE's related research, and for their patience, motivation, and immense knowledge. Their guidance helped me in all the research work and writing this thesis. I could not have imagined having better supervisors for my MS study.

In addition to my advisors, I would like to express my gratitude towards the Dean, School of Systems and Technology, Dr. Shaukat Iqbal and every individual who gave their precious time and constructive criticism to complete this undertaking. Likewise, I want to thank University of Management and Technology for supporting the students with facilities especially library and IPC where I availed countless opportunities of success.

Particularly, I'm very thankful to the Chairman BOG ILM Trust, University of Management and Technology, Dr. Hassan Suhaib Murad who always took the initiative to support the students and open new horizons for the researchers.

Saith Abdul Ghani & Sons (my family), especially my mother Suraiya Begum and father Saith Muhammad Latif, whose endeavors provided me a roadmap towards the success. Their uncountable sacrifices and devotions can never be expressed in words.

It would certainly be pertinent to mention the support and efforts of my patrons and fellows; especially, Iqbal Qaisar, Nimra Najeeb Gull, Rana Iftikhar Ahmad, Shiraz A. Siddique, Babar Bilal Rao, Syed Zeeshan Hussain Shah Gellani, Shaheer Malik, Aisha Naeem, Mehak Shahzad, Amin Anjum, Arslan, Hafsa Mahboob, Ambreen Liaqat and Syed Muhamad Waji Haider. I would always be indebted to their love.

O' Okara!!! I am in love with thy nostalgia__ the sparkling thoughts that swing in my mind; create fair luminous clouds of rumination.

As much as this research work is easy to understand, it would be hard to forget…

PREFACE

This thesis was prepared as a partial fulfillment of the requirements of acquiring the degree Master of Science in Software Engineering (MSSE), in the School of System and Technology, SST, located at the University of Management and Technology UMT.

The thesis consists of six chapters. Chapter 1 explains the introduction of the thesis related to the effectiveness of agile development frameworks especially dynamic systems development and lean software development. Chapter 2 provides a review of the literature, raising and discussing key concepts to lay the conceptual and theoretical foundation for this study. Chapter 3 presents Research methodology in which the steps are explained of the research conducted. Chapter 4 describes the Information Collection, chapter 5 discusses, the Results and Analysis and Chapter 6 explain the conclusion and Future Work. At the end, the references of all the research work are presented followed by research questionnaire in Appendix.

ABSTRACT

Software Development Life Cycle models are used as the basis to design software applications. In these models, there exists some tribulations and to overcome these defects, agile models are presented. The mature software applications of agile development utilize both iterative and incremental style. It can be viewed as a response against conventional procedural activity. It is viewed not as to meet the volatility and varying situations but as the modern business ethics for software development. By examining and exploring the agile models, it has been found that efficiency of agile models can be improved. This research investigates the agile frameworks (DSDM and LSD) in terms of testing. It also examines that the effectiveness of dynamic systems development model (DSDM) from testing perspective the results show the performance of DSDM agile framework and the estimated efficiency of this framework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

CHAPTER 1

1. Introduction

Software Engineering is the process to analyze clients' requirements, designing the required structures and testing the end user applications. It also includes creating standards for software development. It provides the basic framework for handling larger and complex systems that are both reliable and productive. Software Engineering is being utilized in every field of life, from nanotechnology to large spaceships, it plays a pivotal role. It deals with complex mathematics and large data structures. A software engineer considers needs of end user then plans and creates new applications.

Software Engineering also includes a method of breaking down existing software design and altering it to meet current end user needs. Software development has made life easier. Difficult tasks are now done by just a click. It has made everything systemized and easily accessible that develops the interest of the user in the desired activity. This is because every activity in software development is itself organized and systematic.

1.1 Challenges in Software Development

There are bunch of difficulties which still need to be addressed in the field of Software Engineering. It's difficult to accept the robotizing of everything which will prompt a superior world. As software applications are now supercharging our efficiency, applications are less demanding. Software Engineering is the orderly way to deal with the advancement, operation, support, and accomplishment of software. Software Engineering utilizes an all-around characterized and systematic way to deal with creative software development. This approach is thought to be the best method for delivering great applications. In spite of this elderly approach in software development, there are still some genuine difficulties being faced in Software Engineering. Some of these difficulties are recorded underneath [2].

- One of the key issues in software engineering is changing the cost structure; this requires a completely different performing art. Nowadays, the development strategy for small cadre is not always limited to large structures.
- The strategic framework should be used to develop large corporate software, using improved project management and the project combines all the generosity. It believed on that management of personnel, systems and equipment with associated technology

and rare exercise instruction in a small attempt can also be used to improve software engineering.

- Whatever it is, you need to be far more formal for large companies. Whatever it was when the scale was changed in law to a wide range of systems to solve such problems. It also needs to compare and improve the formal approach to promote the government in order to move in both directions, it is important to make the project official.
- The cost of the system is to use the structure of the asset because of the software, hardware and other cost recovery assets of software development focus on work.
- The cost of the project, measured on individual month basis is considered to be the cumulative number of individual months. Schedule essential calculator is a lot of problems. The possibility of selling the business model, i.e. the reduced object, is directed to the fact; it should be a small overhang length from conception to transfer.
- Any business with such a fundamental will in like ways requires that the strategy traverse for building an application required by the business must be small. One of the primary issues driving any creation is Quality. A method can be seen having three estimations i.e. Application Operation, Application Transition and Application Revision.
- The unimportant exertion and small process term are the central concentrations of any project. For a relationship, there is another objective which is consistency. An association required in software change does not simply require immaterial effort and high bent.
- Any business with such an essential will in like manner require that the procedure length for building an application required by the business must be small.
- The negligible exertion and small process length are the fundamentals of any project. For a relationship, there is another target which is consistency. An affiliation required in software improvement does not just need insignificant exertion and high bore for a project; it may need these negligible exertions and high bore dependably [4].
- The strategies used to grow small or medium-scale undertakings are not appropriate with regards to the improvement of extensive scale or complex systems.
- Changes in software development are unavoidable. Today, changes happen quickly and obliging these progressions to create quality software which is one of the real difficulties confronted by the application engineers.
- The progression in computer and software technology is required for the adjustments in nature of software systems. The application systems which are different from each

other are not for much utilization. Consequently, one of the difficulties of Software Engineering is to create outstanding software, adjusting to the changing needs inside worthy timetables. To address this difficulty, the question situated approach is favored, yet obliging changes to software and maintaining cost is a challenge [3].

- Informal correspondences take up a significant segment of the time spent on software projects. Such wastage of time defers the completion of tasks in the predefined time.
- The client by and large has just an unclear thought regarding the extension and prerequisites of the application system. This generally brings about the improvement of software, which does not meet the client's necessities.
- Changes are normally joined in reports without taking care of any standard system. In this manner, confirmation of every single change regularly winds up plainly trouble.
- The improvement of high quality and dependable software requires the product to be altogether tied. In spite of the fact that intensive testing of software demolishes the greater part of assets, developers are bound to think about micro reason which can cause the system failure.

The previously mentioned key difficulties and the obligations of the system examiner, architects, and software engineers are not properly characterized. Additionally, if the client necessities are not definitely characterized, software engineers can misunderstand the significance of the critical project. Each of the difficulties should be addressed properly to guarantee that the application is created inside the predefined time and cost limitations. Also it must meet the necessities indicated by the client.

1.2 Software Development Lifecycle

In order to overcome problems and challenges at the beginning, the software development life cycle (SDLC) was introduced. Software Development Life Cycle (SDLC) is a description of the structure for assembling software applications from the beginning to the final stage of the program. In some strategic SDLC, a cost-effective, powerful and high assessment test is carried out [6]. The SDLC technique overall runs with the following stages: Requirements, Analysis, Design, Implementation, Testing, Deployment and Maintenance.

SDLC begins with the examination and definition steps, where the reason behind the application or framework must be established, the objectives of what it requires to achieve should be created and a strategy of positive fundamentals is prepared. The application is plotted and passed on, while trying to achieve a large portion of the requirements that were

progressed inside the previous stage. Next stage in the application movement lifecycle is the endeavoring stage [7].

It is an outstandingly arranged approach with clear definitions between one phase accompanying, and empowering to make a more sorted out arrangement with specific due dates and expectations. Likewise, it is a settled and attempted rationality that has been exhibited to work. The system has been around for quite a while, and if used appropriately, can be successful. The SDLC is straightforward and simple to appreciate with each one of the stages. SDLC particularly focuses on fulfilling the goals. With SDLC there is a high plausibility that targets will be refined. The goal of any SDLC is the deliverance of a quality thing to the end customer. With the execution of SDLC this goal is really based on the customers being ending up to great degree happiness.

1.3 Standard SDLC Models

Software development lifecycle models have been proposed like Waterfall, Iterative, Agile, etc. The software development life cycle provides the most regular element in the entire structure of the project, regardless of any stages. It's a kind of project plan, which uses various SDLC methods. The basic SDLC methodologies are discussed here: - [8].

1.3.1 Waterfall Model

Waterfall is the most arranged and the most composed SDLC demonstration. It completes in one stage, and then proceeds forward to the going without drawing their activities. Each stage depends upon data from the previous stage and has its own particular project course of action. Waterfall is clear and easy to coordinate regardless of the early rearrangements is possessed from the whole project course of events. Moreover, since there is no place for changes, once a phase is done; issues can't be settled until the point when you get to the help plan. This model doesn't work exceptionally if adaptability is required or if the project is entirely constant arrangement and arranged.

1.3.2 V-Shaped Model

Identifying the verification and validation, it appears that the V- shaped model is better than Waterfall and provides related testing stage for each sort of movement. Like Waterfall, each stage starts fundamentally after the previous one has ended. V-Shaped model is advantageous, there are no faulty requirements, and it’s hard to withdraw and make improvements.

1.3.3 Iterative Model

The essential advantage of iterative model is emphasis that it is represented, rather than beginning with completely known requirements. Execute a plan of software essentials at that point test, assess, pinpoint and advance requirements. Another adjustment of the thing is passed on with each stage. Wash and do again until the total framework is prepared. One favored perspective over other SDLC strategies is that: This model gives a working structure perfect on time and at the same time it makes it more sensible to acknowledge the changes. One boundary its assets can rapidly be eaten up by revising the system again and again.

1.3.4 Spiral Model

It is a champion model among the SDLC techniques. Spiral model gets incite from the Iterative model and its reiteration; the project is overlooked through four stages and converts it in a "spiral" until it is finished refining on all sides. Spiral model considers the working of a reiterated thing, and client input can be joined from at a reasonable time in the project.

1.3.5 Big Bang Model

A touch of multiplicity among SDLC methodologies, Big Bang Model occurs on behalf of no particular strategy, with no time spent on procedures. Most of the work advantage is blended in progression, the customers or consumers do not have a strong grip on what they want. Big Bang is one of the initial SDLC methods routinely utilized for small assignments with just a single or two software engineers. Enormous impact is not suggested for wide or composite attempts, as it's mostly used model. If the essentials are mystified at lead position, you could get to the end and understand the project that may be started from the earliest starting point once again.

1.4 Agile Software Development Model

By flouting the thing into cycles, the agile model rapidly passes on a working thing which is viewed as a phenomenally practical revolutionized enhancement. The model generates constant release, by means of each little and incremental change from the previous discharge [5]. Improvements made in all the cycles. This model underlines joint exertion, as the clients, planners and analyzers partake all through the project. Agile depends solidly on user’s correspondence.

Agile software development presents standards for software development under which requirements and arrangements are created through the group effort of self-dealing with cross-practical gatherings. It advocates flexible coordination, transformative progression,

early movement, and steady change. It stimulates quick and versatile response to change. These principles support the definition and continue with headway of various application change procedures. Agile Software Development is an umbrella term for a course of action of methodologies and is practiced in the light of the qualities and benchmarks [9]. The progress between self-dealing with cross-powerful gatherings and utilizing the appropriate practices through joint exertion is an exceptional situation.

Agile Methods have been around for more than ten years, while the supporting thoughts and by far most of the practices related with quick software improvement have been around for many years. There is still no addition to swift software improvement, however clearly agile procedures plans to answer a need to make software quickly, in a situation of rapidly developing requirements [10]. The usage of iterative progression is normal to each and every agile methodology and when in doubt there are visit releases to customers.

Each planned method is creative in its specific approach; they offer an ordinary vision and qualities. They persistently condemn that it gives a dynamically refine software application system [11]. They incorporate persistent organizer, relentless testing, steady compromise, and distinctive sorts of perpetual improvement of both the project and the application. They are lightweight, from standard waterfall-style shapes, and typically adaptable. The basic of light-footed systems is that they focus on connecting with people to collaborate and settle on decisions together quickly and suitably.

1.5 Research Questions

In this research try to answer the following questions in our research:

- What is the impact of requirement analysis in the LSD & DSDM agile frameworks?
- Which agile framework (LSD & DSDM) is productive and industrious with respect to testing?

1.6 Research Scope

This writing is characterized on research based estimation of agile development models. Their elements are determined for the general users and furthermore, the advanced state software industry is improved too. The project will give the interface through which users can enter their predefined necessities as per to build up their project so the framework will give the predetermined models. The interface is composed by the exploration and technique of testing on the framework and plans the criteria in context of that examination.

1.7 Research Objectives

The project we are examining is about agile development and which model is proficient according to user prerequisites as indicated. There are two stages, in order to examine which model is productive for testing approach among agile models. In the first stage to accomplish the target we need to develop a concise research through which we will discover the working of each project and in later stage, develop the standard/criteria through which we will assess the effectiveness of the technique.

The primary goals of this research are following:

- The examination of agile model, analyze it through various testing techniques and decide which model is effective.
- To build up the criteria, that the user will get the model agreeing to their required prerequisites.
- Through this information and research we will get the constraints on effectiveness of the agile models.

1.8 Problem Statement

In agile development, the products associations which are not been utilizing agile before faces few difficulties attempt to actualizing agile development process. Agile is based on time boxed development, where improvement life-cycles are called Sprints, which are short and bound in expansion. Agile sprints typically take two weeks to a month. Agile development of joining of new elements constantly checked time by time, which implies that code is checked regularly and re-gathered at the meantime. So in this way, software is always showing signs of change.

The Agile has been developed deliberately, on center of standards, strategies and offer numerous systems. Defects always exist when tasks are perceived wrongly, irrespective of user requirements. Required functionality does not work. There is certain need to compare the efficiency of different agile frameworks with respect to testing. Agile development and their models in terms of efficiency and testing is the main focus of this research. To make this examination intense and significant, this research focuses on two agile framework structures which are lean software development (LSD) and dynamic software development (DSDM), to assess the feasibility of agile advancement systems with difference of testing. Agile strategies are utilized diversely as indicated by the project's point of view. The primary point of this

exploration is to separate these systems, so that an effective, powerful and proficient structure can be selected which will work as per the client prerequisites.

Chapter 2
Literature Review

CHAPTER 2

2. Literature Review

A large number of organizations are realizing that their software development procedures must be in consistency with some type of direction. This is necessary in order to have simplicity in the development process as well as for the creation of specific software in short time, for example, automotive software, robotics, medical devices or financial management frameworks. Agile Software Development is the main strategy for the improvement in the software developing process which consequently improves the end product. A great deal of research has been done and the investigations are still going on. This research mainly focuses on an examination of agile models as indicated by their standards. The light-footed approach has turned into the principal consideration of the product businesses since that structure has been built up. Abrahamson has done research on the philosophy of the agile models and has compared them from project management point of view. This research has explained various models of agile with software development lifecycle, extending administration systems, and experimental standards [12].

The other notable research on agile strategies productivity was carried out by Shelly. Shelly has talked about the conceivable components and attributes, such as, emphasize on length, documentation, the span of the group and an approach such as iterative/incremental. Besides, much consideration is being given to methods for enhancing the effectiveness of organizations, for instance, by adopting lean standards. This brings up the issue that how one can adopt lean standards for software development inside a controlled domain. By examining and exploring the agile models, it has been found that efficiency of agile models can be improved with respect to testing. This research investigates the agile frameworks in terms of testing [13]. The agile has been developed on core rules, methods and offer many frameworks. To make this research acute and profound, this study discusses two agile frameworks (LSD & DSDM) to estimate the effectiveness of agile development frameworks with respect to testing.

Software has been part of current society for over fifty years. There are a number of software development procedures that are being used today. Few organizations have their own customized techniques for building up their software but the main issue revolves around two

types of methodologies: heavyweight and lightweight. Heavyweight techniques are the traditional approaches to create software, focus on advanced result, detailed documentation and extensive outlining of the design. The lightweight procedures, also called agile modeling, have gained significance among the software designing teams in the last few years. Unlike conventional strategies, agile techniques utilize short iterative cycles, and depend on implicit information instead of documentation [14].

In this research, the qualities of some traditional and agile methodologies that are generally applied as a part of software development are explained and compared. This research also examines the qualities and shortcomings of the two opposing techniques and tries to provide a solution of the difficulties related to executing light-footed procedures in the software application business. This topic is gaining much importance with respect to the viability of agile systems in specific situations. Agile approaches can provide great advantages to small and medium scaled projects but for substantial scaled projects traditional strategies appear to be useful.

Software development procedures are continuously advancing because of the everyday advancements in the technology and modern requests from clients. Nowadays, unique business atmosphere has offered ascend to new associations that seamlessly adjust their structures, procedures, and approaches to suit the new conditions. Such associations require data frameworks that are always ready to meet their evolving requirements, but the conventional strategy driven software development systems do not have this adaptability to progressively modify the development procedure.

Object-oriented methodology provides a reasonable technique to gradually develop the agile development procedures or large-scale new strategy. Various software development strategies, such as Extreme Programming (XP), feature-driven development, clear technology, Scrum and dynamic adaptive software development methods, fall into this category [15]. The goal of this research is to identify the challenges that software team must be counter in their attempts to hold the light-footed software development. While traditional techniques, for example, based on life cycle and object oriented methodologies keep on dominating the frameworks development field, various sentiment pieces and a few overviews unmistakably exhibit the developing popularity of agile development methodologies. The introduction of these strategies has partitioned the software development teams into restricting groups of traditionalists and agilest, each party broadcasting the predominance of

its own philosophy. A more adjusted perspective of the two challenging approaches is offered by a few who recommend that every technique has its qualities and additional constraints, and is appropriate for particular sorts of projects.

According to Boehm, organizations should precisely develop the best mix of agile and plan-driven techniques that fits their circumstances. Most organizations do not disregard the agile procedures; however for organizations saturated with the traditional frameworks development techniques, practically during the agile software development process there will be a few challenges, since the two software development approaches are grounded in contradicting ideas. Previous studies have shown that changes in the software development process and complex changes in miracle systems differ only in the use of new tools and techniques to replace existing tools and techniques to improve. This change can affect different parts of the organization, including its structure, culture and management practices. Therefore, understanding the phenomenon of the evolution is related to the basic initial steps of coping up with these changes [16].

Software community has intentionally taken an authoritative and administrative viewpoint of this change phenomenon fundamentally on the grounds that such point of view, although critical in actualizing hierarchical change, is a great extent missing from the present dialogue on appropriation of agile techniques. This research gives a concise correlation of agile development procedures with conventional systematic development approaches and examines the difficulties of embracing agile development techniques. Software development has conceptual complexity which can be understood by running assignments and requirements that show a high level of fluctuation. Instabilities are additionally intensified by the differences and unpredictability of individuals who take part in such errands. The changing nature and development of instruments (for instance, development situation including software dialects, methods, et cetera) may likewise compound development issues. A legitimized, building based approach has commanded software development nearly since its commencement. Such an approach grounded in the standards of hard frameworks accepts that issues are completely specifiable, and that an ideal and unsurprising strategy exists for each issue.

Frameworks development in the conventional approach is guided by an existence cycle model, for example, the waterfall show, the spiral model, or a few varieties of these. Most associations cannot disregard the light-footed wave, but for those saturated with conventional

frameworks development, reception of agile procedures will probably represent few difficulties. Correspondence among project members is formalized through these archives. Clients assume an essential part amid determination development, but their cooperation is insignificant in different exercises. These techniques are proper both for protest arranged and object oriented developments [17].

Agile software development techniques are expert strategies for software development practices. They are made to think and think about the project, including age, community-based leadership, integration and code change and change the time fast enough to adjust the short-term uninterrupted iteration of the Loop being developed to present the framework to describe. A project is divided into sub-extensions; each element generally includes provisions for development, connection, testing and transport. Collective basic leadership including partners with different foundations and objectives is normal for agile development. Agile strategies support an authority and-joint effort style of administration where the project supervisor's part is that of a facilitator or organizer. The deliverable of every development cycle is working code that can be utilized by the client. Agile strategies are incapacitating of the traditional way of software development. Item information, in this manner, ends up noticeably implied. Turn of team enrolment guarantees this learning is not consumed by a couple of people. The iterative techniques portray that agile systems are best by object oriented innovations. The transformative conveyance show, proposed by Gilb gives a significant system to direct agile software development [18]. To cut the long story short, agile development is portrayed by social request in which broad agile effort and correspondence give the premise to aggregate activity. Assorted partners including engineers and end clients experience rehashed cycles of thought-activity reflection that cultivate a domain of learning and adjustment. Colleagues, enabled with more options and basic leadership forces, are not bound to a particular part. This expands the differing qualities/assortment of the teams and empowers them to self-compose and react with energetic willingness to eminent circumstances.

Agile software development Methodologies are completely composed of practices that have been made by professionals and scholars. For software applications development, this research examines the agile structures as far as testing. By analyzing and investigating the agile models, it has been found that proficiency of agile models can be enhanced by testing. In this research, two different approaches have been discussed for agile structures namely lean software development (LSD) and dynamic software development (DSDM), these

techniques can be seen as a cause or a response based on the traditional strategy, which emphasizes the on the defend based approach to ensure that the problem is completely specified, so each question is not surprising that there are ideals and policies. Traditionalists are considered to advocate a wide range of arrangements e.g., Systematic program after thorough reuse, to make productive and humble. By complexity, light-footed procedures address the test of a flighty world by depending on individuals and their imagination as opposed on procedures [19].

2.1.Lean Software Development

Lean software development focuses on the creation of high value-added products through waste disposal in the development cycle. Lean Software Development is an adaptation of the Lean Manufacturing principles and agile development model of software. Lean model focuses on customer feedback and reduces the waste and it is also useful to implement the principles of lean in the development of software-based environment. Lean Software Development promotes an incremental model which makes strong adaptability, faster than any other development process, creating more predictable quality products.

Agile programmers follow the statement which was written in 2001 [20]. "Agile Manifest" that developers need to pay attention to:

• Personal and interactive processes and tools

• complete documentation software work

• Driving customer collaboration through customer consultations

• react as expected

This statement is so interesting that some companies in the manufacturing sector, before the "Agile Manifesto" was designed followed a similar trend. In a software building environment, this development model is known as lean development. This is a lower cost model than the traditional development models because of faster production and customers' value-added products. Lean focuses on any reduction in waste production, and maximizing the value of the product. To reduce waste, the lean manufacturing model relies on the fast flow process to produce quality products, precisely supply the needs of customers at the time they want. To achieve these effects in the same software environment, the principles of lean should be followed in the software development process.

Historically, the software is using a cascading variation model to plot almost entire design of the system. First of all requirements are collected, and then the entire product development lifecycle is changing according customers mind. Software architect carefully plans and designs, and then discards system changes or problems, inappropriate to the previous plan. The root cause of these problems is a predictive model, where everything is predicted; assuming product according to the beginning of the program. However, opting for the feedback model these problems can be avoided [21]. The model focuses on the evolution of small increments; each increment quickly adds useful features requested by the end customer feedback. This is at the heart of Lean's additional software development. Lean Software Development Experts, Mary and Tom Poppendieck used it in the Toyota plant, and provided a solid foundation for agile programming. They created a set of lean software development principles. Many steps have been implemented in other programs, such as extreme programming [22].

The first principle of waste disposal is based on the elimination. The objective is to eliminate the worthless principle to create the final product of all parts of the product design. Waste can be divided into three types: waste code, project management and potential labor waste development. The first example of a code development loss is partly completed work. Partially completed work tends to become obsolete and unusable. Each time the code expires, the time spent previously is wasted. Another source of development is wasted when the generated fault is found and then corrected and re-checked to make a sustainable system. Both wastes are removed through an iterative boycott development system. In the iterative loop system, only once, we have developed a small part of the complete coding system, eliminating the need to complete the code part. Iteration of the loop also has the advantage of allowing the working version of the code to be monitored by regression test suite so you can correct the error immediately. After the error was detected less complexity of the errors was found in large systems.

Waste project management is often the most difficult to diagnose, especially in the case of additional processing [22]. These processes introduce additional documents that offer no real value to the final project, but because typical software is developed in this way so it is being employed." In order to prevent this wasteful struggle, each document must be considered necessary. However, developers also contributed to create problems in project management. Code or programming almost always results in the loss of knowledge transition, even complete documentation. Finally, the customers want to require their desire product but

unfortunately it’s not going to meet with the requirements. Then project manager deal between design, development and testing team to make it consistent. So it is a problem to allow developers work directly with the customer [23]. This is another part of the project management waste, more features that customers do not need. Studies have shown that more than 45% of all applications will never be used by customers [24]. To design or develop the customer requirements quickly the iterative model method is useful for that, the best way to ensure that developers only work to implement what the customer want, it makes the developer and customer relation frequent, and as a result product is better. Multitasking is generally considered more effective, but studies have shown that multitasking reduces a person's ability to concentrate on a single task. Lean development shows that a request focusing on a single task, developers will be able to put more of their efforts. Developers can fully focus on a task in short iterations on the other hand variety of assignments or heavy task for short-term incremental will not be result good.

Finally, the most extensive potential labor force develops information or instructions. Sometimes a developer who have not any task or assignment, he is in hibernate mood while wasting their time and their skills tend to be very expensive for the company. The best way to stop this waste is to let the developers make their own decisions and allow easy access to all the information needed by the developer. The iterative development model in the lean development is the key to remove most of the waste in the organization. Once these wastes have been removed, the following principles can help in value addition and improve the process of continuous development.

Expand learning or create knowledge, in any reasonably complex software project, it is impossible to complete the entire system at the beginning of the project. This can be done with traditional planning software designed to write code that drives the most authoritative system before the problems arise. Most of the system must be redrawn, when the problem is caused by an unexpected change in the system. Lean software development methods facilitate short iteration and full-cycle solution to this problem [25]. Each loop generates a function code that is tested and deployed in time according to the customer's requirements. This approach manages the changing needs and adapts to technical issues rather than the "complete" traditional approach. Sustainable development also relies on interacting with customers, solving problems, and providing information. By pooling customer needs and using customers to better understand problem domains, Lean developers can improve their

system knowledge and provide products that best meet with customer needs. This will reduce the cost of additional features that its customers do not care about. By participating in the decision-making process of the client, they can also begin to see what kind of products can be prepared.

Delayed commitment, Predictive model approach tends to produce erroneous assumptions. The evolving needs in software is so common, almost becoming expected of any project. This is a fundamental issue of fully specified articles, which is why rationalizing development will delay a decision until the last responsible moment. First, delaying the decision to allow developers to take full search problem, the developer may decide to use more knowledge to make decisions at the beginning of the project. Secondly, the late decision to keep options open, allowing developers to manage uncertainty even by creating undecided clients. Another option is to quit and start deleting dependencies in a way, the ultimate goal of the system can always add a function [26].

Powers conferred team; although the decision is very important, as far as possible, being defined as "low" is also very important. In other words, people should make the decision to do the work of man. In software engineering, usually the developers are writing the code. As long as developers quickly decide how to design and implement customer demand the iterative model. Instead, create templates and documents, models, and documents to precisely specify how each part of the project should be designed and coded, and the document should be closer to the criteria and goals of a particular part of the project. Low-level decision-making has several advantages. Developers' problems can make informed decisions for their work by allowing developers to make their own decisions based on those criteria and goals, the closest to perform. Due to Streamline's short development cycles, fast feedback, developers can quickly see the results of their decisions, and constantly improve the decision-making process [27]. This allows for a wider range of objectives of the project management process, rather than focusing on how to implement the project. However, for the low-level decision-making method, project developers must have experience in the field, or be able to quickly access the information needed to make informed decisions. Giving the team the ability to make skilled decisions no longer generates errors and waste, and ultimately eliminates the benefits of allowing low-level decision-making. As before, the interaction between customers and developers is the key to providing the most relevant information of the design issues most important to developers.

Quick delivery of software industry generally makes developers think as wrong code "hacker". However, developers have realized that lean delivery; should not mean the poor quality of the code [28]. When you use the feedback model to determine the application, the results of decisions is implemented in a clean event flow. At this point, due to the development of the welding process it has been optimized to allow time to apply to approve the deployment; the rapid delivery of high-quality products is quite feasible and expected [29]. Fast delivery is to meet all the requirements of customers. While the more developed model allows customers to change their minds mid-way through the project's demands, but fast delivery customers do not have time to change their minds. When they have requested and approved the product, it will release an increment in the next version, usually in about two weeks. If clients want to change their application, they can see clearly how their first application works and can make more knowledge about the upcoming changes. Fast delivery requires that each developer has the only amount of work that can sustain incremental version of management. If the job starts to build, everything that goes through the lean craft removes all the advantages of fast delivery [30].

As mentioned in the waste part, any part of the work done may become obsolete and at any time without any value. The application of fast delivery is the only way to consistently keep the requirements of the meeting without modification. In the construction of integrity, the completeness of the software includes two varieties: conceptual integrity and holistic concepts. With the perception of the integrity of the software it provides what customers need, even if they do not realize that they like it [31]. Perceptual completeness, through continuous interaction with customers or domain experts guides the project and adds maximum value to the application in the project. Incremental release plans also allow customers to see how their applications are implemented and to make enhanced request notifications. With the concept of integrity, software runs smoothly and functions well. Conceptual integrity is achieved through the continuous interaction between the project and its components, as well as with the customer's interactive developers, how they view the entire product. When developers talk about different parts of the program, they can plan their own considerations with the developer's rest to provide a coherent plan for personal decisions, although it will be designed by several developers [32].

While focusing on the short cycle of the feedback model, it can produce rapid and valuable products, it is still important to consider the whole project, especially about optimization. The short delivery cycle can lead to a very good part of the optimization, but through the chip

optimization chip does not guarantee that any product will provide the best and seamless experience. Developers must work together to produce consistent products that work well together in all parts of the world. Lean software develops small lifecycle increments and no longer worries the developers about the scope of the project. When the project scope is changed at any time during the project process, it soon becomes unnecessary. Lean focus, rather than meet the needs of customers, in every joint cycle of high-level needs understanding [33]. Considering the development based on these two principles, it becomes clear that the range will match what the customer really needs.

Lean software development model introduces incremental feedback, providing a stable development model for enterprises to build their own agile projects. By switching to a lean business model, the company benefits from a product-oriented customer that offers a high value at a very low cost. They also benefit from rubbish reduction in their organization because they are perfect in the development process. Modern software developers consider the Lean approach to seven principles. It is not needed to define all of them, but it must be said that it provides the product with high quality, delivered as soon as possible and respect its developer's professional skills is the main.

As mentioned above, the lean principle is a typical agile method. The development team is small and self-managed. Its members are interchangeable because each of them can perform multiple functions. In addition, the lean project has an iterative structure. This allows the developer to test the product after each iteration cycle. It is very important for lean projects because it focuses on quality. Most modern scholars believe that even in software development projects, it is best to use lean short-term agile methods. This is because the lean team is not great, and it is very effective. This means that they can be met in the short term. On the other hand, the lean project deals with the disposal of waste. Lean is the best way to save money from customers. However, you can also use other methods of software development. For example, if your software development project is very large and complex, it is best to use cascading methods. Cascading devices are large and do not require uninterrupted communication between their members. On the other hand, the lean team is small. This means that they need more time to write a code. If you want to participate in two or more teams in the lean project delivery process, you should consider how to coordinate your activities. This is not a simple question.

Given the point of the structure to depict a lean software development setting, this segment has an unmistakable position inside the system that's why it's necessary to make testing consistent [30]. At an abnormal state we may see lean and administrative consistent as conflicting, in any case we propose this is not really the situation. From an authoritative viewpoint lean recommends a mentality that ought to be received in all business related procedures including software development. In this sense the impact is seen by the way in which the association aides and backings the development staff. For instance, is basic leadership incorporated or is simply the strategy overseeing and self-sufficient development teams bolstered. What exactly degree is process change advanced? The seven standards of the Lean Philosophy segment expect to introduce a lean outlook in how individuals complete their day by day exercises.

2.2.Dynamic systems Development method

Methodology of the current professional work environment is a way to guide people, resources and processes defined and repetitively. The more research methods, the more obvious models become as common guidelines for developing new methods. DSDM is about people, not tools. It is about truly understanding business needs, providing the fastest possible delivery of work solutions and software. Dynamic Development Systems Method provides a control framework and best practices for rapid application development. It was created by a coalition of organizations, since its launch in January 1995, has proved very effective in the supply of maintainable systems, these systems are more in step with the traditional life cycle of production requirements than business needs . The Dynamic Systems Development Method approach reflects the basics of much project management knowledge. DSDM is rooted in the software developer's community, but the software development, through the development and development of the project development of a comprehensive process, DSDM structure is the basis of this problem has changed [34].

It can be implemented the DSDM framework for the agile and traditional development process. To illustrate how DSDM and agile methods, learn the principles associated DSDM critical value associated with agile development process. The following nine principles are essential for the implementation of any DSDM, ignoring a principle that would call into question the concept of the framework, and considerably increase the risk of the project [35]. In contrast, the DSDM project, some steps can be omitted or modified to suit the specific conditions of the configuration item.

Active user participation is a must, the first is considered to be most important, since the users effectively reduce the number of errors for cognitive participation in the project, thus resulting in the reduction of code errors. DSDM project in cooperation with a large number of users is not a guide, but must constantly use the user to select a group, rather than a regular review meetings or seminars. The continuity also needs some other DSDM principles, more important its single project property.

The team must have the right to make decisions, the need to avoid transaction costs due to project participants and managers to communicate the friction caused as soon as possible. Need to authorize modest resources or even simple need to change so that the project will slow down significantly. Attractive to users DSDM these inefficiencies have limited power to make decisions in the following areas:

• Current requirements

• What is the function required for a given increment?

• Functional requirements and priorities

• Detailed information on technical solutions

Focus on frequent delivery to ensure that fast delivery errors occur frequently, which is easy to change, and approaches the wrong cause. This also applies to program code, but it also applies to other files and data models that are required.

Business is deliverable standard of acceptable fitness; as its name suggests DSDM framework, the main effort is to provide good enough software to solve business needs and make improvements in subsequent iterations. As part of the natural life cycle of any software system, reconstruction, design and engineering improvements must be considered part of the project, not only after the completion of the project. DSDM recommends that you do not write special software, but meet the needs of subsequent business iterations, as well as recovery time and related operations. Even in the DSDM project, identifying key issues is also important to design the durability of these problems. Agile methods are effective, and the needed models and architectures have a good understanding, to a greater extent, depending on the way the traditional way of development, as they are likely to make a decision, which will lead to thirty or forty subsequent iterations.

Iterative and incremental development to create a complex control project, it must be broken down into functional packages; each version as long as one doesn't install a complete set of business functions to add new features. Under this principle, we must acknowledge the fact that the system software can be changed at any time. This principle is easy to enter, even at the beginning of the project; it can repeat the specifications and other results. Possible responses increase and more variable demand.

All changes in the development process should be reversible; for changing the system configuration it is necessary to increase the development process due to changes in the priority of the demand change. Modern software tools support this principle and require dynamic configuration of these projects. Inversion is often a matter of concern because the development process creates a valuable initial loss of the facts, but as DSDM provides a small step-by-step process, the total loss of work is very limited.

These requirements represent the high impact of the basic level; to limit the freedom of demand; it can be changed in the process of development and installed at some high level of demand. The grassroots should be interpreted as a "consistent" requirement for the process of commercial design.

The test is integrated throughout the life cycle; many methods require last-stage design or implementation aspects of development. DSDM must be tested at an early stage of the development process.

Cooperation is required; in order to avoid the separation and encouragement of enterprises in the DSDM project and technical staff, because the cooperation is the key project of DSDM project success. If there is no trust and a fair atmosphere, it would be difficult to collect the demand and then get honest feedback from the resulting product.

Meaning: the Agile Manifesto needs four times the value and 12 principles that are considered to be the beginning of a flexible development approach. To show how DSDM is a flexible idea, we will combine the DSDM principles of the system, which describes the value of the Agile Manifesto.

People: Many modern management practices emphasize the importance of people like some Agile Manifesto especially DSDM. The principles of DSDM 1, 2, and to a lesser extent nine prominent individuals are attending a key role of the project. Since the DSDM project should implement all nine principles of DSDM, it is implied that the individual relationship is still

being applied flexibly by the DSDM framework required for the tools and process ratings. Highly interactive, DSDM is required to be significantly less hidden in principles 5 and 3, which require the possibility of communication and coordination to reduce the friction of a healthy working environment.

Working software; the company's needs a leading position, of four principles of DSDM to control the success of the project and to measure the value of the company; technical aspects do not apply to the user. This is a controversial value that leads to a temporary solution that is often justified by such a statement. First of all, it is important to remember that the implementation of the project management method is limited when the agile value expresses a simple preference order. Secondly, the working software includes: work in any future environment. In order to facilitate the next bounce, it is reasonable that the developer does not contain new features to change the design. These decisions should be made in cooperation with the users involved in the project.

Cooperation; is the most important factor in the paradigm of change to react, whether it is a part of a project or personal life, cooperation, DSDM (nine principles) and flexible methods. Fair information, the number of changes should be adjusted value. No political obstacles talk; others explain the words and misleading mistakes. Unfortunately, cooperation is not easy, because people tend to make selfish decisions, especially if you expect to have an immediate sense of satisfaction. It can create a seamless collaborative environment - manage and lead the work.

Reply; react to change is anchored in the principles of six and seven principles. Management is changing business needs which are often regarded as the most agile and traditional difference method.

The DSDM project includes seven stages, the organization and integration of various functions and responsibilities, supported by a number of basic technologies. In DSDM project, there are several roles for classification, namely the role of the project, workshops and teams. Everyone has a role; the role of IT staff cannot tell whether a person is a programmer, designer or analyst, they are called developers and testers are an exception. If another group of people, internal users and consultants are involved in the project, the only part-time employees, they are called the user's ambassador.

In addition, a full-time team is the team leader, project manager and technical coordinator. In one project team, project managers can be represented by the same person, but in different projects with several teams, separate teams might be working on different tasks. The project manager is responsible for coordinating the work and is responsible for the specific working group for the technical responsibility of the person. Technical aspects of coordination such as construction and quality issues are the responsibility of the technical coordinator in a multi-team project.

In order to ensure adequate food and resources to support each DSDM project it should also include implementing perceptive users and sponsors as one of the part-time participants. Visionaries committed to motivating the team, project objectives and business objectives, managers must commit resources to the project and establish appropriate relationships with other project-related managers.

The DSDM project team usually consists of one or two components, the second team can be responsible for the development teams for testing the product. Studies have shown that the team of five experts is ideal in a team, and 6 persons team is highly recommended. If a team can provide more jobs, structure of a multi-team DSDM project size project of 150 people is made. Several projects have found that any single feature set of test and development teams are useful, with the reason and the project is the most important quality of each component. Because the test is a destructive job, it usually produces more efficient task by team [37].

The DSDM development process is divided into seven phases, some of which may be implemented in specific projects that can be omitted. Each stage has several key tasks, that can be changed to include more tasks (if any), and may require other methods for developing DSDM. In particular, DSDM does not provide the necessary technical tasks that allow DSDM in different situations and projects, because this can be achieved through the framework of each organization. These three steps are designed to be iterative, which means that they must be performed in each increment. DSDM assign specific results to each task and each stage. However, the specification of the results is sufficient for the general use of DSDM projects and commercial projects. Here the different phases of dynamic system development methods are discussed [38].

Pre-project: The project phase includes project proposals and options for candidate proposals. The project is pre-determined, whether a project must be carried at all.

Feasibility study: The general consideration in the feasibility study is to address the definition of the problem, possibly the cost, and provide a computer system to assess the technical feasibility of the business problem to be evaluated.

Trade research: The feasibility study is that DSDM is a suitable methodological framework that has been identified as an industry research that is the basis for all follow-up work. As the feasibility study, is short as possible (for several weeks, not months, calculated), it obtain sufficient insight and requirements.

Functional model of iteration ; Functional model iterative approach focuses on improving the commercial aspects of computer systems, that is, based on high-level business case identification processing and information needs. A given item can have several IMF weather type boxes.

Design and build iterations; Route design and construction in the computer system is enough to be designed for safe placement in the hands of the user. A given item can have multiple boxes of such time DBI.

Implementation: The implementation phase covers the development environment, the beginning of the operating environment.

After the project: Post-project tasks include performance metrics for deployed systems if further improvement is required. Often these measures are held by the technology of the project which is about six months later.

To identify important factors, management must start before the inspection and DSDM process, the practice is critical to the success of the project. DSDM Alliance has experienced the 10 most important factors for its members [39]:

• Before you start the DSDM concept acceptance

• The decision of the user and developer in the development team.

• Senior management with the aim of ensuring the constructive engagement of end users.

• Incremental delivery.

• Developers have easy access to end users.

• Team stability.

• Development team experience

• The size of the development team.

• Good business relations

• Development of technology

About this list it is interesting to note that many factors are even the responsibility of senior management. This fact shows DSDM requires quite advanced corporate culture as successful as long term. Some factors of success may not even apply to projects, namely "business relationship support" which usually does not exist in open source projects.

DSDM principles suggest that the current collection, synchronization always contains a number of best practices in one method. In 1994 the first version of DSDM intentions was published. Now, version 4.2 has been released. DSDM will always be the best practice framework, not in the other best practice framework; DSDM association is recognized by many changes in the project environment and framework to meet these requirements [40].

DSDM itself is a very dynamic and modular system. During development, DSDM designer is on the "limit case" of interest, not only the item "ingredients", meaning the most important thing is to apply a time limit to make choices. DSDM does not require the user to implement the entire project structure, but we must strictly adhere to nine principles, in addition to, any project manager who can implement a more or less agile development process, depending on the specific circumstances. Even in several DSDM environments it allows the use of other restraint methods, even welcome. DSDM is the basic idea: In order to cope with the first day of development, cannot survive it is this vibrant, and recommended to use some technical framework. Finally, DSDM with all the development issues is decided not to Downing. At the end of the product, all nine principles are not achievable then it may not be the best choice for DSDM.

The DSDM framework is the foundation on the basis of best practices for the direct implementation of the project structure, the advantage is simple, scalable and in the past it turned out, but it does not pretend to choose design solutions. DSDM authors issued a warning about its use once, which does not include the reason but, it provided more confidence, but about the drawbacks of another similar method. Switch to DSDM is neither too quick nor cheap. In any organization, it requires a significant cultural shift, because it

suddenly delivers the task which can become an alternative. This research describes that many people have accepted this fact: that their projects will be on time and in the budget, but it does not necessarily provide the required functionality. Initially, it will be difficult to accept. DSDM positioning itself is more mature and flexible and programming of the model. Choose several flexible methods and DSDM together that is SCRUM and DSDM, the Promotion Authority team. Crystal Method is considered a good collection DSDM Union after each revision, the version to control its architecture, in addition to fully demonstrate its further development [41].

British Airways, Deutsche Bank, HP, Renault and Los Angeles, are the long-term users of DSDM. Many other organizations use DSDM, but do not support DSDM public participation. Even in world's largest software maker, Microsoft has released a semi-agile solution for synchronization framework DSDM.

Agile method is very obvious, and this is another concept of offshore outsourcing is introduced. Xansa using DSDM is organizing their offshore development in India, where practices prove to be very profitable land-based and offshore development project using agile methodologies, sufficient to make the case business. Further research on agile methodologies and offshore outsourcing provides viable evidence for the development of IT and businesses should migrate, whether to support agile methods change.

Chapter 3
Research Methodology

CHAPTER 3

3. Research Methodology

It is necessary to follow the appropriate research method to confirm the strength and value of our work. The method of research is a scheme for solving systemic problems as a scientific research. In essence, we are going to be describing our work, explaining and predicting the phenomena according to specific problem statement. It also defines the gathered information about research. This chapter explains the tools that have been used to gather data and analyze it. First of all the previous work that has been done on Agile Software Development processes is analyzed critically and the gaps have been identified. Agile Software development techniques have been a topic of great interest among the software development companies. Both LEAN and DSDM are being used at the moment. Both of these have their own pros and cons. But DSDM seems to be ahead of Lean in some aspects. Both of these techniques are very much important in software development, but no comparative analysis has been done in the recent past. This research will help software development companies to get an idea about which technique will be feasible for their company. On the basis of information gathered through sources like research papers, conference papers and magazines, three hypotheses were developed. The hypotheses mainly dealt with examining which agile technique is better than the other.

The research strategy used in this research is Quantitative. Quantitative study is objective. This means that reality does not change in any situation. The target population consists of Software development companies. The participants of the research were software developers who have a hand on experience on both LEAN and DSDM Strategies. The research instrument used in this study is Questionnaire. The data gathered has been analyzed using SPSS. T tests were mainly applied to test the hypothesis.

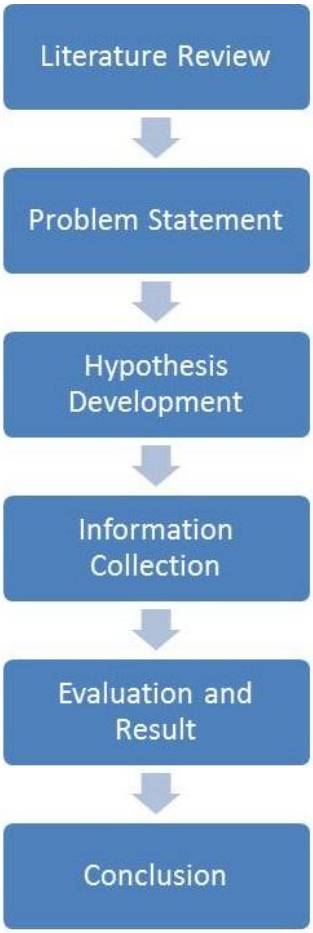

Figure 1: Methodology of research work

3.1. Literature review

By reviewing the previous work about agile, lean and DSDM; the current knowledge in the field summarized is very critically, moreover strengths and weaknesses of the previous work is also determined. In this research, literature review mainly includes a summary of the previous work of many authors. We provide the background of the specific problem statement; identify the problems, standards, principles, models, concepts and related research that are relevant to this area. We believe that a thorough and comprehensive literature review is very important. For this research, the main section that can be referenced or acknowledge, the fact that in throughout the research. This phase provides proper understanding with brief detail of ideas, thought implications, consequence and worth of published writings and important work to the readers. The task is not just to write a summary of the literature of previous research, but a significant aspect is to critically analyze the research, and then also present it in own perspective, as a scholar.

3.2. Problem Statement

A statement of the problem is a short outline of the problems that must be solved. In this research the statement of the problem describes the importance of the topic. In this phase; we have been explained right and wrongs, perfections and imperfections. The problem statement has brief overview of the issues or problems that exist in the chosen study. In the second stage, when all the work is studied and understood completely, a value for investigation of the problem statement is generated. A careful reading of the material is required. When readers read our research on setting a task, they know the importance of this research.

3.3. Hypothesis Development

In this research, hypotheses are used to support scientific research and create breakthroughs in knowledge. With the help of hypotheses the research is evaluated. The main goal of the hypothesis is the identification of targets, main points and the relationship between the problem and background work. The observed event have the measurement of results and conditions. It only depends on the mentally, maturity and scholastic approach, without the ability to measure, assume, analyze or guess else the hypothesis will be a failure.

Hypotheses are given below;

H1: DSDM is more efficient and effective agile development framework than LSD.

H2: Project development by using the testing technique of DSDM have higher success rate as compared to those of LSD.

H3: DSDM have a more scrutinized process of software development as compared to LEAN.

3.4. Information Collection: Questionnaire

For this research to collect information a questionnaire has been developed. The purpose of the questionnaire is to collect the accurate information, because if the data collection is successful, the comparison and analysis of the data is easy. Initially questionnaires were conducted on paper, but to get more accurate and essential responses divert attention on an e-mail and alongside electronic questionnaires were also published on social media platforms. For example, Facebook and Google form. In this phase the questionnaire for the information collection, we visit various software houses and meetings with experts were also conducted.

3.5. Results and Analysis

In this research results and analysis contains pure descriptive facts, not consequences. After data collection, we analyze and evaluate the research to obtain the results applying the statistics toll SPSS. The result is almost what appears as in the presence of specified problem statement. All the hypotheses are discussed, in detail and statistically analyzed to confirm this research. Individual estimates raw data unless they are a single instance or an illustrative sample. The aim is to collect the final data about this research. This will help readers to be familiar with the consequences of this research and the track.

3.6. Conclusion

At this point, with hypothesis analysis the results are being summarized. Conclusion of the collected data by using the statistical or analytical approach is being done. Furthermore, the implications of these results are evaluated in relation to the initial issue of research and interpretation. The ascension of this research is summarized in the conclusion. The result of this research is about the basic judgment and the ideas organized about the research.

Chapter 4
Information Collection

CHAPTER 4

4. Information Collection

In order to analyze or determine specifications of required scientific problematic statement, this research needs to gather information from different sources. For this research the information gained is to make accurate, complete, well descriptive and functional.

To make this research ceremonial, initially we start brainstorming to think how this research leads. Brainstorming is a common way to generate ideas, especially in a group environment. This usually arises at the start of a new project or discusses about any problematic issues, including all the numbers of stakeholder in order to capture first idea. Everyone will be invited to comment in this early stage without fear of the effects. Brainstorming helped to incorporate this research and gather wealth of ideas. I have also attempted to find other ways of idea and information and the possibility of more innovations. We think it is easy to be over information. On the basis of the developed hypothesis, it strengthened our sense of how we predict, prescribe and assume the solution of scholarly problems by giving an explanation and observation of the previous work. On the basis of hypothesis and assumptions for this research, we developed questionnaire which has been forwarded to number of organizations and software developers especially agile software developers.

In this research, all the data collection is based on questionnaire which is authenticated and frequent with the problematic specifications. The opinions of several people are collected via questionnaire to make this research consistent. Through this research I've analyzed that questionnaire and surveys are an effective way to gather data, to analyze and to understand views and opinions. To gather more effective views over the software industry, we designed questionnaire on Google form because it's easy and time saving. It's easy for professionals and experts to respond quickly. It is also interesting and attractive because it has sequence and simulates the questions. Google auto forms generate results automatically from the number of filled questionnaire which helps to save time in collecting information. For this research, our questionnaire has 30 questions to which 103 responses were on received Google form. The responses were satisfying because questionnaire about the research was quite understandable. The questions were according to our problem statement are relevant to the format and respondents. In this chapter, few questions are briefly described with the help of a

Pi-Chart diagram and also statistically find out their frequency through SPSS. The complete questionnaire has been attached at the end of this thesis report (Appendix A).

4.1 Agile software progress increases…

Close joint efforts in the supervision of project manager with clients is empowered with its necessities and acknowledgements. The participation of all the stakeholders made the whole system much magnificent, to increase the productivity and reliability of the system. Some professionals are of the view that it is not necessary for a successful project because in their observation it is all time wasting. Thus, in response to the question that in: "agile software progress increases due to the participation of the whole team in the plan with project manager". 70.9% responses are positive, but on the other side there are 29.1% professionals who do not agree with the statement.

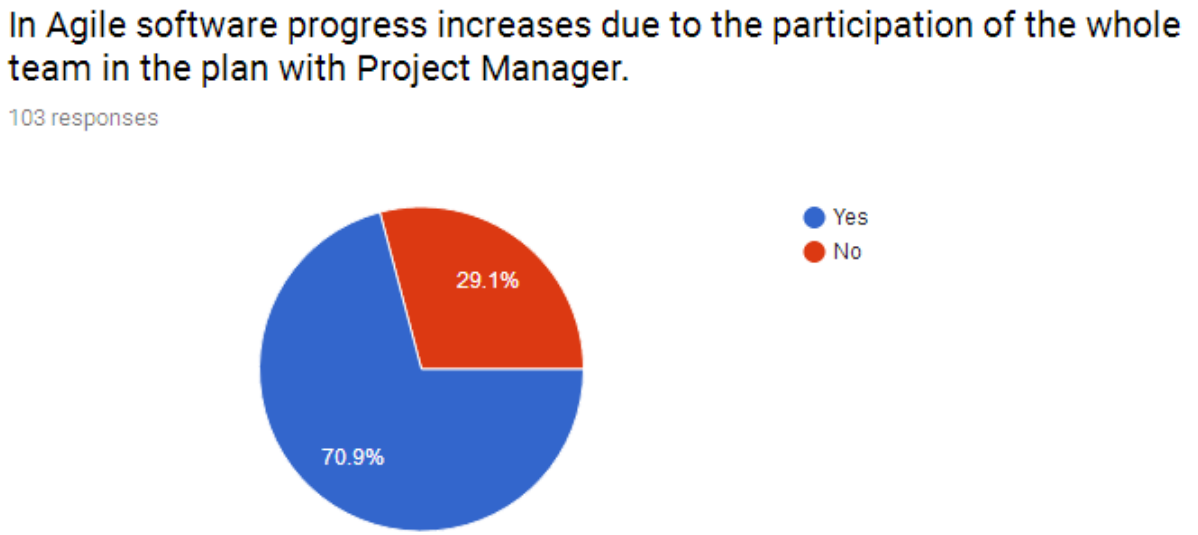

Figure 4.1 agile software progress increases…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	30	29.1	29.1	29.1
	Yes	73	70.9	70.9	100.0
	Total	103	100.0	100.0	

Table 4.1 SPSS result of the statement "agile software progress increases due to the participation of the whole team in the plan with project manager".

4.2. In agile, team collaboration is the main thing…

Agile software development portrays an arrangement of standards for software development under which prerequisites and arrangements are developed through the community oriented exertion of self-sorting out cross-practical groups. To gain more effective approaches it is necessary to discuss and communicate each and every aspect of the system and also talk about the expected results. That is why the statement "In agile, team collaboration is the main thing to increase the productivity" is prescribed in the questionnaire. Responses about this question is satisfying 69.9% experts agrees to it and 30.1% experts said No which means they are not agreeing to statement.

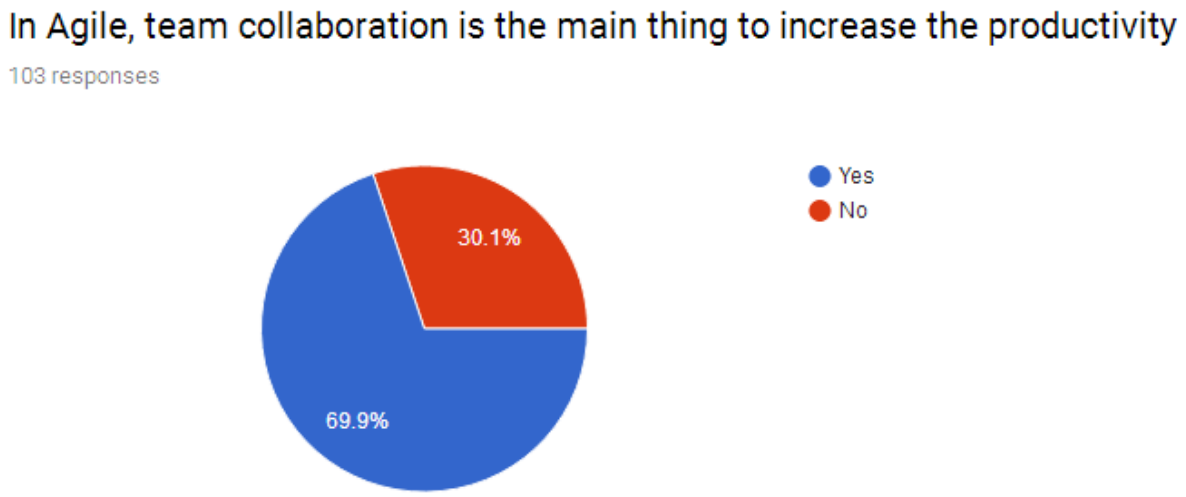

Figure 4.2 In agile, team collaboration is the main thing…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	31	30.1	30.1	30.1
	Yes	72	69.9	69.9	100.0
	Total	103	100.0	100.0	

Table 4.2 SPSS result of the statement "In agile, team collaboration is the main thing to increase the productivity".

4.3. Agile Methodologies help to improve…

Agile advocates versatile arranging, transformative advancement, early conveyance, and consistent change, and it energizes fast and adaptable reaction to change. These standards boost the definition and proceeding with advancement of numerous application improvement strategies. Agile Software Development is an umbrella term for an arrangement of strategies and practices in light of the qualities and standards. The main procedural activity in the lean is testing. Testing make the system proficient, efficient and professional. The question that "Agile methodologies help to improve the testing efficiency", this statement got a response of 69.9% experts agreeing to it and 30.1% mark No. Therefore, this opinion shows that the agile methodologies help to improve the efficiency.

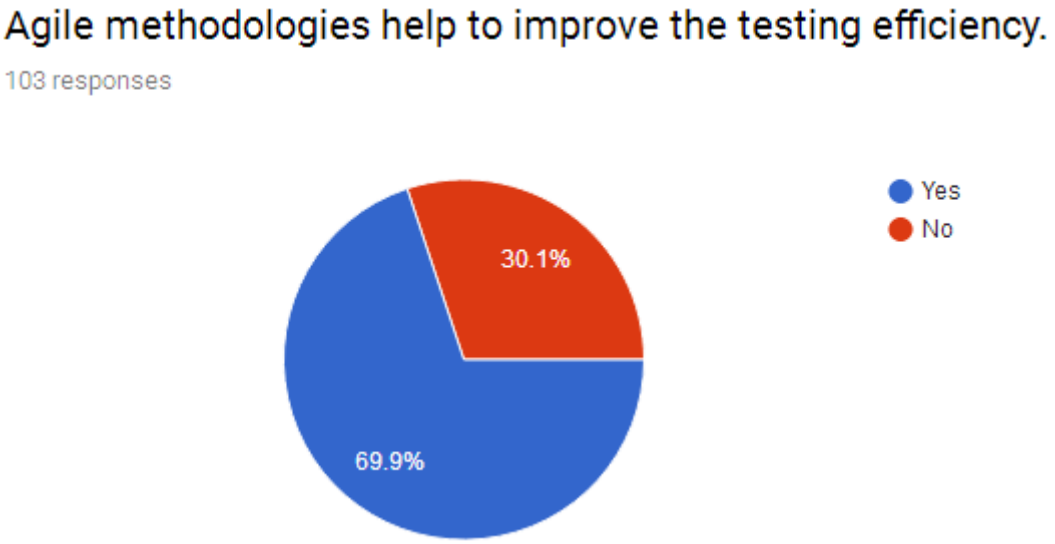

Figure 4.3 Agile Methodologies help to improve…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	31	30.1	30.1	30.1
	Yes	72	69.9	69.9	100.0
	Total	103	100.0	100.0	

Table 4.3 SPSS result of the statement "Agile methodologies help to improve the testing efficiency".

4.4. Agile process helps to build…

The agile process helps to build a product of professional quality which fits the business needs because agile software development Methodologies are completely composed of practices that have been made by professionals and scholars. For software applications development, this research examines the agile structures as far as testing. By analyzing and investigating the agile models, it has been found that proficiency of agile models can be enhanced testing. About that the question asked in the questionnaire is that "The agile process helps to build a product of professional quality which fits the business need". In response to the magnifying statement there are 72.8% experts who completely agree to it and on the other hand there are 27.2% professionals who do not agree and responded No.

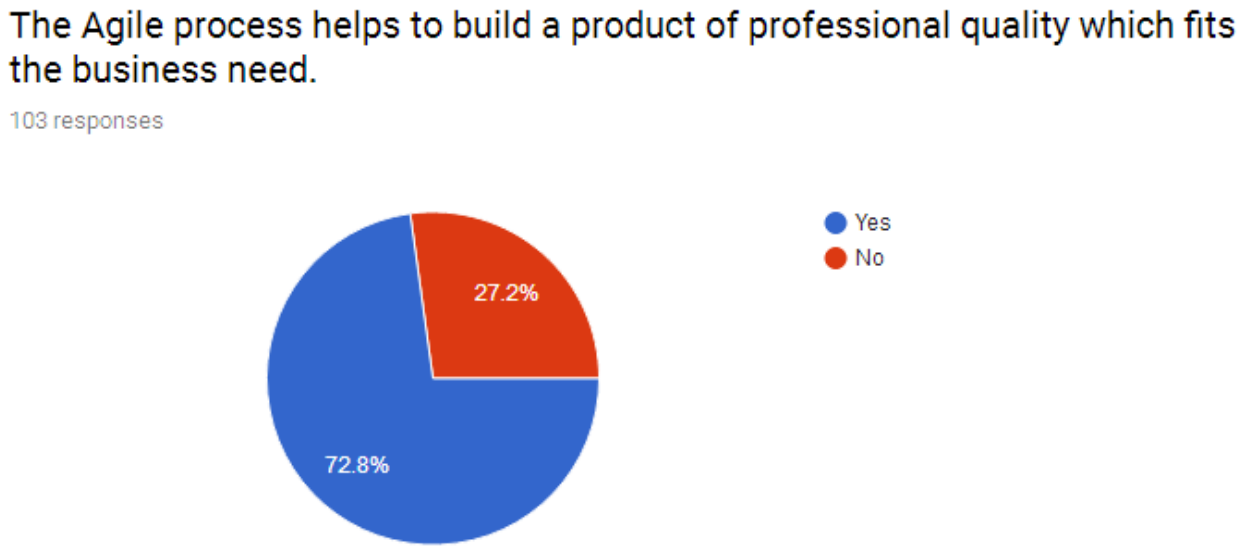

Figure 4.4 Agile process helps to build…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	28	27.2	27.2	27.2
	Yes	75	72.8	72.8	100.0
	Total	103	100.0	100.0	

Table 4.4 SPSS result of the statement "The agile process helps to build a product of professional quality which fits the business need".

4.5. DSDM planning, managing, executing and scaling…

DSDM provides a comprehensive framework for planning, managing, implementing and distributing flexible and iterative software development projects. As an extension of rapid application development, DSDM focuses on information systems that are characterized by compact schedules and budgets. DSDM solves the most common failures in information systems projects, including budgeting, lack of time and lack of user participation and senior management commitment. Here the discussion question in the questionnaire is about "DSDM methodology provides the comprehensive foundation for planning, managing, executing and scaling agile process and iterative software development". In response to this statement there are 80.6% experts in favor of DSDM and marked Yes and 19.4% experts do not agree with the statement.

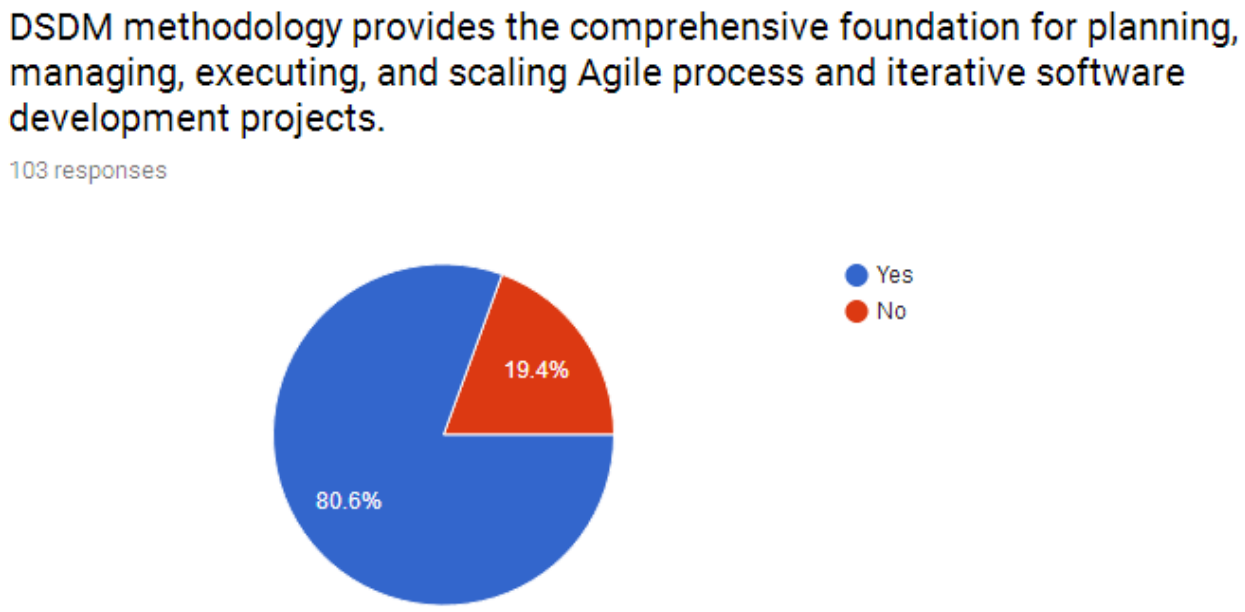

Figure 4.5 DSDM planning, managing, executing and scaling…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	20	19.4	19.4	19.4
	Yes	83	80.6	80.6	100.0
	Total	103	100.0	100.0	

Table 4.5 SPSS result of the statement "DSDM methodology provides the comprehensive foundation for planning, managing, executing and scaling agile process and iterative software development".

4.6. LSD emphasizes on the speed and efficiency…

Lean software development is a software development concept designed to simplify the production of applications and software products and improve its efficiency. The shorter the iteration, the better the team's learning and communication. With speed the decision can be postponed. Speed ensures satisfaction of the current needs of the client. This gives them the opportunity to postpone their thoughts until they find out what they really need. Customers pay attention to the fast delivery of quality products. The statement asked in the questionnaire is that "LSD emphasizes on the speed and the efficiency of software workflow and provides reliable feedback between programming and customer". The responses are much critical there are 50.5% experts on the side of the favor of this statement, on the other hand there are 49.5% experts are not in the favor of this statement.

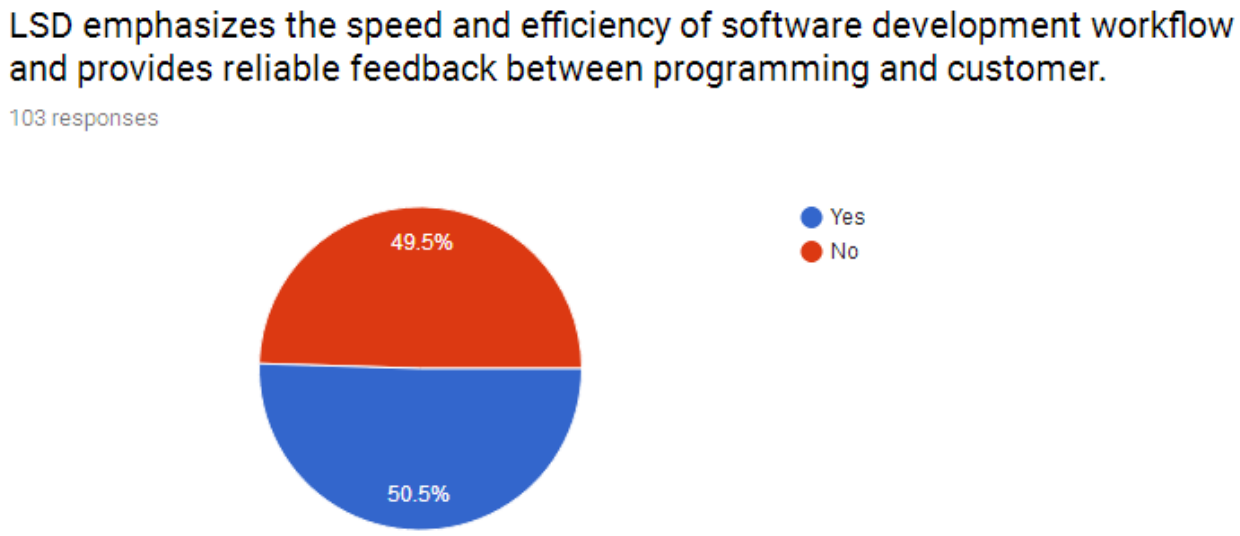

Figure 4.6 LSD emphasizes on the speed and efficiency…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	51	49.5	49.5	49.5
	Yes	52	50.5	50.5	100.0
	Total	103	100.0	100.0	

Table 4.6 SPSS result of the statement "LSD emphasizes on the speed and the efficiency of software workflow and provides reliable feedback between programming and customer".

4.7. LSD eliminates waste and prioritizes…

To eliminate waste, LSD must be able to recognize it. If some actions are bypassed the desired results cannot be achieved, which is waste. In the development process, the final rejection of a portion of the finished code is lost. Additional processes, such as office work and clients do not often use those functions that are lost. Switching between tasks is waste. Expecting other actions of the team, the process is wasted. Necessary exercise to complete the work is wasted. Defects and quality deterioration are lost. Management costs that do not generate real value are lost. According to that the statement open for opinion in the questionnaire is that "LSD methodology eliminates waste through such practices as selecting only the truly valuable feature for a system, and then prioritizing the selected features, and finally delivering them in small batches". In response of this statement 45.6% experts show their experience in favor and there are 54.4% experts are not agreeing with this statement.

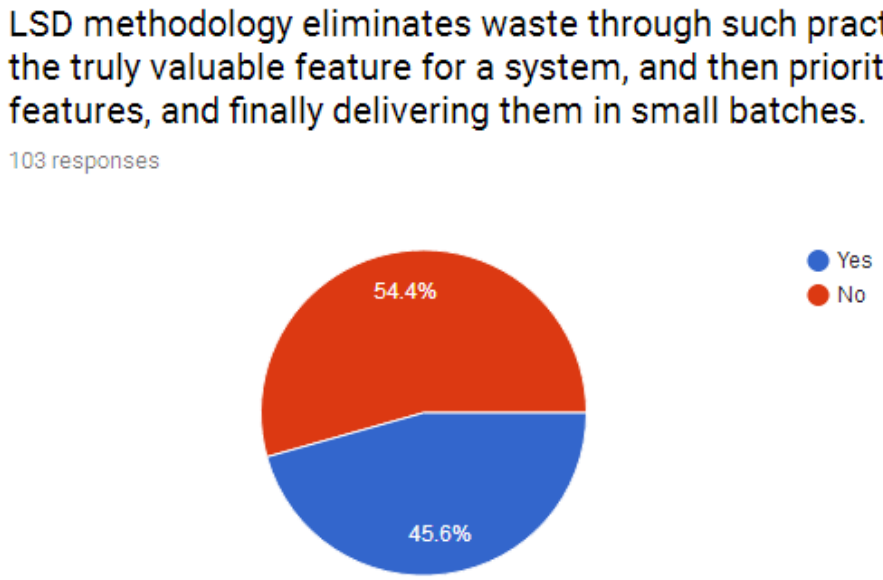

Figure 4.7 LSD eliminates waste and prioritizes…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	56	54.4	54.4	54.4
	Yes	47	45.6	45.6	100.0
	Total	103	100.0	100.0	

Table 4.7 SPSS result of the statement "LSD methodology eliminates waste through such practices as selecting only the truly valuable feature for a system, and then prioritizing the selected features, and finally delivering them in small batches".

4.8. Proper Management of Changing Requirements is …

Research shows that proper management of changing requirements is very necessary for system success. Whenever anyone is going to make changes in an already developed and functional system then it is really necessary for then to understand the whole systematic feasibility. The changes alter where it is necessary to modify or renovate the system to make it state of the art. The responding question which is put in the questionnaire is that "proper management of changing requirements is very necessary for system success". In response 34% experts strongly agree, 38.8% agree with the statement, on the other hand 13.6% people strongly disagree with proper management claiming that it is not necessary. There are few people who disagree with the statement or are neutral.

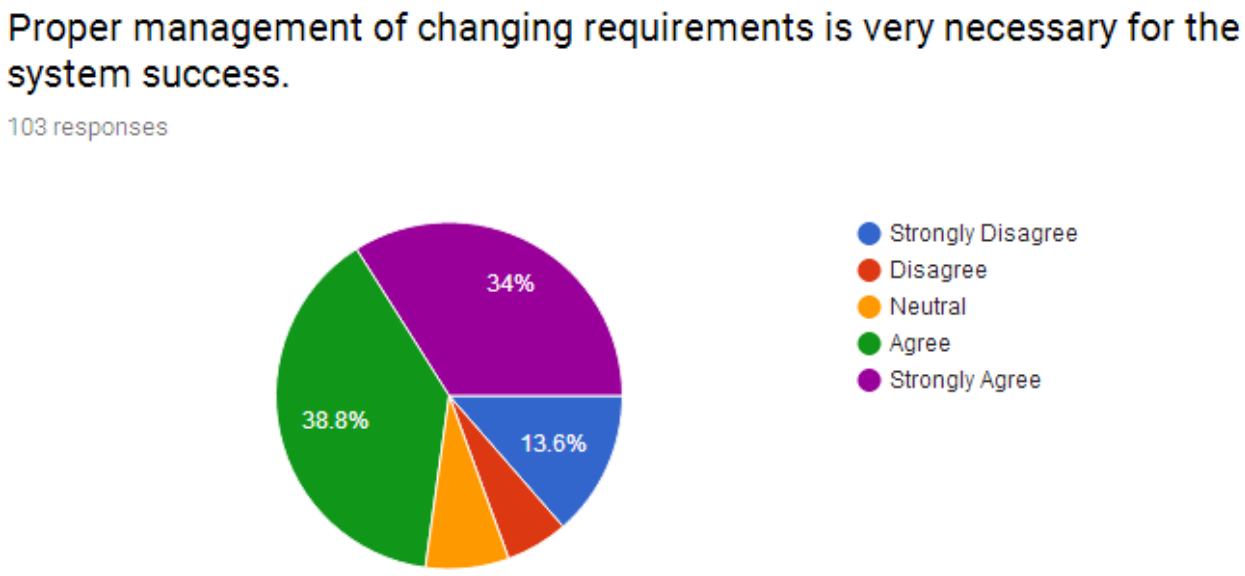

Figure 4.8 Proper Management of Changing Requirements is …

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No	30	29.1	29.1	29.1
	Yes	73	70.9	70.9	100.0
	Total	103	100.0	100.0	

Table 4.8 SPSS result of the statement "proper management of changing requirements is very necessary for system success".

4.9. When LEAN system fail…

Lean Practices are the software development procedural activity. If lean practices are not followed appropriately then what kind of circumstances is being faced, it's a realistic question about this research. There are some of methodologies having normal follow procedure but some have critically followed all the procedures and instructions. Lean has ability to complete the whole procedure in limited time. Below diagram show the result of question, that when lean practices are not followed appropriately then the system will fail. In the response 18.4% experts strongly agree with this, 33% professionals agree, while 8.7% neutral, 29.1% disagree and the 10.7% experts are strongly disagreeing with this statement. The results of this question finding tell us that, it has flexibility to complete the system.

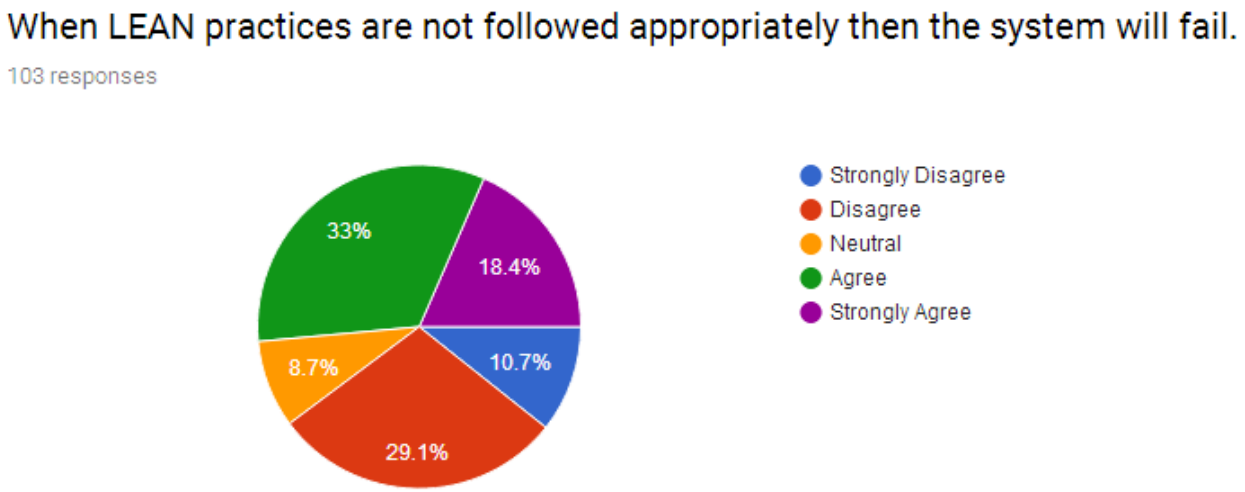

Figure 4.9 When LEAN system fail…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	34	33.0	33.0	33.0
	Disagree	30	29.1	29.1	62.1
	Neutral	9	8.7	8.7	70.9
	Strongly Agree	19	18.4	18.4	89.3
	Strongly Disagree	11	10.7	10.7	100.0
	Total	103	100.0	100.0	

Table 4.9 SPSS result of the statement "when lean practices are not followed appropriately then the system will fail".

4.10. Lean Produced software is flexible…

Lean Produced software is flexible, it has an ability to adopt changes but in different circumstances. It is necessary for software to be changeable at any time. If software is changeable for new requirements then it is the best software because it reduces the developer or designer time, cost and resources. Lean Produced software has changeability capacity. Below are the results the statement "Lean produced software us much flexible, it supports changing requirements of the users". 25.2% experts strongly agree with this statement, 32% agree, this mean that lean produced software is flexible; it supports changing requirements of the users. On the other hand 24.3% professionals disagree and 9.7% strongly disagree with this statement, it's all due to the lean adoptability change in different circumstances. Here in this opinion, the neutral opinions are minors in counting.

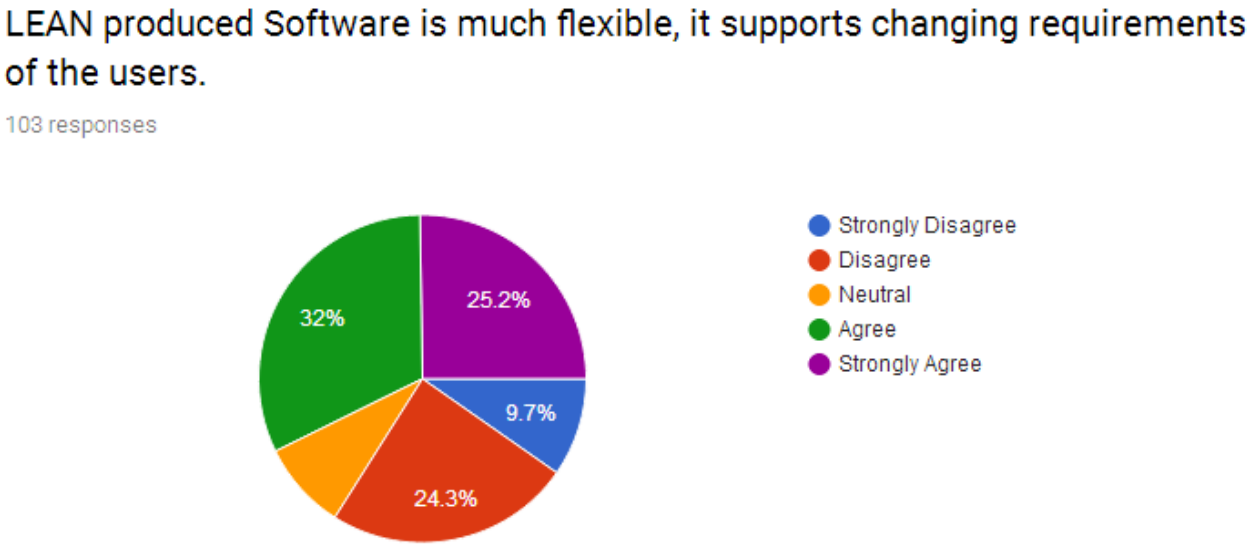

Figure 4.10 Lean Produced software is flexible…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	33	32.0	32.0	32.0
	Disagree	25	24.3	24.3	56.3
	Neutral	9	8.7	8.7	65.0
	Strongly Agree	26	25.2	25.2	90.3
	Strongly Disagree	10	9.7	9.7	100.0
	Total	103	100.0	100.0	

Table 4.10 SPSS result of the statement "Lean produced software us much flexible, it supports changing requirements of the users".

4.11. DSDM Produced software is flexible…

DSDM Produced software is flexible because it has a dynamic prospective. It is about truly understanding business needs, providing the fastest possible delivery of work solutions and software. It is necessary for software to be changeable at any time so that for this agile methodology is best. If software is changeable for new requirements then it is the effective software because it reduces time, cost and resources. DSDM Produced software is flexible because these support changing requirements of users. The responding question in the questionnaire is "DSDM produced software is much flexible; it supports changing requirements of the users". In response there are 43.7% users who strongly agree with the statement, 38.8% are those who agree and there are very small numbers of professionals, who are on the sides of neutral, disagree and strongly disagree.

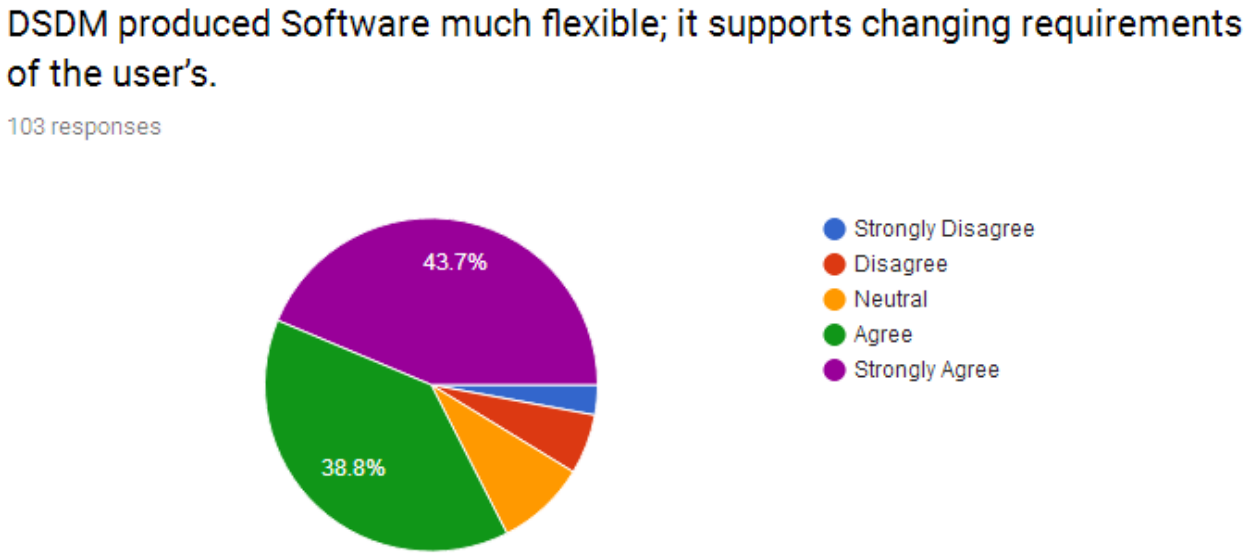

Figure 4.11 DSDM Produced software is flexible…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	40	38.8	38.8	38.8
	Disagree	6	5.8	5.8	44.7
	Neutral	9	8.7	8.7	53.4
	Strongly Agree	45	43.7	43.7	97.1
	Strongly Disagree	3	2.9	2.9	100.0
	Total	103	100.0	100.0	

Table 4.11 s SPSS result of the statement "DSDM produced software is much flexible; it supports changing requirements of the users".

4.12. LSD is easy to understand & reliable…

Lean Software Development promotes an incremental model which try to make strong adaptability, reliable than any other development process and creating more predictable quality products. It's quite difficult for fresh developers to precede all the steps to complete a project. It also creates difficulty during changeable requirements of the users. The user and developer response is quite jumble about this point "lean is easy to understand and reliable". When this question was raised to the professionals, 25.2% experts strongly agree with the statement, 23.3% professionals agree, 7.8% are neutrals, 29.1% disagree and 14.6% opinions are those who strongly disagree.

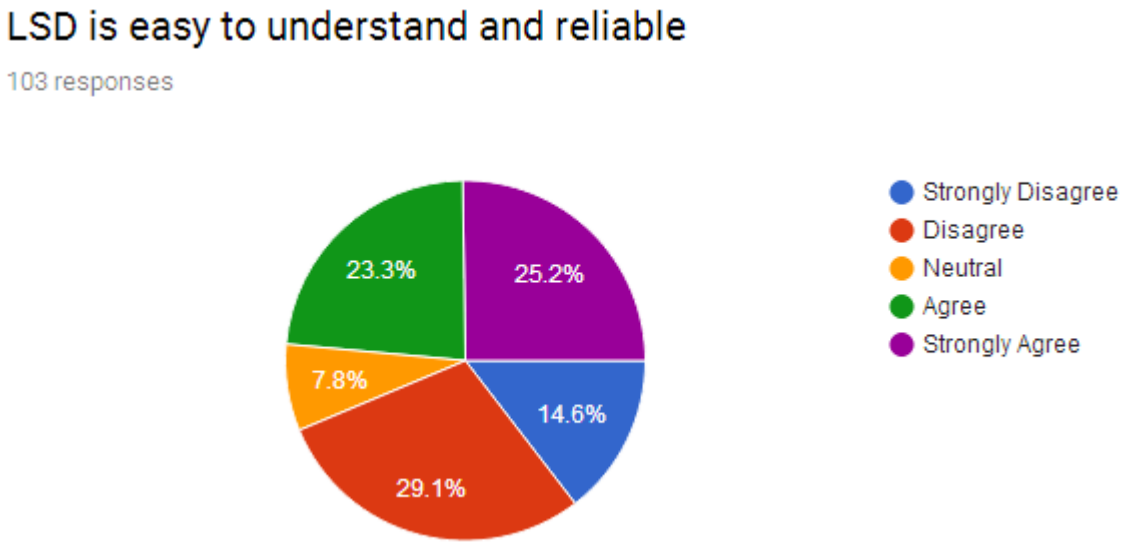

Figure 4.12 LSD is easy to understand & reliable…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	24	23.3	23.3	23.3
	Disagree	30	29.1	29.1	52.4
	Neutral	8	7.8	7.8	60.2
	Strongly Agree	26	25.2	25.2	85.4
	Strongly Disagree	15	14.6	14.6	100.0
	Total	103	100.0	100.0	

Table 4.12 SPSS result of the statement "Lean is easy to understand and reliable".

4.13. DSDM is easy to understand and reliable…

The DSDM framework is the foundation on the basis of best practices for the direct implementation of the project structure, the advantage is simple, scalable and in the past it turned out, but it does not pretend to choose design solutions. Due to its standards and principles, it is easy to understand and is reliable. The question asked to scholars in the questionnaire is that "DSDM is easy to understand or reliable". 42.7% experts strongly agree with this specific question, 35% professionals completely agree with that DSDM is easy to understand and reliable, 8.7% professionals are strongly disagree else on the other hand there is much limited number of people who are neutral and disagree.

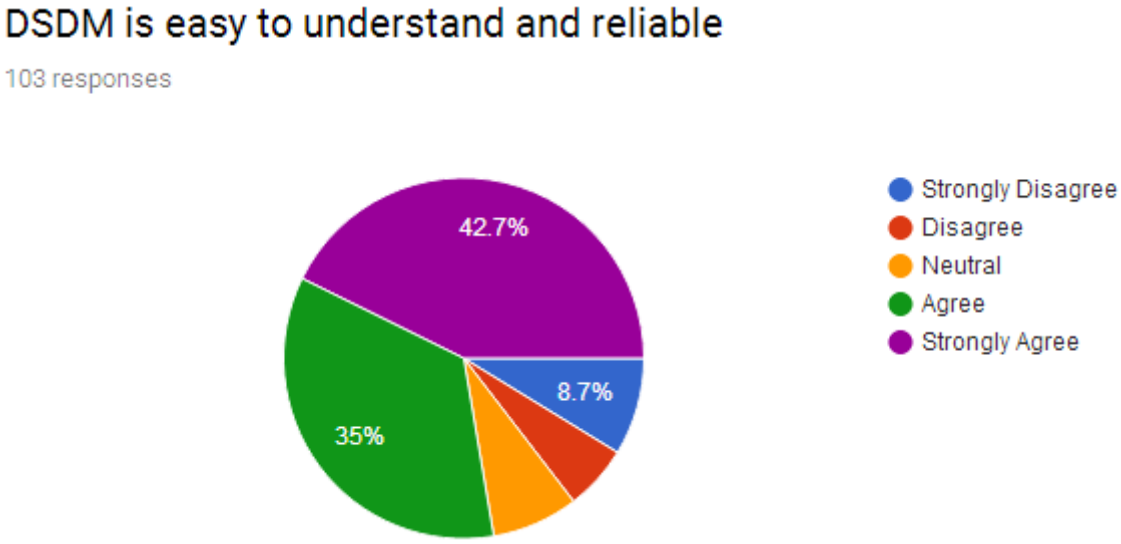

Figure 4.13 DSDM is easy to understand and reliable…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	36	35.0	35.0	35.0
	Disagree	6	5.8	5.8	40.8
	Neutral	8	7.8	7.8	48.5
	Strongly Agree	44	42.7	42.7	91.3
	Strongly Disagree	9	8.7	8.7	100.0
	Total	103	100.0	100.0	

Table 4.13 SPSS result of the statement "DSDM is easy to understand or reliable".

4.14. LEAN meets the requirements of the customer…

Developers have realized lean delivery fast should not mean the poor quality of the code. Fast delivery is to meet all the requirements of customers. The development model allows customers to change their minds mid-way through the project's demands, but fast delivery customers do not have time to change their minds. That's not good behavior with customer to divert their attention towards projects demand. A project must concisely meet the users' requirements. For this statement "The output given by lean meets the requirements of the customer", the responses are; 34% experts strongly agree, 23.3% agree, 25.2% disagree and 8.7% strongly disagree with the statement. Result shows that this statement is in favor of lean but the comparison with DSDM then the responses in favor of lean is low.

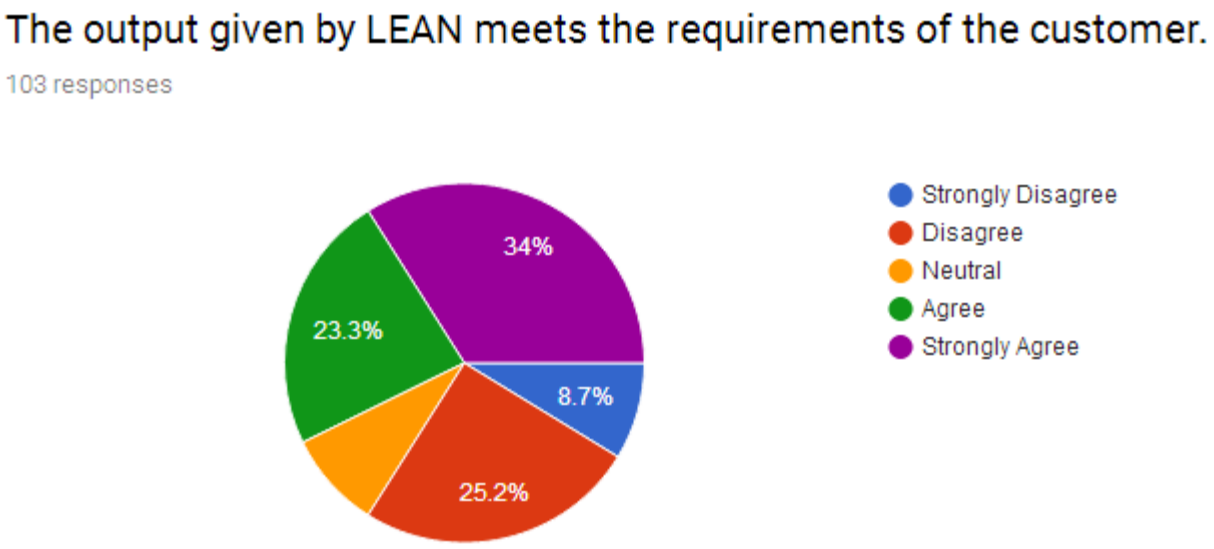

Figure 4.14 LEAN meets the requirements of the customer…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	24	23.3	23.3	23.3
	Disagree	26	25.2	25.2	48.5
	Neutral	9	8.7	8.7	57.3
	Strongly Agree	35	34.0	34.0	91.3
	Strongly Disagree	9	8.7	8.7	100.0
	Total	103	100.0	100.0	

Table 4.14 SPSS result of the statement "The output given by lean meets the requirements of the customer".

4.15. DSDM is according to the requirements of the customer…

A reliable and understandable methodology always succeeds in the market. DSDM positioning is mature; it is all due to methods of flexible development, flexible methods, and programming of the model. The feasibility study is that DSDM is a suitable methodological framework that has been identified as an industry research which is the basis for all follow-up work. It's all due to question that is "The output given by DSDM is according to the requirements of the customer". In result the 42% experts strongly agreed and 39.8% professionals completely agreed with that statement which shows the performance of DSDM while people who are neutral, disagree and strongly disagree are less in numbers

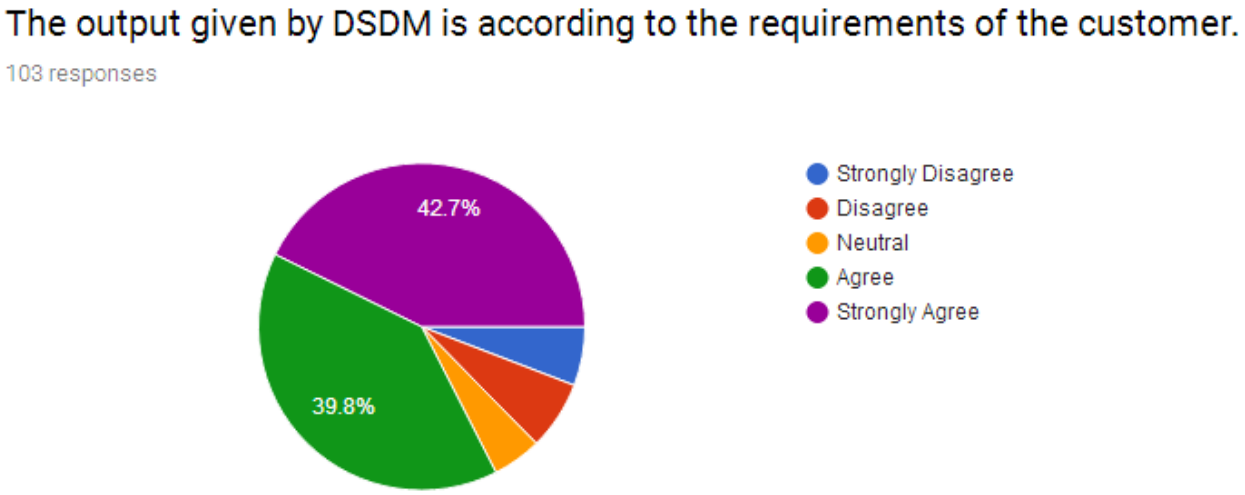

Figure 4.15 DSDM is according to the requirements of the customer…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	41	39.8	39.8	39.8
	Disagree	7	6.8	6.8	46.6
	Neutral	5	4.9	4.9	51.5
	Strongly Agree	44	42.7	42.7	94.2
	Strongly Disagree	6	5.8	5.8	100.0
	Total	103	100.0	100.0	

Table 4.15 SPSS result of the statement "The output given by DSDM is according to the requirements of the customer".

4.16. Timebox testing is efficient, comparison…

Timeboxing means that when a scheduled date occurs, regardless of whether the project is delivered, regardless of whether it is completed "completed", as originally provided. This is the ideal logical path of development. Timeboxing is a very important aspect of all DSDM-Atern projects. The project will be divided into "Increment Timeboxes", which can be broken down into "Timebox" development. The duration, resources and goals of each time period are rigid. Regarding this statement in the question is that “DSDM provides timebox testing which is more efficient than that of lean”. In response there are 35.9% experts strongly agree, 35% agree, 8.7% disagree, 12.6% strongly disagree and the neutrals are few in number.

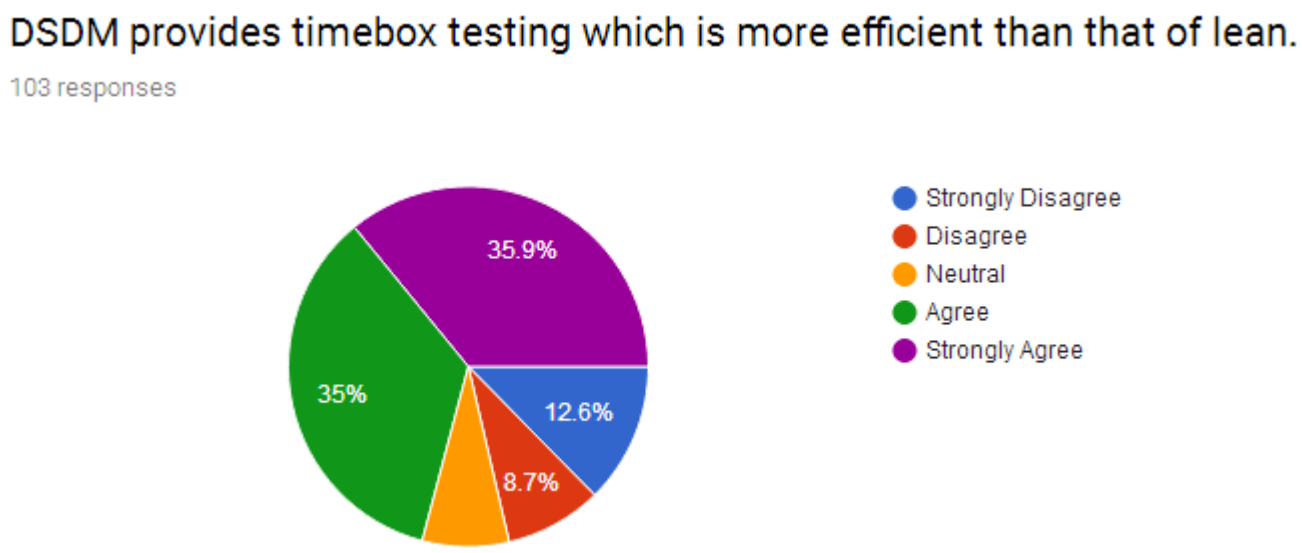

Figure 4.16 Timebox testing is efficient, comparison…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	36	35.0	35.0	35.0
	Disagree	9	8.7	8.7	43.7
	Neutral	8	7.8	7.8	51.5
	Strongly Agree	37	35.9	35.9	87.4
	Strongly Disagree	13	12.6	12.6	100.0
	Total	103	100.0	100.0	

Table 4.16 SPSS result of the statement “DSDM provides timebox testing which is more efficient than that of lean”.

4.17 DSDM dynamic way to defect identification…

Reliable software delivery is the key to the success of software development organizations. Successful and reliable software is available only if the requirements document is reliable. At the demand stage, there are many threats that lacks in the further phase of the software development process. Provided and improved the critical aspect of software reliability, it is important to be sure that the requirements for more stages must be reliable. The statement which is present in the questionnaire is that "DSDM provides a dynamic way to testing which means that defect identification made at different phases during development rather than at the end as in lean". In response to this statement there are 39.8% experts who strongly agree, 35.9% agree, 11.7% strongly disagree and there are limited number of neutrals and disagree peoples in response.

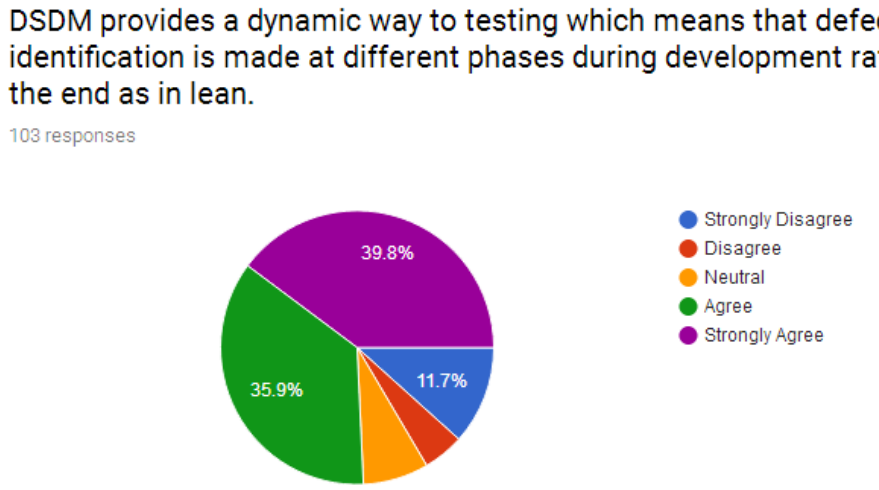

Figure 4.17 DSDM dynamic ways to defect identification…

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	37	35.9	35.9	35.9
	Disagree	5	4.9	4.9	40.8
	Neutral	8	7.8	7.8	48.5
	Strongly Agree	41	39.8	39.8	88.3
	Strongly Disagree	12	11.7	11.7	100.0
	Total	103	100.0	100.0	

Table 4.17 SPSS result of the statement "DSDM provides a dynamic way to testing which means that defect identification made at different phases during development rather than at the end as in lean".

Chapter 5
Results and Analysis

CHAPTER 5

5. Results and Analysis

In this chapter, the results of the effectiveness of agile development frameworks are presented. The information collected is based on the questionnaire filled by the experts and professionals. To carry out this research, three hypotheses were developed. To investigate and evaluate the hypothesis, a questionnaire was designed in context of problem statement. The questionnaire link was shared on social media groups and also forwarded to numerous software houses. Different questions from the questionnaire as previously discussed in the information collection with pi-chart shows the responses relevant to the questions. The hypotheses were tested through SPSS on the basis of which the hypothesis were approved or disapproved. One sample T-test was used to conduct analysis.

The confidence interval was taken to be 95%. The value of alpha was taken to be 0.05. The hypothesis was tested using one Sample T- Test and the results were analyzed on the basis of significance value.

5.1 Hypothesis# 1

DSDM is more efficient and effective agile development framework than LSD.

Following are the results of one sample T Test conducted in SPSS:

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
H1	103	2.8762	1.06979	.10541

One-Sample Test

	Test Value = 3					
	T	Df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
H1	-1.174	102	.243	-.12379	-.3329	.0853

Table 5.1. SPSS results for the hypothesis 1.

It can be seen from the above tables, that the mean value of responses (2.8762) is less than the test value (3) which means that the responses are more inclined towards the disagree side. But when we see the significance value it comes out to be p=0.243 which is greater than alpha=0.05 and shows that the difference between the mean and test value is not significant which means that the stated hypothesis is supported.

This hypothesis is based on the extensive literature review of existing work on frameworks of Agile Software development. This hypothesis basically focuses on comparing the effectiveness of DSDM over Lean. The hypothesis is tested by questions which address the efficiency, effectiveness, reliability, understanding, scheduling and resource management, quick delivery, shortcomings and usability.

The responses of people on the above mentioned dimensions were in favor of DSDM. This means that people found DSDM to be more reliable, understandable, effective, easy to use and free from errors as compared to LEAN. This is also evident from the above results of T-Test. So based on this information we can say that DSDM is a more effective agile software development framework than LEAN.

5.2 Hypothesis# 2

Project development by using the testing technique of DSDM have higher success rate as compared to those of LSD.

Following are the results of one sample T Test conducted in SPSS:

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
H2	103	3.6990	.88655	.08735

One-Sample Test

	Test Value = 3					
	t	Df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
H2	8.002	102	.000	.69903	.5258	.8723

Table 5.2. SPSS results for the hypothesis 2.

It can be seen from the table that the Mean (3.6990) of the sample is greater than the test value (3). Since the significance value p=0.000 is less than alpha=0.05, this means that the difference between the mean value and test value is significant, which means that the stated hypothesis is supported. This implies that a project developed by DSDM is more successful than software developed by Lean.

This hypothesis is solely associated with the project development techniques. Results of these techniques are compared on the bases of the success rate of both DSDM and LSD techniques in this hypothesis. It has been examined and analyzed by the questions which are addressing the output, flexibility, production and complexity.

The respondents had shown their interest in the favor of DSDM because they think that it is more flexible, adjusting and goal oriented than LEAN. One-Sample T-Test has been applied on the above mentioned responses which prove that DSDM comparatively have higher success rate than LEAN.

5.3 Hypothesis# 3

DSDM have a more scrutinized process of software development as compared to LEAN.

Following are the results of one sample T Test conducted in SPSS:

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
H3	103	3.7282	.83071	.08185

One-Sample Test

	Test Value = 3					
	T	Df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
H3	8.896	102	.000	.72816	.5658	.8905

Table 5.3. SPSS results for the hypothesis 3.

It can be seen from the table that the Mean (3.7282) of the sample is greater than the test value (3). Since the significance value p=0.000 is less than alpha=0.05, this means that the difference between the mean value and test value is significant, which means that the stated hypothesis is supported. This implies that a project developed by DSDM is more successful than software developed by Lean.

This hypothesis is the result of in depth research of work practiced in the field of DSDM and LEAN with respect to scrutinized process flow. DSDM and LEAN both are compared to know that which one of them is more error free. This hypothesis is venerated by the questions which are focusing on the flow of error free process of software development.

The responses of the questions related to hypothesis three (H3) are supporting the DSDM as compared to LEAN. T test applied on the responses, whose results have proven that according to respondents, DSDM is more scrutinized process of software development as compared to LEAN.

Chapter 6
Conclusion

CHAPTER 6

6. Conclusion

The overall intention of this research is to estimate the effectiveness of the agile development frameworks (LSD & DSDM) via thorough research, based on the responses of industry experts. This research also describes some of the issues and their solutions available for agile development frameworks. The goal of this research is to find a software development procedure, which is the most suitable out of the two agile development techniques. The results show the validity and effectiveness of DSDM agile model with respect to testing. This study will be useful in several aspects of software development lifecycle and help improve skills of software engineers for developing software according to the needs of end users.

This research investigates problem statement in different aspects to identify the clear differences between the two agile development techniques. The main task is to estimate the effectiveness of lean and dynamic systems development methodologies. In the concluding remarks, this research identifies that the dynamic systems development is effective than lean software development in all the perspectives. DSDM does not compromise on quality and delivers on time and eliminates waste, caters any late changes in development process and giving output compatible to the needs of the user. DSDM collaborates the whole procedural activity and provide information continuously and clearly during the development procedure as compared to lean. DSDM is an incremental process which means that all activities are sequential. DSDM and LSD both employ different testing techniques such as timebox testing, MoSCoW prioritization technique and flow of error free process of software development. DSDM show their effectiveness with respect to testing in the whole study.

The crux of this study is that institutes need to increase the quality of the quantitative and agile software development frameworks. Agile software development methodology, such as DSDM, is extensively being used in the industry and deserves further attention. We see that in this area there is a research backlog. It is necessary to create an agile software development framework particularly for dynamic systems development (DSDM) for future research.

6.1 Future Work

Future researches are required to develop an effective software development framework that is comprehensive in all aspects of software development. The software development lifecycle model is used to develop software products, in the same way, the agile models or frameworks are most commonly used in the software industry. In order to improve the functionality and specifications of agile development methods, some of the following issues need to be resolved.

6.1.1 Deal with high-level issues

The software industry is now being deployed and accessed globally, so the mass organization must ask a question to address the issues related to the lifecycle model. Every organization has designed the development lifecycle model according to their needs but there is no general framework present at the moment which is compatible with every organizations need.

6.1.2 Demand analysis

Many researchers have found that most of the software failures are because of the software development techniques do not conform to the users' requirement. Current software industry still has some problems which can be overcome through the agile model.

6.1.3 Improvement of efficiency measurement

We have conducted various analyses of the test methods to measure and monitor the efficiency of the agile model. Maintenance, design, development and other aspects can be made better which would result in a better agile development model. Since the companies are discreet in providing data, so it is really difficult to obtain a broader view.

6.1.4 Improved research

A lot of data on agile development methods is collected, but there is no general framework present which satisfies all the requirements of the customer. The reason is being the unavailability of data on such frameworks.

References

References

[1] Werner, Linda, et al. "Software engineering education via the use of corporate-sponsored projects: a panel discussion of the approaches, benefits, and challenges for industry-academic collaboration." Software Engineering Education and Training (CSEE\&T), 2013 IEEE 26th Conference on. IEEE, 2013.

[2] Fitzgerald, Brian, and Klaas-Jan Stol. "Continuous software engineering and beyond: trends and challenges." Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering.ACM, 2014.

[3] Wong, W. Eric. "Experience of teaching Executive Master's program in Software Engineering: Challenges, lessons learned, and path forward." Software Engineering Education and Training (CSEE\&T), 2014 IEEE 27th Conference on. IEEE, 2014.

[4] Petriu, Dorina C. "Challenges in integrating the analysis of multiple non-functional properties in model-driven software engineering." Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development.ACM, 2015.

[5] Dybå, Tore, and TorgeirDingsøyr. "Empirical studies of agile software development: A systematic review." Information and software technology 50.9 (2008): 833-859.

[6] GUPTA, RAM SHANKER, and VIJAY LAXMI. "Software Development Life Cycle (SDLC) Implementationin Information Technology \&Management."International Journal of Recent Advances in Information Technology \& Management 1.1 (2015).

[7] Kumar, Naresh, A. S. Zadgaonkar, and AbhinavShukla. "Evolving a new software development life cycle model SDLC-2013 with client satisfaction." International Journal of Soft Computing and Engineering (IJSCE) 3.1 (2013): 2231-2307.

[8] Johnston, Simon K., and Martin P. Nally. "Representing models in systems development lifecycle (SDLC) tools using a network of internet resources." U.S. Patent No. 9,122,422. 1 Sep. 2015.

[9] Turk, Dan, Robert France, and Bernhard Rumpe. "Limitations of agile software processes." arXiv preprint arXiv:1409.6600 (2014).

[10] Hoda, Rashina, James Noble, and Stuart Marshall. "Self-organizing roles on agile software development teams." IEEE Transactions on Software Engineering 39.3 (2013): 422-444.

[11] Stoica, Marian, MarinelaMircea, and BogdanGhilic-Micu. "Software development: Agile vs. traditional."InformaticaEconomica 17.4 (2013): 64.

[12] Abrahamson Pokka, Salado Outi, Ronkainen,Jussi and JuhaniWarsta. " Agile Software Development Methods". Espoo 2002. VTT Publications 478. 107 p. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf [Accessed: 16th March. 2017].

[13] Shelly. "Comparative Analysis of Different Agile Methodologies." International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 1, pp: (199-203)

https://www.google.com/url?sa=t\\\&rct=j\&q=\&esrc=s\&source=web\&cd=1\&ved=0ahUKEwiU546l4JnVAhVFWBQKHf\\bzAtUQFgglMAA\&url=http\%3A\%2F\%2Fwww.researchpublish.com\%2Fdownload.\\php\%3Ffile\%3DComparative\%2520Analysis\%2520of\%2520Different\%2520Agile\%25\\20Methodologies-1178.pdf\%26act\%3Dbook\&usg=AFQjCNEBC0zk3aAMLyBGdK_u\\hbkcQELXIQ

[Accessed: 09th February, 2017]

[14] Stoica, Marian, MarinelaMircea, and BogdanGhilic-Micu. "Software development: Agile vs. traditional."InformaticaEconomica 17.4 (2013): 64.

[15] [Nerur, Sridhar, RadhaKantaMahapatra, and George Mangalaraj. "Challenges of migrating to agile methodologies." Communications of the ACM 48.5 (2005): 72-78.

[16] Dingsøyr, Torgeir, and Nils Brede Moe. "Research challenges in large-scale agile software development." ACM SIGSOFT Software Engineering Notes 38.5 (2013): 38-39.

[17] Gandomani, TaghiJavdani, and Mina ZiaeiNafchi. "Agile transition and adoption human-related challenges and issues: A Grounded Theory approach." Computers in Human Behavior 62 (2016): 257-266.

[18] Qazi, AsadMasood, Sidra Shahzadi, and MamoonaHumayun. "A Comparative Study of Software Inspection Techniques for Quality Perspective." International Journal of Modern Education and Computer Science 8.10 (2016): 9.

[19] Kruchten, Philippe. "Contextualizing agile software development." Journal of Software: Evolution and Process 25.4 (2013): 351-361.

[20] Manifesto for agile software development. (2001). http://agilemanifesto.org/

[Accessed: 06th March. 2017].

[21] Poppendieck, Mary. “The Challenges of bringing lean to software development.” (2007)http://www.poppendieck.com/pdfs/The\%20Truck\%20Driving\%20\\Problem.pdf

[Accessed: 24th January. 2017].

[22] Poppendieck, Mary. “Principles of lean thinking.” (2002).http://www.poppen\\dieck.com/papers/LeanThinking.pdf

[Accessed: 29th April. 2017].

[23] Ophir, Eyal, Nass, Clifford, \& Wagner, Anthony D. “Cognitive control in media multitaskers.” (2009). http://www.pnas.org/content/106/37/15583.abstract

[Accessed: 17th November, 2016].

[24] Poppendieck.llc http://www.poppendieck.com/

[Accessed: 28th November, 2016].

[25] Poppendieck, Mary. (2003, Fall). Lean software development. C++ Magazine, methodology issue, Retrieved from http://www.popp\\endieck.com/pdfs/Lean_Software_Development.pdf

[26] Feyh, Markus, and Kai Petersen. "Lean software development measures and indicators-a systematic mapping study." Lean Enterprise Software and Systems.Springer, Berlin, Heidelberg, 2013.32-47.

[27] Antinyan, Vard, et al. "Identifying risky areas of software code in Agile/Lean software development: An industrial experience report." Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on. IEEE, 2014.

[28] Reinertsen, Donald, and Tom Bellinson. "The principles of product development flow: second generation lean product development." (2014).

[29] Feyh, Markus, and Kai Petersen. "Lean software development measures and indicators-a systematic mapping study." Lean Enterprise Software and Systems.Springer, Berlin, Heidelberg, 2013.32-47.

[30] Rodríguez, Pilar, et al. "Combining lean thinking and agile methods for software development: A case study of a finnish provider of wireless embedded systems detailed." System Sciences (HICSS), 2014 47th Hawaii International Conference on.IEEE, 2014.

[31] Rodríguez, Pilar, et al. "Building lean thinking in a telecom software development organization: strengths and challenges." Proceedings of the 2013 international conference on software and system process.ACM, 2013.

[32] Poppendieck, Mary, Poppendieck, Tom. (2002). Agile software development ecosystems. Addison-Wesley Professional.

[33] Poppendieck, Mary. (2001, May).Lean programming. Software Development Magazine, Retrieved from http://www.poppendieck.com/lean.htm

[34] Sani, Abdullahi, et al. "A review on software development security engineering using dynamic system method (DSDM)." International Journal of Computer Applications 69.25 (2013).

[35] Mead, Nancy R., VenkateshViswanathan, and DeepaPadmanabhan. "Incorporating security requirements engineering into the dynamic systems development method."Computer Software and Applications, 2008.COMPSAC'08.32nd Annual IEEE International.IEEE, 2008.

[36] Aydin, Mehmet N., et al. "Adaptation of an agile information system development method." Research Issues in Systems Analysis and Design, Databases and Software Development.IGI Global, 2007.54-88.

[37] DSDM Consortium: DSDM Handbooks, Agile business consortium. https://\\www.agilebusiness.org/resources/DSDM-handbooks

[Accessed: 21st April, 2017]

[38] Abrahamsson, Pekka, et al. "New directions on agile methods: a comparative analysis." Software Engineering, 2003. Proceedings. 25th International Conference on. Ieee, 2003.

[39] Coyle, Sharon, and Kieran Conboy. "A study of risk management in DSDM." Agile processes in software engineering and extreme programming (2009): 142-148.

[40] Highsmith, Jim, and Alistair Cockburn. "Agile software development: The business of innovation." Computer 34.9 (2001): 120-127.

[41] KariemSlegten, and Daniel Alami. “Dynamic Systems Development Method (DSDM)”http://www.students.science.uu.nl/~5766664/me/Review\%20Kariem\%20\\Slegten /PeerReview_KariemSlegten.pdf

[Accessed: 19th October, 2016]

Appendix

Appendix

Questionnaire

The scale is defined below:

Strongly Disagree = A

Disagree = B

Neutral = C

Agree = D

Strongly Agree = E

Yes = 1

No = 2

1. Agile software progress increases due to the participation of the whole team in the plan with project manager.
- 1
- 2
2. In agile, team collaboration is the main thing to increase the productivity.
- 1
- 2
3. Agile methodologies help to improve the testing efficiency.
- 1
- 2
4. The agile process helps to build a product of professional quality which fits the business need.
- 1
- 2
5. DSDM methodology provides the comprehensive foundation for planning, managing, executing and scaling agile process and iterative software development.
- 1
- 2
6. LSD emphasizes on the speed and the efficiency of software workflow and provides reliable feedback between programming and customer.
- 1
- 2
7. LSD methodology eliminates waste, through such practices as selecting only the truly valuable feature for a system. Then prioritizing the selected features, and finally delivering them in small batches.
- 1

- 2

8. Proper management of changing requirements is very necessary for system success.

- 1
- 2

9. When lean practices are not followed appropriately then the system will fail.

- A
- B
- C
- D
- E

10. Lean produced software us much flexible, it supports changing requirements of the users.

- A
- B
- C
- D
- E

11. DSDM produced software is much flexible; it supports changing requirements of the users.

- A
- B
- C
- D
- E

12. Lean is easy to understand and reliable.

- A
- B
- C
- D
- E

13. DSDM is easy to understand or reliable.

- A
- B
- C
- D
- E

14. The output given by lean meets the requirements of the customer.

- A
- B
- C
- D
- E

15. The output given by DSDM is according to the requirements of the customer.
- A
- B
- C
- D
- E

16. DSDM provides timebox testing which is more efficient than that of lean.
- A
- B
- C
- D
- E

17. DSDM provides a dynamic way to testing which means that defect identification made at different phases during development rather than at the end as in lean.
- A
- B
- C
- D
- E

18. DSDM is easy to understand and reliable
- A
- B
- C
- D
- E

19. LSD is easy to understand and reliable
- A
- B
- C
- D
- E

20. The scheduling & resource management is ignored in DSDM
- A
- B
- C
- D
- E

21. The scheduling & resource management is ignored in LSD
- A
- B
- C
- D

- E

22. LEAN provides quick delivery of results and there are no shortcomings in the outcome.

- A
- B
- C
- D
- E

23. Software created by DSDM is easy to use as compared to software created by LEAN.

- A
- B
- C
- D
- E

24. Requirements clearly meet with the output in DSDM

- A
- B
- C
- D
- E

25. DSDM produced Software is much flexible; it supports changing requirements of the user.

- A
- B
- C
- D
- E

26. LEAN produced Software is much flexible; it supports changing requirements of the users.

- A
- B
- C
- D
- E

27. Software developed by DSDM is compatible with the existing systems.

- A
- B
- C
- D

- E

28. DSDM provides a dynamic way of testing which means that defect identification is made at different phases during development rather than at the end as in lean.

- A
- B
- C
- D
- E

29. DSDM corrects the detected defects at every stage and verifies the correction after every test but lean does not.

- A
- B
- C
- D
- E

30. MoSCoW is a prioritization testing technique, which supports an effective development strategy with respect to testing in DSDM but this technique does not exist in lean.

- A
- B
- C
- D
- E