

**Adopting Design Thinking Practices to Satisfy Customer  
Expectations in Agile Practices: A Case from Sri Lankan  
Software Development Industry**

W.M.D. Ruchira Prasad

(169124K)

Degree of Master of Business Administration in Information Technology

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2018

## **DECLARATION**

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

.....

W.M.D. Ruchira Prasad

(Signature of the candidate)

Date:

The above candidate has carried out research for the Masters thesis under my supervision.

.....

Dr. Indika Perera

Signature of the Supervisor

.....

Date

## **COPYRIGHT STATEMENT**

I hereby grant the University of Moratuwa the right to archive and to make available my thesis or dissertation in whole or part in the University Libraries in all forms of media, subject to the provisions of the current copyright act of Sri Lanka. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

## ABSTRACT

While application of Agile principles leads to better project success, some projects are still failing due to lack of understanding of exact client requirements. Agile teams have recently started adopting Design Thinking (DT) to better understand what is in customers' mind. In this research, we explore DT practices in Agile teams using inductive reasoning. The research first formulated a conceptual framework based on a literature review. Based on this, conducted a set of interviews where the researcher interviewed ten professionals, including project managers, business analysts, tech leads, and architects of different IT service organizations. Interview findings were then analyzed using the Straussian grounded theory. Customer journey, story mapping, prototypes, POC, UX, and scenarios were identified as the most suitable methods to identify real need of the customer. Moreover, practicing human-centered approach through workshops, discussions, team communication, and end-user interaction through UAT were also identified to be effective. Based on these findings, this research further derived a model to achieve customer satisfaction through DT in agile-base projects. The proposed model categorizes best practices into five categories such as customer real need identification, transforming customer real need into pilot solutions, visualizing pilot solution for customer feedback, idea generation for pilot solution, and brainstorming.

**Keywords:** agile practices; design thinking; design thinking practices; ground theory

## **ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to all those who helped me in successfully completing my research study on “Design Thinking Practices to Satisfy Customer Expectations in Agile Practices”.

First and foremost, I wish to thank my research supervisor Dr. Indika Perera, Senior Lecturer of the Department of Computer Science and Engineering, University of Moratuwa, for the continuous support, encouragement and attention extended to me in realizing the research objectives.

I wish to convey my special gratitude to respondents of the software development companies who gave me their precious time and valuable insights in making this research a reality. Furthermore, I extend my earnest thanks to MBA research coordinator Dr. Chandana Gamage and Ms. Jeeva Padmini for the guidance and resources provided in bringing this study a success.

Finally, I wish to convey my heartfelt thanks to all those who helped me in many ways for bringing this study a success.

# TABLE OF CONTENTS

DECLARATION.....	I
COPYRIGHT STATEMENT .....	II
ABSTRACT.....	III
ACKNOWLEDGEMENT.....	IV
TABLE OF CONTENTS .....	V
LIST OF FIGURES .....	VII
LIST OF TABLES .....	VIII
LIST OF ABBREVIATIONS.....	IX
1. INTRODUCTION .....	1
1.1. Background.....	1
1.2. Motivation.....	2
1.3. Research Scope .....	4
1.4. Problem Statement.....	5
1.5. Research Objectives.....	6
1.6. Research Significance.....	6
1.7. Outline .....	7
2. LITERATURE REVIEW .....	8
2.1. Design Thinking .....	9
2.1. Design Thinking Elements.....	16
2.3. Design Thinking Vs Engineering Thinking .....	20
2.4. The Role of Customer Expectations.....	20
2.5. Agile .....	21
2.6. Synthesis of Design Thinking and Scrum .....	23
2.7. IT Maturity in Sri Lanka .....	24
2.8. Summary .....	24
3. RESEARCH METHODOLOGY.....	26
3.1. Research Approach and Method.....	26
3.2. Population and Sample Selection .....	30
3.3. Data Collection .....	31
3.4. Process of Data Collection.....	31
3.5. Grounded theory approach.....	32
3.6. Summary.....	33

4.	DATA ANALYSIS .....	35
4.1.	Data Preparation for Analysis.....	35
4.2.	Data Analysis.....	35
4.3.	Summary.....	57
5.	RECOMMENDATIONS AND CONCLUSION.....	58
5.1	Conclusion .....	58
5.2	Observations, findings and recommendations .....	60
5.3	Research limitations.....	66
5.4	Future work.....	67
	APPENDIX A: INTEVIEW QUESTIONS.....	71
	APPENDIX B: REVISED INTERVIEW QUESTIONS .....	74
	APPENDIX C: MEMO WRITING OF GROUNDED THEORY .....	77
	APPENDIX D: CODING BASED ON GROUNDED THEORY.....	79

## **LIST OF FIGURES**

Figure 2.1 Design Thinking Stages	10
Figure 3.1 Research Methodology	28
Figure 3.2 Conceptual framework	29
Figure 3.3 Population and sampling	30
Figure 4.1 Usage of agile methodologies	36
Figure 4.2 Dimensions related to human-centered approach	38
Figure 4.3 Dimensions related to thinking by doing	42
Figure 4.4 Dimensions related to Visualizing	46
Figure 4.5 Dimensions related to Synthesis of divergent and convergent	50
Figure 4.6 Dimensions related to Collaborative work style	53
Figure 5.1 Framework to enhance customer satisfaction using DT practices in agile practices	62



## **LIST OF TABLES**

Table 2.1 DT Categories and Elements	16
Table 3.1 Design Thinking Categories and Elements	29
Table 4.1 Profiles of interviewed	35
Table 4.2 pain points in customer satisfaction and ways to improve	37
Table 4.3 Open and Axial coding for human-centered approach	41
Table 4.4 Open and Axial coding for thinking by doing DT practice	45
Table 4.5 Open and Axial coding for Visualizing	49
Table 4.6 Open and Axial coding for Synthesis of Divergent and Convergent	52
Table 4.7 Open and Axial coding for Collaborative work style	54
Table 5.1 Observations identified when conducting interviews	60
Table 5.2 Best practices to satisfy customer expectations	61
Table 5.3 DT practices and factors	64

## LIST OF ABBREVIATIONS

ASD	Adaptive Software Development
DSDM	Dynamic systems development method
DT	Design Thinking
ET	Engineering Thinking
FDD	Feature-Driven Development
HCA	Human Centered Approach
XP	Extreme Programing

# 1. INTRODUCTION

## 1.1. Background

Agile framework is an iterative and incremental process. It delivers products in collaboration with cross-functional teams. Main objective of introducing agile into software development industry was to achieve rapid delivery of high-quality products while responding to frequent changes in customer's mind. Extreme Programming (XP), Scrum, Dynamic systems development method (DSDM), Adaptive Software Development (ASD), Crystal, Feature-Driven Development (FDD), Pragmatic Programming, and Lean Development are practices under agile umbrella. Though all the practices adhere to agile principles and values, each practice have their own recipes, terminology, and tactics when it comes to software implementations. Design Thinking (DT) is also an iterative process. Which is a creative strategy to provide solutions to complex problems. It is centered on thoroughly understanding customer's needs and try as many possibilities before coming up with a preferred solution. DT is extremely useful for tracking complex problems that are ill-defined or unknown. In this approach, problem is first analyzed and experimented with different designs. The best solution is then developed out of several experimental designs. Development process needs to be flexible to use this approach and apply best solution that fits to the problem.

As described by Holloway (2009), DT helps designers to resolve the problem by giving a 360-degree view of the problem. By considering different aspects of the problem, such as end user requirement, social factors of the environment, current market, and trend, designer can get 360-degree view. DT helps to look beyond what you see in the tradition problem solving methods, and it gives innovative ideas with considering environmental and social factors.

Moreover, based on Clark & Smith (2008) advantage of design thinking is for managers to deep understanding of customers that what they really want and how they expect it and deliver results that delight the customers. Moreover, inventors can use DT to win the market with innovative product development. Understand what

customer really want is the most critical part in software development. Major reason for many project failures are misunderstanding user requirements and failure to manager end-user expectations (Al-ahmad, Al-fagih, & Khanfar, 2009). Moreover, described by Kaur & Sengupta (2011), it's take considerable time and a lot of communication needs be happen to clearly defined a requirement. Otherwise, project team don't know what customer really require. Other thing is, customer's lack of knowledge and experience to describe what they really require. DT practice can be used to understand what customer really wants at the initial stage of the project.

Agile is primarily about how project is delivered with planning, scope and budget while time and quality are fixed as well as while adding value to the customer iteratively. DT is focusing on exploring and generating ideas while understanding users and innovation to meet customer needs. Putting them together, amplifies the innovation process with achieving success of the delivery.

While DT is known to be highly effective in other industries such as entertainment, medicine, education, and space exploration, it is a novel practice to software development, especially in the agile content (Razzouk & Shute, 2012).

## **1.2. Motivation**

Almost every software development companies in Sri Lanka are multinational companies. Many of them follow agile methodologies such as; Extreme Programming (XP), Scrum, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Feature-Driven Development (FDD), Pragmatic Programming, and Lean Development are practices under agile umbrella. Hence, Scrum process meet end user requirements fast and provide solution and deliver quality software incrementally by using an iterative process with adding value to the customer incrementally.

Lucena, Braz, & Tizzei, (2016) further describes, even Agile emphasizes the collaboration with customer, but does not guarantee that software development team work towards to solve correct problem. But, Agile methodologies could achieve higher success rate than traditional methodologies (Lucena, Braz, & Tizzei, 2016).

However, the question too be raised is “Is it possible to deliver the features with achieving the client’s expectation and quality?” Furthermore, end user expectation is the most critical aspect to deliver the correct product to the customer. The final result of the product is totally depending on the implemented requirements. Nevertheless, in many scenarios, customer had accepted the product since they might not be any other solution because they also paid to the service provider iteratively. Whereas customer satisfaction is not achieved. This is known to be a major issue in software development and which is need to be addressed. Researcher aims to study how the right feature and the quality aspect is contributing to the customer satisfaction.

Companies have been developing many strategies and approaches to address customer satisfaction. However, it is being challenging for many companies to satisfy customer with complexity of technological innovation. Software development industry totally depend on innovation and continuous innovation is a must to sustain in competitive technology industry. However fast deliveries of Agile, do not satisfy user requirements in the end due to lack of proper design or solutions (Gurusamy, Srinivasaraghavan, & Adikari, 2016). Generally, some software companies in Sri Lanka focus to deliver product on time and budget without focusing on the best possible solution that can be given to customer’s requirements. Hence, the question to be raised is “How companies can identify the correct feature and provide best possible solution with quality of the product?”

Even though agile is a successful software development methodology practiced worldwide the question to be raised “can agile development provide customer satisfaction by implementing the right feature?” Gurusamy et al. (2016), further describes Agile is a successful approach for software development however achieving customer satisfaction by implementing the correct feature is questionable. DT is the new norm adopted by agile software development in order to identify the real need of customers and identify their expectation and provide best possible solution for it. Therefore, with applying DT to agile, can provide best solutions for the requirements and finally deliver a product which making customer happy. Because DT helps to define goals clearly, understand customers deeply, and align team to deliver the results (Clark & Smith, 2008). Nevertheless, this practice is not a

new approach for many companies, whereas they practice DT without the knowledge of DT. The researcher aims to identify how service based software development companies using DT practices to capture the customer real need and achieve the satisfaction of the product.

Moreover, I'm not going to prove DT practices have positive affect to customer satisfaction. Because DT practices always have positive affect to satisfy customer as well as there is no any literature that mentioning there is a negative effect of DT practices to satisfy customer. Since the IT industry in Sri Lanka is well matured industry, DT is ready to apply with Agile.

### **1.3. Research Scope**

The scope is limited to the mid and large size service based multinational software development companies which having operations in Sri Lanka and who follow the agile as the software development methodology.

Companies and the employees who aware about design thinking and practically follow design thinking practices with agile software development practices.

Identify the industry awareness and usage of applying design thinking for agile practices.

Advantages that can gain to managers and leaders with combining DT practices with agile practices.

In this research I'm considering only use of DT practices to satisfy customer expectations and only show the importance of DT practices to customer satisfaction and analyzing the importance of DT. I'm not considering any technical tools to identify information that related to customer expectation such as critical functions, performance of the product etc.

#### **1.4. Problem Statement**

Agile made significant improvement than waterfall software development methodology. It developed smaller functions of application rather developing entire application then test and made improvement and finally delivering that smaller functions to the customer iteratively. However, in other way, it made things worse, fail fast and fail often with problems. Then needs to be learn from it, develop the same function of the product again, test and deliver. This iterative approach needs to be followed till the customer accept that function of the software product. This make more time wastage of the development team and needs to be allocated more budget.

According to Saiedian & Dale (2000) understanding of the real requirement is the most critical step in the software development. Many projects failed due to unclear user expectations and their environment. There should be a way to identify the problem to be addressed and ideate a best possible solution. It should be able to provide early design with visualizations. Then they can prototype the idea quickly and gather some quick feedback on it. It should get feedback from the customer before involve to the actual development to reduce the iteratively develop same function respectively.

Without ideating the best solution for the identified problem, can agile lead to build solution more quickly? Without proper information and methods, agile simply build badly designed solutions (Gurusamy et al., 2016). This kind of failures should be avoided to optimize the design and build the right solution iteratively with minimizing the chance of failures.

Based on the literature, identified design thinking elements that can be used with agile software development methodology that can provide the customer expected solution with increasing customer satisfaction. These elements have been divided into three categories (Practices, Cognitive Approach, and Mindset) based on the three dimensional framework mentioned by (Hassi & Laakso, 2011). In this research, I have considered only the elements are in the practices category and how they can be used to identify customer expectations correctly and provide customer expected solution with the quality. Hence, best practices and benefits of DT are largely

unknown. Agile and DT practices go hand in hand because both are iterative processes.

Therefore, I framed the question to fill that gap as:

*"How to apply design thinking practices to improve customer expectations in agile practices?"*

### **1.5. Research Objectives**

Following are the specific objectives of the research:

- Identify design thinking elements that can be applied for agile practices.
- Identify the use of design thinking practices within agile practices.
- How to make use of design thinking practices for effective project management in agile teams.

### **1.6. Research Significance**

Design is widely considered as a critical aspect of engineering. Designing effective solutions can meet the social needs. When managing a software development project, there should be a solid understanding of design process, project scope, cost, timeframe, deadline, what customer is expecting, and how to deliver a quality product. To deliver a successful project needs to be managed through these variables. Frequently Agile achieve deadlines with considering time, cost, and scope. The best way of achieving deadline is providing customer expected solution with making the customer happy. Agile came in handy to add value to the customer iteratively, minimize the failures and achieve deadlines within the budget. But there is a question to be raised whether agile provides best possible solutions to the customer with increasing his satisfaction.

This research contributes to identify the real problem of the customer with considering customer's business context and social context and get the customer perception and end-user feedback before going for actual implementation. If the project team can clearly understand the correct requirement or the real problem of the customer before going for actual implementation, no need to wait till develop the product to get the feedback as an iterative process of agile. Many scenarios the



customer accepting the given solution after giving feedbacks for many iterations. Such situations, it is required to apply changes to the implemented functionality many times. Therefore, will have to allocate more cost and time to finalize the functionality. But with DT can provide the best possible solution without going for several iterations. With DT can satisfy customer with providing customer expected solution in additions to rapid delivery and accepting frequent changes of Agile. Finally, it has been provided a framework that can apply DT to software development process to satisfy customer expectations.

### **1.7. Outline**

Rest of the thesis is organized as follows. Chapter 2 presents an extended literature review on covering topics, design thinking, design thinking elements, design thinking vs engineering thinking, the role of customer expatiations, agile, and synthesis of design thinking and scrum. Chapter 3 represent the research approach and method, population and sample selection, data collection, process of data collection, and grounded theory approach. Chapter 4 discusses the data preparation for analysis, and data analysis. Chapter 5 provides the conclusion of the study, observation, finding and recommendations, research limitations, and future work.

## 2. LITERATURE REVIEW

Agile principles and values emphasize on collaboration with customer, early and continuous delivery in a shorter time scale. Iterative and incremental process in agile makes ensures customer feedback within shorter time scale. Agile methodologies could achieve higher success rate than traditional methodologies (Lucena et al., 2016). It has further been described by Lucena (2016), agile emphasizes the collaboration with customer, but does not guarantee that software development team work towards to solve correct problem. Agile methodologies such as scrum already incorporate with getting user feedbacks as the part of requirement process. It created user stories to understand the problem with adopting simple requirement gathering process defined by a single product owner (PO) who is the representative of the customer and the end-users. Under these conditions the development team implements a limited solution for the problem which is described in the backlog (Lucena et al., 2016). While agile practices are known to substantially increase the project success rate (Jeeva Padmini, Perera, & Dilum Bandara, 2016), still a significant number of projects fail. This is mainly due to the lack of understanding of customer's expectations (Hussain, Mkpojiogu, & Kamal, 2016).

Moreover, described by Jeeva Padmini, Perera, & Dilum Bandara (2016), according to the real time case information collected in "The chaos report", agile practices reduce the project failure rate. However, significant fraction of projects still fails even though the development teams practicing agile. This is mainly due to the lack of understanding of customer expectations. At the project initiation stage, if the customer can spend valuable time with vendor to properly understand what he/she really wants, it leads to better process success. Furthermore, end user expectation is the most critical aspect to deliver the right product to the customer.

The primary measurement for determine the information system success is achieve its original purpose (Vetterli, Brenner, Uebernickel, & Petrie, 2013). Most concern aspect of a project is requirement engineering to achieve the real-world goals. Further explained by Vetterli et al. (2013), analyzing requirement is a very complex process as well as requirement gathering always starts with ill-defined and

conflicting ideas. Therefore, requirement engineering needs a very extensive analysis to identify the components such as; technological, legal, human, and cultural factors.

As the product implementation primarily depends on the requirement gathered by client, DT can be used to eliminate project failures. DT can be described as a methodology, process, or approach for eliciting customer needs rather than gathering requirement (Vetterli, Brenner, et al., 2013). With providing fast and simple prototypes, it can be eliciting customer needs effectively.

Rest of the chapter aligns as below. Section 2.1 describes about design thinking which covers the importance of DT, and DT is software development. Section 2.2 discuss DT elements. Section 2.3 discuss a comparison on DT vs. engineering thinking. Section 2.4 further elaborates on the roles of the customer expectation. Section 2.5 and section 2.6 discuss the agile and synthesis of design thinking and scrum process.

## **2.1. Design Thinking**

DT is an iterative and incremental creative process. It helps business to find solutions for complex problems and generate new ideas for product development. Figure 2.1 illustrates how the DT process evolves through empathize, define, ideate, prototype, and test stages (Mickahail, 2015). Within each stage, problems are framed, questions are encouraged, along with more ideas, until the best answers are evident and chosen. In *empathizing* user starts to explore the problem and try to understand it. In this stage, user can consult domain experts, and observe, engage, and empathize with people. Within a limited time, user develops best possible understanding about the problem. In *define* stage user defines the problem with the help of information gathered during the empathize stage and formulate the problem in the form of a question. During the *ideate* stage user starts generating ideas, and this process can be supported by ideation tools. Then user *prototypes* the initial version of the product. Key idea is to understand the best solution for the problem. Finally, during the *test* stage, designers rigorously test the product. Because this is an iterative process, team

can use the identified solutions to evaluate again to identify the best possible solution.

DT developed gradually over last decades as a required best practice for business to solve problems for developing new products.

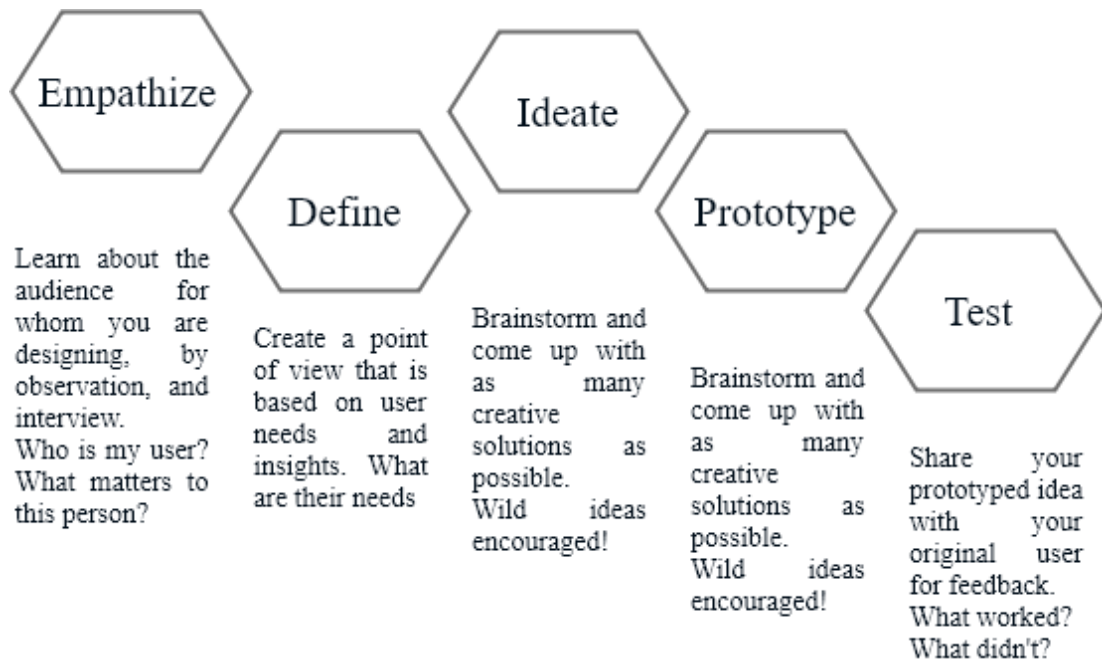


Figure 2.1 Design Thinking Stages

Source: (Mickahail, 2015)

## Empathize

- What – Empathize is the human-centered design process with understanding people. It is a way of understanding what the people doing, why they do such, their physical and emotional needs, how they think about the world etc.
- Why- As a design thinker needs to be understand the real problem, who are the people and what they need. Observing people, you can get a clue about what they think and feel. Furthermore, you can get their experience and make insights based on that. These insights direct you to innovative solutions.
- How – You can empathize by,
  - Observing – by interviews understand the behavior of the users.

- Engaging – convert interviews to conversations and get the stories from the users.
- Watch and Listen – Ask from users how they are doing tasks and go through the steps and ask why they do so and what they do. Go for deeper questions to understand the situation.

### **Define**

- What – Define will create a meaningful and actionable problem statement. This should be a guiding statement based on the insights and need of particular user.
- Why – This is critical for design process. It explicit the expression of the problem you are striving to address.
- How – consider what needs to be taken when you taking and observing people. If you noticed something interesting, ask what it might be.

### **Ideate**

- What – Ideate is come up with design process which you concrete on idea generation.
- Why – You are ideating for identified problems to provide solutions for your customers. It will ideate possible range of solutions, then several good solutions. The solution will come later through the user testing and feedback.
- How – brainstorming you and your team to reach new ideas by building on others ideas. Then adding constraints, inspiring related materials and embracing misunderstanding you can find a solution.

### **Prototype**

- What – Prototypes iteratively generating artifacts which closer to the final solution.
- Why – Prototypes for ideate and solving problems. Build and think about the solution. To communicate with the customer with showing a picture about the

solution. Then it can give to end users for testing and get feedbacks. If it not expected solution, will fail quickly and cheaply. This will break the large problem down to small testable chunks.

- How – Start building the prototype based on the idea. Then start testing. It will give all the answer the questions when tested. Answering the questions will help focus you prototyping and meaningful feedback from end-users.

## **Test**

- What – Test is the phases of gather feedback for the prototype. Based on feedback you can understand your end-user. Focus on what you can learn from your end-user and the problem and your potential solution.
- Why – Testing feedbacks are the next iteration of your prototype. It will build empathy through observation and engagement. If the solution fails, it will fail at the initial steps.
- How – give you prototype to your end-user's hand ask to test it with their experience. Watch how they use it, handle and interact with it. Then listen to their feedbacks and questions they have.

In each stages, identify the problem, emerge questions with more ideas until find the best answer. These steps can be repeated literately which used to understand the user challenges and attempt to identify best solution or alternative strategies and solutions for the challenge. This will encourage questioning, assumptions and implications which extremely useful to clear understand of problems that are defined or unknown.

Design thinking also describes as an approach as well as methodology that can be used to resolve complex problems and find desirable solution for clients. (Razzouk & Shute, 2012) DT is analytic and creative process for experiment, create prototype, gather feedback and redesign. Visualization and the creativity is the major characteristics of design thinking. DT helps for designers systematically extract, teach, learn and apply human-centered techniques to resolve problems in creative and innovative way. Currently “Design Thinking” is identified as an exciting methodology for dealing with problems in many professions including engineering,

business etc. But most preferred for information technology (Brooks, 2010). According to Dorst (2011) design thinking is a new paradigm for dealing with problems in sectors such as IT, Engineering and Medicine.

There is no specific way to use design thinking for a specific matter. Further explained by Dorst (2011), people use design thinking in different ways. That is depend on the challenge that they are facing. Design Thinking generally defined as a visualization and creativity process that engage to person experiment, create prototype, gather feedback and redesign it acceptable manner.

According to Mintz (2017), Design Thinking is a broad picture which trying to be create. In order to do so, there are important concepts to be considered. First one is the concept to be addressed and this describes as empathies. Second one is the interactive thinking in order to bring more ideas also described as define. Third one is the process and the result. With Design Thinking, no matter how challenge the problem may be, there must be a best solution out of several alternatives and this is called ideate. Forth one is experiment to explore the result in a creative way. Prototypes can be used to describe those and present it to get feedback. Last one is the collaboration to test the solution. In order to get the best solution, designer must work with marketers, architects, engineers etc. and need to understand every aspect. Best use of design thinking, can get best out of several ideations. It has further been described by (Pourdehnad, Wexler, & Wilson, 2011), that design thinking generally applying designers sensibility and method to problem solving without mattering what is the problem is.

As mentioned by Mickahail (2015), DT is a business model to companies who seeking profitable innovative venture in the competitive market. While using existing approaches, many companies use DT in their product and service development to enhance the existing approaches (Frye & Inge, 2013).

### **2.1.1. Importance of Design Thinking**

DT is an analytical and systematic process that engage with experiments with creating prototypes, gathering feedbacks, and redesign to understand the real

problem of the customer (Razzouk & Shute, 2012). Moreover, DT is a process of thinking rather than the traditional thinking to develop new innovative solutions. It is always pushing you to discover undiscovered possibilities. Successful innovations are rather defined by the user's point of view than by technical perfection (Thienen, Noweski, Meinel, & Rauth, 2011). Design thinking can be considered as a toolkit for innovation. This is focus on the core problem. With the time there are many things has been changed. Customers are changed, life style has been changed and the way to communicate also changed. To success with the business, people should move with these changes and different skills are needed. DT is one of those skills (Razzouk & Shute, 2012). DT can be considered as a map who doesn't have any idea about the beginning and do the things.

Moreover, stated by Razzouk & Shute (2012), product design and competition is a major part of the a business competitiveness. Therefore, there should be a way to attention to this part of the business. DT probably integral part of design and engineering field. Because it involves for creative thinking and generate solution for the problems. Most of the time you might not realize it as DT. Often think that DT is used to problem solving. But it is more than problem solving. DT is a practical approach to develop solutions. The importance of design thinking is, focuses on end users. When getting collaborative ideas, it automatically builds the teamwork and increase the team spirit. Design thinking always consider about the human nature and the social needs. Since it is encouraging testing, possibility of failing solutions are very less.

DT is more important for requirement engineering (Vetterli, Brenner, et al., 2013). Requirement engineering is a process to gather information for developing information systems that the way of need for today's world. Today's world is changing very rapidly. Therefore, web apps, tools, and mobile apps should more closely fit to the customer needs. DT is an approach to drill down deeply about core issues. In depth analysis on a problem will provide a good understanding about the issue and can come up with strong relevant solution. DT is giving opportunities to understand the market. Understand the market need and develop a product to fulfil



those. With understanding the people, you always with real customer with real problems. Then you can create real solutions.

The aim of DT is fail early rather going forward and fail nearly at the time of product delivery (Vetterli, Brenner, et al., 2013). Therefore, most important aspect of DT is, it minimizes the later failures. Today most of the business are failing. DT encourage experimentation with prototype then getting feedback which allow you to fail fast. Learn from them and eliminate weakness to ultimate success.

### **2.1.2. Design Thinking in Software Development Industry**

With comparing other industries, IT industry dealing with technical product and services. IT industry is an innovative industry across the world. Highly trained professionals dealing with complex issues to solve them. Large number of people working for IT companies in every country. Therefore it is grooming day by day. Due to this situations, it is not easy to solve the problems in IT industry without an expert knowledge (Thienen et al., 2011). Most of the IT companies follow agile methodology as a project management technique. Agile use iterative incremental approach to design and develop projects. Normally developers assign tasks and those needs to be developed within few days. Most critical aspect is resolve complex problems within few days.

With comparing Agile with DT, it shows some parallel relationship like; user-centric, iterative process, iterative learning and development, and collaboration (Thienen et al., 2011). As explained by Thienen et al. (2011), DT already introduced to IT development. Further explained by Thienen et al. (2011), Agile tendency to avoid divergent thinking about the problems. But DT always provides divergent solutions for the problems. Most important thing of such cases requires intuitive thinking and understand the scenario. After getting the problem from client development team supposed to brainstorm ideas and suggests different solutions. It is important to understand the problem extract manner otherwise both solutions and the time spend for them get wasted. The correct solution is the client's experience. DT use in IT companies for brainstorm solutions for customer's problems.

DT is a exiting new paradigm to deal with problems specially Information Technology (Dorst, 2011). Information technology always focusing on exploring new ways to be the market leader or at least survive in the market. Therefore, employees in the IT industry should have an innovative mindset or need to change the mindset for such. User-centric design, user-experience design, and feedback based learning are highly important for IT engineers (Thienen et al., 2011). Therefore, DT is very useful to identify the user-centric needs based on applying different techniques. DT consists with initial techniques with requirement engineering with applying its iterative process to involving customer to agile development methods (Vetterli, et al., 2013).

## 2.1. Design Thinking Elements

Hassi & Laakso, (2011) has mentioned three dimensional framework for the different elements associated to DT. Table 2.1 mentioned three main categories **Practices, Cognitive Approach, and Mindset** associated with DT elements.

This research will focus on **Practices** category rather looking on Cognitive Approach and Mindset. These two most relevant to answering questions. By considering practices category, analyzing how individuals use this practices to deliver best product to the customer.

Table 2.1 DT Categories and Elements  
Source: (Hassi & Laakso, 2011)

Design Thinking Category	Design Thinking Elements
Practices	<ul style="list-style-type: none"> <li>• Human-Centered Approach (HCA)/Empathy</li> <li>• Think by doing / Action based</li> <li>• Visualizing</li> <li>• Synthesis of Diverging &amp; Converging</li> <li>• Collaborative work style</li> </ul>
Cognitive Approach	<ul style="list-style-type: none"> <li>• Holistic viewpoint</li> <li>• Integrative thinking</li> <li>• Abductive thinking</li> <li>• Reflective reframing</li> </ul>
Mindset	<ul style="list-style-type: none"> <li>• Future oriented</li> <li>• Experimental and explorative</li> <li>• Ambiguity tolerant</li> <li>• Optimistic</li> </ul>

### 2.2.1. Practices

There are five elements that needs to be considered as DT practices. Human-centered approach, thinking by doing, visualizing, synthesis of divergent and convergent approach, and collaborative at work (Hassi & Laakso, 2011).

- **Human-centered approach / Empathy**

With human-centered approach, first needs to be considered capabilities and the limitations of the human and whether the functions or tasks can be used by human. Human-centered approach is people based, user-centered, empathizing, ethnography and observation based approach to understand the customer or users. Empathizing and ethnography are the key elements for empathic understanding about the problem (Hassi & Laakso, 2011).

DT is always around people and studying what people expect, their needs and how they behave. This is really need to know much about the business. As mentioned by (Frye & Inge, 2013), DT has an aspect as a user centric philosophy.

Human-centered design can be done with understanding of customers. Understanding of what customer really needs can gain competitive advantage. Well established companies always seeking of new way to compete in the market. This can be minimizing uncertainties and risk innovation by engaging customers through prototypes and getting feedback.

Instead of focusing on a product or customer, design thinking sees things from customer perspective. DT mind set is solution focus instead of focusing problems.

Human and social factors are very important for developing software. Therefore, Phycology and computer science are growing with interdisciplinary relationship (Crawford, De La Barra, & Letelier, 2008). Software developed by people for people in the society, but most of the software engineers not considering the importance of human and social aspects. It has further been described by Crawford, it is important to consider new proposals for agile methodologies to focus on human and social factors.

Persona is a one of human-centered approach technique (Faily & Flechais, 2011). It describes the characteristics of the user such as age, location and education etc.

Scenario based design is the second technique of human-centered approach. It provides guidance of the functionality for development of the system which enabling user experience (Rosson & Carroll, 2002). It has further been described by Rosson, scenario based design focus on functional specifications and how people use the system to accomplish their work.

Third technique is the use-cases which is used to identify the usage of the system and provide description about how the system is used (Memmel, Gundelsweiler, & Reiterer, 2007).

- **Thinking by doing / Action based**

Thinking by doing is a rapid iterative development by early and fast prototyping while fast learning. Thinking by doing automatically creates prototypes that helps for ideation. Prototypes can be considered as a tool to develop and explore the idea before going with actual developments. It shows, how will be the end result and take feedback for adjustments (Hassi & Laakso, 2011).

It always about trying something to be developed and show. Making prototypes instead of real implementation. Go out there and test it with real users and show what will be the end product and getting feedback. The objective of thinking by doing is analysis and evaluate an issue in order to get judgement. With giving a prototype to a customer, improving it some more based on his feedbacks (Dunne & Martin, 2006)

- **Visualizing**

Visualizing is how to express ideas the way of understanding. It allows to common understanding of ideas that can be shared and discussed. It is a visual approach of a solution with visual thinking about the problem (Hassi & Laakso, 2011). According to Dunne & Martin (2006) visualizing and imagining something will take care of the user's needs. Because it is important to get the ideas about how it's looks like.

- **Synthesis of Divergent and Convergent approach**

DT typically starts with divergent approach to come up with multiple alternative ideas instead of taking initial idea as the best. Then the convergent approach used to identify patterns and relationships with moving towards the best solution by the selection (Hassi & Laakso, 2011). Most of the companies in all industries now looking for creating innovative product and services in order to compete in the market. Those companies always looking for alternative options to develop a desirable product (Frye & Inge, 2013). It always assumes that DT has a better solution can be found than what we have at the moment.

- **Collaborative work style**

DT emphasis collaborative working style. It involves wide range of stakeholders. Collaborative thinking is thinking outside of your own mind and thinking with group of people (Hassi & Laakso, 2011).

DT brings people with different capabilities into a one table. Getting feedback from all. Not only the mentioned people, constantly seek feedback from the public, large organizations, start-up and academic institutions.

Collaborative work style is used to understand the logical connection between ideas, identify mistakes, solve problems systematically, and to identify relevant and important ideas. According to (Pourdehnad et al., 2011) “Problems cannot be solved by the same level of thinking that created them.” Most of the problems encounter because of nature of change in social environments. At the same time, it will increase the complexity and uncertainty of the problems (Pourdehnad et al., 2011). The approach of DT is resolve complex problems from a variety of perspectives. They further explained the way that people think could help for innovative solutions.

Collaboration commonly involving for creative work and from that can be understand the interaction between an individual and the socio-cultural context (Crawford et al., 2008). There are many organizations who promoting DT culture. (Pourdehnad et al., 2011).

### **2.3. Design Thinking Vs Engineering Thinking**

Engineers always want to find the optimal solution to a problem. Normally engineers make judgements based on the experiments they did. Engineers always work with devices, components and systems. According to Waks, Trotskovsky, Sabag, & Hazzan (2011), engineers need both knowledge and experience to come up with optimal solution for a problem. Engineers solve problems by using mathematical knowledge and physical science. Therefore, gathering engineering knowledge is not very easy. It will take many years to mature.

However before developing a new product, engineers involving to a design process. According to Waks et al. (2011), engineering design is a systematic intelligent process that generate, evaluate and specify concepts for devices, components and systems. The objective is to satisfy a set of constraints. According to Waks et al. (2011), “understanding the nature of engineering design thinking is important for educators in all engineering areas.”

Engineering Thinking (ET) and DT share a similar process when it comes to the design stage. Both share same mindset for critical thinking to find the desired solutions in the design stage. However, ET use deductive methods while DT follows inductive reasoning while analyzing unpredictable information. DT principles are people, social needs, and user expectations which are unpredictable (Waks et al., 2011). Whereas ET focuses on predictable things rather than people and social needs. Both ET and DT use same approaches such as prototyping to express the ideas. However, ET deals with more certainty while DT details with uncertainty. Therefore, DT is more suite for human-centered problems solving rather ending with technical feasibility.

### **2.4. The Role of Customer Expectations**

Generally, software project success and the performance are determined by the target cost, time, expected quality and delivery of the product. Failures of the projects are determined by deliver the project on time and target budget but lesser functionality or develop desired functionality with high cost on time. As described by Agarwal &

Rathod (2006), this practice is difficult to determine that the project really success or not. As further described by Agarwal, there are two aspects to determine success criteria of a project. First one is the internal characteristics of the organization like time, cost and scope. Second one is the characteristics of the external parties like customer and the end users. First one is useful to control of the project. But second one is very important to deliverables to the users for leading to long-term impact. Moreover, that software development companies more internally focused. But both internal and external factors should be focus on to satisfy customer expectations.

Good customer experience is a way to have competitive advantage. To understand the customer experience, needs to be considered human and social factors. Customer perception is the main concern when considering the customer experience. Using agile main expectation is to satisfy customer. According to Dybå, T., & Dingsøy, T. (2008), the main aspect is the customer must practically involve to the process.

## **2.5. Agile**

Agile origins in the manufacturing industry and manufacturer had flexibility to deliver the product with continues modifications(Frye & Inge, 2013). This needed to respond the changes in the software projects in the software development industry.

### **2.5.1. Agile software development principles**

Agile software development manifesto developed by based on following principles (Frye & Inge, 2013).

- “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. “
- “Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”

- “Business people and developers must work together daily throughout the project.”
- “Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”
- “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”
- “Working software is the primary measure of progress.”
- “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”
- “Continuous attention to technical excellence and good design enhances agility.”
- “Simplicity the art of maximizing the amount of work not done is essential.”
- “The best architectures, requirements, and designs emerge from self-organizing teams.”
- “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

### **2.5.2. Agile software development methods**

Within the agile software development approach, several methods of agile such as Scrum, Extreme Programming (XP) and Test Driven Development (TDD) make the software development process flexible and responsive (Frye & Inge, 2013).

#### **Scrum**

It has further been described by Frye & Inge, the agile scrum is a process perspective framework that is mainly used or software development process to provide iterative or incremental delivery of the product to the customer. According to Begel &



Nagappan (2007), scrum is the most popular agile practice among other various agile practices. According to the study of Azizyan, Magarian, & Kajko-Mattson (2011), 54% companies use scrum and 32% use xp.

Agile methods are effective for projects which are large, complex ensure success with changing requirements. According to Coram & Bohner (2005), using agile, project managers can clearly understand project progress and performance.

“Scrum is also not a complete methodology since it does not include specific practices” (Frye & Inge, 2013). It is used to optimize the software development process and make it smooth. There are two main roles in scrum process. One is product owner who responsible for the functionality of the product and other one is scrum master who concerned with execution of the project smoothly. Software development process is more complex and filled with uncertainty. Therefore, there should be a method to encourage this process.

According to Mundra, Misra, & Dhawale (2013), scrum team structure (4-10 members) is a one successful factor. Because team work is very critical to success of scrum project management. Accordingly, small teams perform better and team members follow each other.

## **2.6. Synthesis of Design Thinking and Scrum**

Scrum is an agile method to develop software product been flexible and responsive. It focused on the priorities and the value of the customer and what to be released iteratively. However, DT is a process focus on to find out what customer really need, and customer satisfaction (Razzouk & Shute, 2012). Scrum is mainly focus on the customer rather than the expectation of end user. There is no step in the scrum method for end user testing. But getting the customer feedback based on the demonstrations and adjust priorities of the next phase developments.

Collaboration DT with Scrum methods can be achieve the desired outcome. This make the self-organizing teams with bringing necessary skills and values to the project.

DT elements visualize through story board. In scrum, the written requirements and specifications are in the product backlog. If DT integrated with software development, the story board could be considered as the foundation for the product backlog.

DT is an action based approach. It exposes the ideas through prototyping. With this prototyping approach can learn and fail quickly and cheaply (Vetterli, Brenner, et al., 2013). It is exposing to Scrum methods to learning constantly and continuously getting feedback for early failure or success.

## **2.7. IT Maturity in Sri Lanka**

In 2013 and 2014, Sri Lanka was awarded as the offshoring destination of the year award by the National Outsourcing Association, UK (SLASSCOM, 2018). According to the Sri Lankan IT/BPM Industry 2014 Review report (SLASSCOM, 2018), Sri Lankan IT/BPM sector showed an impressive growth trend of 238% from 2007-2014, which was primarily contributed by the IT service industry. Therefore, Sri Lanka IT services industry is quite mature.

## **2.8. Summary**

Agile is the most popular software development methodology which achieve higher success rate than traditional software development methods. It has enhanced the fast delivery of the software product with increasing customer collaboration. Since it is trying to achieve fast deliveries with minimum requirement, there is a question it is delivering the right solution to the customer as early as possible.

DT come up with innovative way to understand the real problem and give a best possible solution for it. It is a best practice for business to solve problems for developing new products. DT is a systematic process rather than traditional thinking focus for innovation. DT is not a new concept to the world. But, it is new to the software development industry. According to the framework identified by Hassi & Laakso (2011), eleven elements have been categorized to three categories.

There are few differences with DT and engineering thinking because Engineers always take judgement based on the experiments. However before developing a product, they also involving a design process. Therefore, they also using DT for their design process.

Customer expectation is a critical aspect of a software development product. At the end win or loss depend on the delivered product. If the functionality and experience are not available as expected level, project will be failed. Agile is a software development method which already achieved higher success rate with traditional methods. Since Agile trying to achieve fast deliveries, it is difficult to guarantee it is developing best possible solution to the customer. Therefore, in this literature tried to identify the elements which can apply for software development process with agile project management to provide customer expected solution.

### **3. RESEARCH METHODOLOGY**

This chapter describes the steps, process, and procedures practiced during the data gathering, interviews, and analysis of the research. Related work is used to gather the factors and identify the research gap, which was essential in formulating the research methodology. Chapter is breakdown as follow. Section 3.1 provides the research approach and method. Section 3.2 illustrates the population and sample selection, while Section 3.3 explains the data collection. Section 3.4 describes the process of data collection. Section 3.5 elaborate the use of grounded theory in this research and how theory is generated based on the grounded theory, respectively.

#### **3.1. Research Approach and Method**

I focus of this study to measure the usage of DT and understand the best practices in an agile environment. Figure 3.1 illustrates the conceptual framework for the research. DT can be categorized as practice, cognitive approach, and mindset as seen in Table 3.1. As the first step, this study focused only on Identify the use of design thinking practices within agile practices.

Deductive reasoning method is more suitable for quantitative research. However, quantitative approach is not suitable for this research because it is difficult to quantify the usage of DT practices and the benefits that gain from it. And other reason is awareness of DT techniques of employees were not through enough. First needs to be explained the theory and the usage of those techniques then needs to be get the output from them. As this is not quantifiable, this study selected qualitative approach and inductive reasoning method.

Conducting interviews to get their ideas and opinions as a case and analyze findings and recommendations. Qualitative research approach has been followed with conduction semi structured interviews created based on the conceptual framework. Design thinking is relatively new for the software development industry and approach focus to understand how agile teams applying design thinking practices to identify the correct problem and implement the customer expected solution.

Meanwhile questions were designed to identify which agile methodology that teams are using and the awareness of the techniques of each and every design thinking practice as well as to identify the ways that can satisfy customer.

First conducted pilot interviews for using three people and re-organized the interview questions based on several factors. Pilot interviews are important to understand the interview questions are will in the real world. Basically how the interviewees are answering to the questions, can interviewees understand the questions, any repetitive questions, same answer giving for several questions, verify the validity of the answers, and the time it's taking to conduct the interview. Three project stakeholders were chosen for pilot interviews in three categories; Project Manager, Business Analysis, and a Tech Lead in two different companies. Mentioned project stakeholders are main contributors for requirement gathering, managing project, and take technical decisions.

After revising the interview questions, conducted the interviews for 10 software service based companies including Project Managers, Business Analysis, Architects, and Tech Leads. I didn't consider Associate software engineers, Software engineers, or QA personals as interviewees because they are not involving at the time of requirement gathering, designing, and finalize requirements with decision making.

To conduct this study, selected only software service based companies. Because they are the ones who develop customized software products as requested by customers. This kind of projects are customer centric projects which trying to achieve requirement of a specific customer. Product based software development companies were not selected for this study. Because, those products are not targeting one customer. They are targeting a customer segments. Those products can't be customized for a specific customer. Customers' needs to be adjust their processes according to the product. Therefore, there is no user centric development on such products.

Figure 3.1 illustrates the research methodology that used for this study.

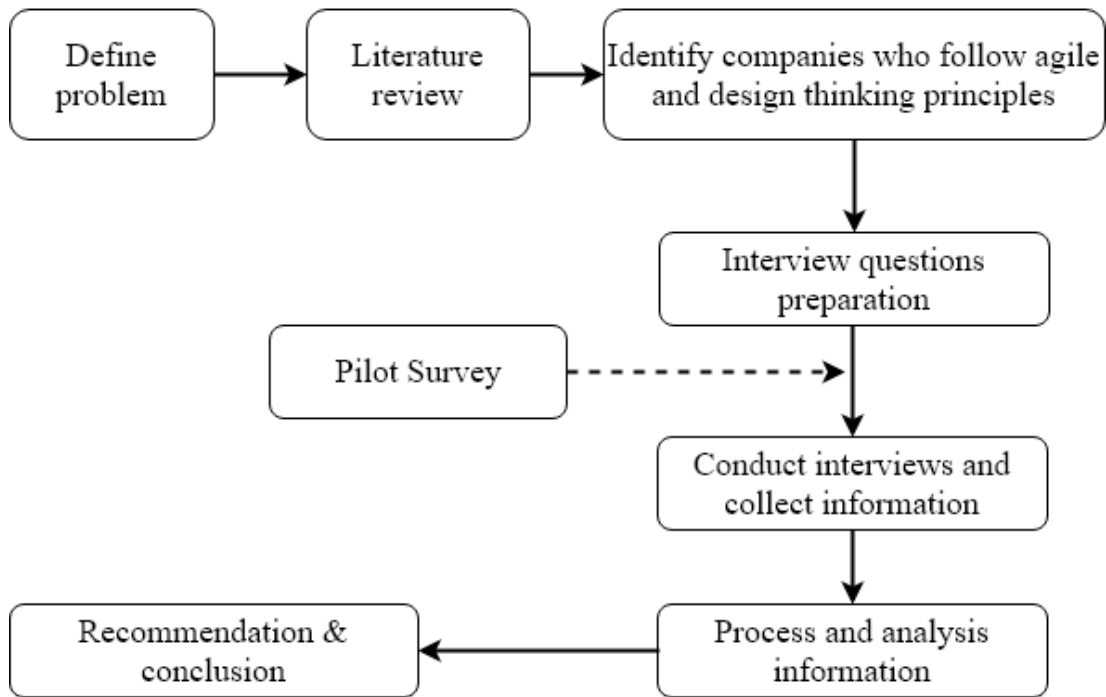


Figure 3.1 Research Methodology

Based on the research idea an extensive literature review and a set of pilot interviews were conducted to identify the problem statement of the research. Based on the pilot interviews and literature researcher identified the way needs to be organize the questions to get the outcome. Then a set of interview questions were derived based on the key points needs to be highlighted. Collected interview data was analyzed using the grounded theory approach and preparing a list of observations, findings and recommendations.

### **Conceptual / Theoretical framework**

The conceptual / theoretical framework describes the interrelationship among the independent variables to measure the dependent variable. The proposed conceptual framework illustrated in Figure 3.2.

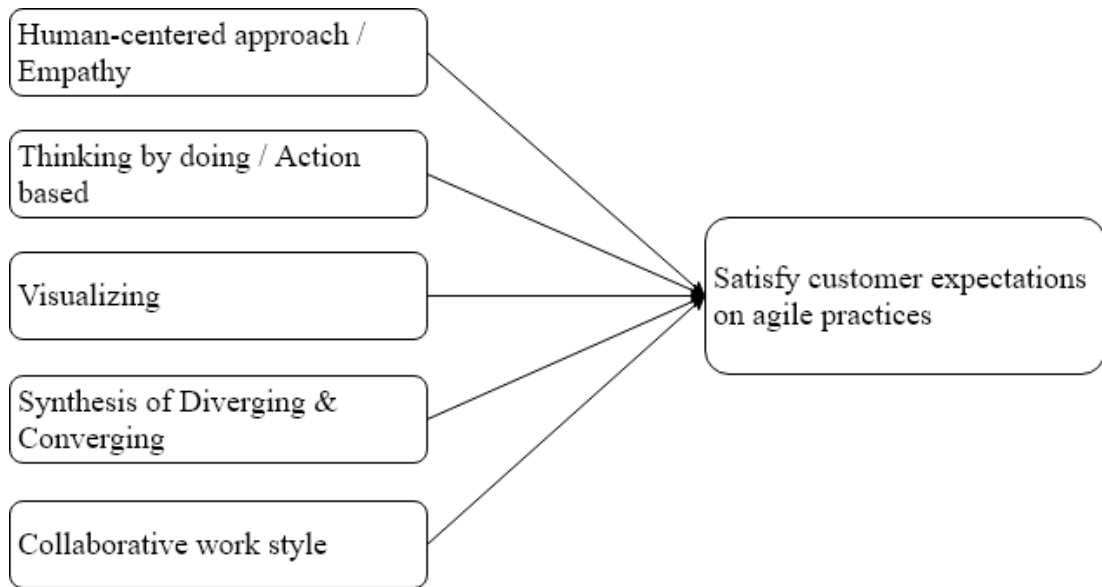


Figure 3.2 Conceptual framework

Table 3.1 Design Thinking Categories and Elements

Source: (Hassi & Laakso, 2011)

<b>Design Thinking Category</b>	<b>Design Thinking Elements</b>
Practices	<ul style="list-style-type: none"> <li>• Human-Centered Approach (HCA)/Empathy</li> <li>• Think by doing / Action based</li> <li>• Visualizing</li> <li>• Synthesis of Diverging &amp; Converging</li> <li>• Collaborative work style</li> </ul>
Cognitive Approach	<ul style="list-style-type: none"> <li>• Holistic viewpoint</li> <li>• Integrative thinking</li> <li>• Abductive thinking</li> <li>• Reflective reframing</li> </ul>
Mindset	<ul style="list-style-type: none"> <li>• Future oriented</li> <li>• Experimental and explorative</li> <li>• Ambiguity tolerant</li> <li>• Optimistic</li> </ul>

### 3.2. Population and Sample Selection

The population of this study is IT service organizations in Sri Lanka. While 77 organizations are listed in the SLASSCOM website [Accessed 01 02 2018], only 50 organizations were determined to be into IT services according to their LinkedIn profiles (see Figure 3.3). Some organizations were no longer in business and some are in telecommunication related business. From the 50 organizations, we selected 26 with more than 50 employees. It is unlikely that an organization with limited resources to practicing DT techniques, especially at startup stage. From the 26 organizations ten were identified to be using the snowballing method which is used with DT to execute agile-based projects (see Figure 3.3).

The objective of sampling is a process of data collection for generating theory based on the collected data, then analyze the data and decide what data to collect next and where to find them. Set of pilot interviews conducted with using three employees in three different companies and identified the way they answer to the questions and any repetitive answers are getting. After the initial interviews the collected data were coded based on the grounded theory and analyzed to guide the interview process.

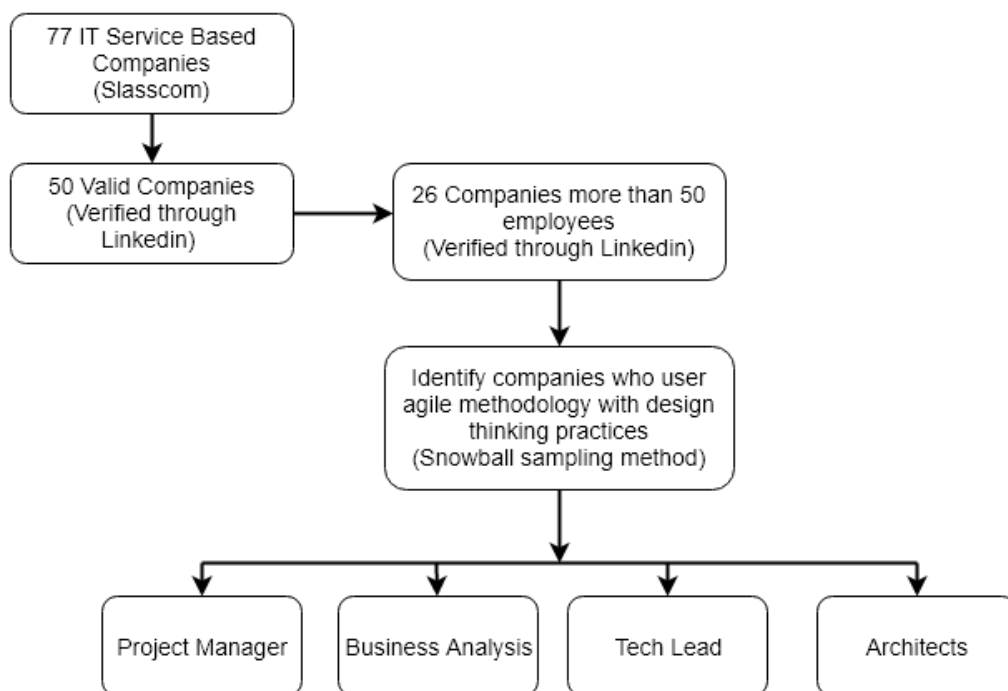


Figure 3.3 Population and sampling



## **Snowball Sampling**

Snowball sampling is a technique to recruit participants for a research or study. This is used where potential participants are hard to find. When rolling a ball, it will collect more snow and it becomes larger. This is a non-probability sampling method and it says simple random sampling.

### **3.3. Data Collection**

I conducted the data collection using a set of face-to-face interviews with project managers, business analysis, and leads from those ten organizations. These stakeholders are the main contributors for requirement gathering, project handling, and take technical decisions.

Face-to-face interview questionnaire were derived from the literature survey. While DT is categorized into practices, cognitive approach, and mindset based on Table 3.1, I focused only on DT practices. There are five elements in the DT practices, namely human-centered / empathy, thinking by doing / action based, visualizing, synthesis of diverging and converging, and collaboration. Interview questions were derived along these five elements. A set of pilot interviews were conducted after selecting three project stakeholders (project manager, business analysis, and tech lead) from the industry to validate the questionnaire. Though project stakeholders have used the DT techniques, they were not familiar with the word “design thinking”. Hence, based on the feedback received, questions were revised. The answers were tracked in an excel sheet in separate manner and summarized them together to apply grounded theory.

### **3.4. Process of Data Collection**

I interviewed fifteen managerial and lead level employees from ten service-based software development companies that applied DT practices into the agile process. These project stakeholders are the key people who take decisions and negotiate with the customer. During interviews I had to explain about DT theory and concepts to the

interviewees, as most of them were not aware about the concepts though they practice those. But they are using DT techniques for their day today work.

I analyzed the data using Straussian grounded theory approach to give meaning to develop a theory. Immediately after each interview we started the open coding process, analyzing textual contents, and break the data into parts and identified the key terms and compared them to identify the categories. Open coding uses words, statements, and phrases from data collected to develop concepts. These indicators are constantly compared with data being collected to identify new insights until theoretical saturation is reached. Using axial coding, we then identified disaggregation and relationships of the open coding. Finally, using selective coding we identified the variable factors.

### **3.5. Grounded theory approach**

Glaser & Strauss, (1967) grounded theory was used to analyze unstructured data to bring meaning and ultimately develop a theory. This section describes how the data were analyzed based on the grounded theory approach's stages such as open coding, axial coding, and selected coding, based on the memo writing to generate theory.

#### **Coding**

According (Strauss & Corbin, 2008) to the researcher should begin with open coding after immediately conducting an interview. *Open coding* is the process of analyzing textual contents and breaking the data into parts and identifying the key terms and comparing the key terms to identify the categories. Open coding uses words, statements, and phrases from data collected to develop concepts. These indicators are constantly compared with data being collected to identify new insights until theoretical saturation is reached (Feeler, 2012).

Axial coding consists of identifying relationships among the open codes. What are the connections among the codes?

Selective coding figures out the core variable that includes all of the data. Then reread the transcripts and selectively code any data that relates to the core variable you identified.

### **Category and memo writing**

Performing open coding for all the interview responses may result in many pages of codes as well as duplicates in coding. Therefore, researcher must analyze and identify those and find the similarities and omit the duplicate codes and group them into categories or subcategories and link them to categories. In this study several categories are identified from the literature and pilot interviews and open coded data is assigned into subcategories and categories from the data collected from the interview questions.

Memo writing is a pivotal aspect of open coding because open coding with a few words is not enough to describe the concept often. The notes that are written based on the open coding and the analysis of gathered data is called a *memo*. Writing memos support the researcher to think more abstractly and theoretically to bring more depth into study. A memo contains a set of sentences, paragraph, or even more if needed.

### **3.6. Summary**

Choose to use a qualitative analysis based on interviews to gather data and identify the advantages that software development teams gaining by using DT practices. This study follows a Straussian grounded-theory design to guide the collection and coding of interview data to identify emerging categories and to generate critical factors to be considered. Related work was utilized to identify the direction of the research, identify variables, and form the questionnaire for interviews. Both related work and researchers own observations of the pilot interviews were used to identify prefigured categories. Set of interview questions was then derived based on the prefigured categories. Then a set of service based software development companies from different senior level people was selected for interviewing. Data gathered from the

interviews were analyzed by coding, categorizing, comparing, and memo writing of the interview responses. Key findings, observations, and recommendations are then derived through data analysis.

## **4. DATA ANALYSIS**

This chapter presents the analysis of the collected data through 10 service based software development companies. A qualitative approach was used and data analysis was conducted based on the Straussian's Grounded theory approach. Rest of the chapter is organized as follows. Section 4.1 describes how the collected data were prepared for analysis using the Grounded theory. Section 4.2 presents the data analysis where it is broken down into several sub-sections where we analyze the interview data based on Human-centered approach Section 4.2.1 thinking by doing approach Section 4.2.2, visualizing the idea as tangible Section 4.2.3, synthesis of divergent and convergent approach Section 4.2.4, collaborative work style Section 4.2.5. Summary of data analysis is presented in Section 4.3.

### **4.1. Data Preparation for Analysis**

All the data collected conducting 15 interviews through 10 service based software development companies and 1-3 people interview from each company which including Project Managers, Business Analysis, Tech Leads and Architects. Conducted semi-structured face-to-face interviews and questions were based on the conceptual framework.

I applied Straussian's Grounded theory for data analysis. Using open coding, I categories the answers into different sections and using axial coding I identified disaggregation and relationships of the open coding. Finally, using selective coding, I identified the factors.

### **4.2. Data Analysis**

In this research, the data analysis is based on the identified conceptual framework. As per the analysis, research tried to identify the role of the interviewee, the agile methodology that they are using for their projects, the pain-points of the customers that they are worrying mostly from the point of view of interviewees, and advantages and best practices of having DT practices in agile practices.

Table 4.1 Profiles of interviewed

<b>Designation</b>	<b>Responsibilities</b>	<b>Experience with Agile projects</b>	<b># Agile projects engage with</b>
Senior Project Manager	Scrum master, Project Management, Making processes	6+	3
Technical Specialist	Team Leader, Team Player, Confident	6+	5
Business Analysis	Scrum Master, Product Owner, BA, Client management, Agile coach	4+	3
Technical Project Manager	Budgeting, requirement gathering, Resource planning, Setting up project, Document repositories, Daily scrums, Estimations, dev ops and deliverables.	10+	12
Tech Lead	Leading project, Mentoring Developers, Developments	6+	8
Director / Mobile Architect	Architectural Design, Quality maintain, Facilitating, Decision making	10+	10+
Senior Business Analysis	Interact with customer, Understand requirements and their needs, Develop business case, Observations, Do workshops, Take lead role to ensure communicated with dev, Requirement signoff	7+	10
Tech Lead	Responsible for all technical inputs, Play Architect role some situations, Take requiring not waiting BA release all, Work on user stories, Priorities and responsible for estimations, Technical designs, Creating tasks and assign, Code review	8+	11
Tech Lead	Team lead, Development, Responsible for deliverables Support work related to server and application performance, Incident management, Task management, Estimation, Analysis and recommendations.	6+	3
Tech Lead	Lead developer, Deployments, Scrum master, Estimates	3+	3
Project Manager	Ownership of the project, Analysis requirement with BA and Architect, Identify risk and mitigation plans, Estimations, Delivery quality check, Check progress and provide updates	5+	6
Project Manager	Scrum master, Project Manager	4+	5
Tech Lead	Team Lead, Requirement gathering and analysis, Design the implementation, Development	4+	5
Tech Lead	Team Lead, Customer communication, Design decision, Technical decisions, Scrum process, Requirement analysis	3+	4
Project Manager	Project manager, Scrum master	7	3

As per the analysis all the interviews are plying leading, managerial and decision making roles. The roles and responsibilities are Scrum master, Project Management, Processes creation, Team Leader, Team Player, Delivery Management, Mentoring Developers, Developments, Architectural Design, Quality Maintain, Facilitating and Decision Making.

This study used interviews to gather data and grounded theory was used to analyze the data to understand the use of design thinking practices within agile practices of service based software development companies. Table 4.1 lists the summary of interviewed employee profiles, responsibility, and experience.

As seen in Figure 4.1 Scrum is the most used (71%) agile practice. Kanban, XP, and Lean are the other agile practices.

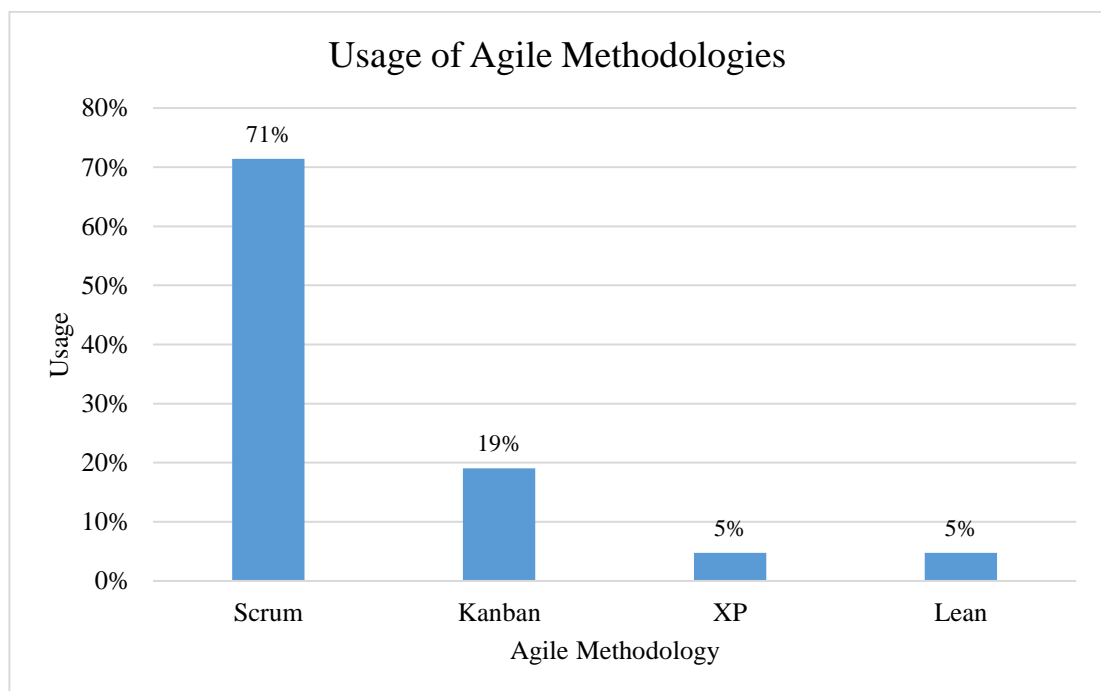


Figure 4.1 Usage of agile methodologies

During the data analysis, I identified customer pain points and ways to improve customer expectations, and those are listed in Table 4.2. As per the analysis of the interview data based on the Straussian's Grounded theory, there are several points' needs to be considered to satisfy customers. The mentioned pain-points are that

customers are worrying about the software projects from the point of the view of interviewees. There are few aspects identified based on the grounded theory.

Table 4.2 Pain points in customer satisfaction and ways to improve

Selective Coding	Axial Coding	Open Coding
Project success and customer satisfaction	Pain points in satisfying customer expectations	<ul style="list-style-type: none"> <li>• Unable to prioritize what customer really wants</li> <li>• Compare and update the product features against competitor's product without considering its applicability</li> <li>• Not focus on feature applicability for tomorrow market</li> <li>• Dealing with fix bid projects (budget, cost and time are fixed)</li> <li>• After completing the implementation only user realize feature is useless.</li> <li>• Requirement changes in a limited budget.</li> <li>• Dealing with very new technology.</li> <li>• Product develop without realizing target market.</li> <li>• Customer always expect to work only with domain experts and subject matter experts.</li> <li>• Micro manages by the customer</li> <li>• Lack of trust on vendor</li> </ul>
	Techniques to understand customer expectation	<ul style="list-style-type: none"> <li>• Conduct workshops at customer end to understand customer and end-user wish list</li> <li>• Develop roadmaps for implementation</li> <li>• Conduct reviews and demos (proof of concepts, prototypes)</li> <li>• Conduct frequent formal and casual discussions</li> <li>• Hire an end user to understand his/her pain points.</li> <li>• Gather and analyses customer pinpoints</li> <li>• Study the customer during pre-sales</li> <li>• Deliver small frequent deliverables with expected features</li> <li>• Communicate the blockers early as possible</li> <li>• Conduct market validation before starting product implementations</li> <li>• Educate client on simplicity on product implementation</li> <li>• Derive business case and form the foundation to the solution</li> <li>• Identify the non-functional requirements</li> <li>• Keep update customer about product implementation progress (Ex. Burn down charts, show and tells)</li> <li>• Study the drawbacks of the existing process</li> </ul>

First one is the pain points in satisfying customer expectations; competitive advantage, user experience of the product, expected outcome of the project, beyond the technology business context, and project team building the right product. As stated by a Technical Specialist in a mid-sized company:



*“End of the day win or lose depend on the customer experience”*

Furthermore, stated by a Tech Lead in a mid-sized software development company:

*“Customer thinks that we are experts in the domain but dev team doesn’t have any knowledge about it”*

This is a critical expectation from the project team by the customer. Otherwise it is difficult to collect the right requirement with asking right questions.

Second one is the techniques to understand customer expectation; Conduct workshops at customer end to understand customer and end-user wish list, Conduct reviews and demos (proof of concepts, prototypes), Conduct frequent formal and casual discussions, hire an end user to understand his/her pain points. As stated by a senior project manager:

*“Visit end-users working places to understand what they are really doing”*

Based on the analysis, trying to explain how the DT practices helps to get rid of these pain-points to satisfy customer.

#### **4.2.1. Human-centered approach (HCA)**

As per the analysis of the interview data based on the grounded theory, HCA is important to identify customer’s real needs. Identifying customer real needs are the biggest challenge that software development teams face during the project. As the project scope, timeframe, and cost are considered as project management principles the customer’s needs and level of customer expectation differs.

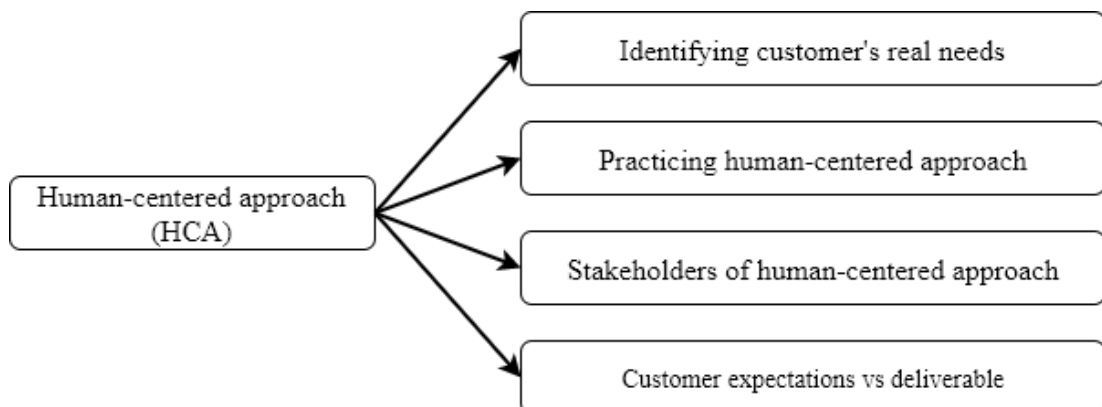


Figure 4.2 Dimensions related to human-centered approach

### **Identify customer's real needs**

As per the findings from interviews, customer real needs can be identified such; Customer Journey Mappings, Prototypes, Proof of concepts (POC), Story Mapping, Use-Cases, Heat map, Customer Profiling and using Scenarios. As stated by a project manager in a mid-size software development company:

*“Customer journey is used to understand how user going use the system, experience with the system and clarifications for gray areas”*

Moreover, a Technical Specialized from mid-size company further explains:

*“Drill down the stories to find the requirement. Discuss what exactly needs. What customer needs and what you feel that customer needs.”*

As a conclusion Customer Journey, Story mapping for requirement, Prototype, POC, UX, and Scenario based for understanding requirement are the most suitable approaches to identify customer real needs in agile software projects.

However, even though it is being identified that general techniques used to requirement gathering has some drawbacks still some of them are use. Some of the traditional methods such as only getting requirements from BA are not recommended by many of the interviewees. As stated by a Technical Specialist of the mid-sized company:

*“Traditionally BA give requirement. But needs to be identified what is there, what needs to be done and business problem”*

Moreover, stated by a Tech Lead in a mid-sized company:

*“Only getting requirement through PO can have communication gap between PO and end-user, and it badly effect to the development team”*

### **Practicing human-centered approach**

As per the analysis, there are several ways to practicing HCA with agile such; Workshops and discussion, PO communicate with end-user and come back to the project team, BA explains current understanding and identify the gap, and end-user interaction through UAT. As stated by Tech Lead of a mid-sized company:

*“Every week discussions to make sure both client and development team are in the same page”*

Moreover, a PM of a mid-sized company further explains:

*“Do workshops and interactive sessions and discussions with the customer to build up the customer’s vision”*

### **Customer expectation and the deliverables**

If the project team has capability of deliver the customer expected solution that can make customer happy. The ultimate goal of the HCA is full-fill customer expectations with deliverables. Based on the analysis, there are many situations that can use HCA to identify the right problem. If the project team can identify the right problem, then only can deliver the right solution to the customer. The ways of HCA usages are; fine tune the collected requirements, identify the right technology for the requirement, understand what customer expecting to be implemented, understand the customer’s business objective and what he is going to be achieved, and sign-off the features with the clients. As stated by a Senior Project Manager of a mid-sized company:

*“It is important to fine tune the requirement to understand the user experience. How user will be interacted with the system. System should make his life easy”*

Moreover, a Tech Lead of a mid-sized company stated:

*“This is a good practice to understand the what customer is expecting and experiencing”*

As a summary, HCA is very important to understand the real problem of the customer with use of HCA techniques. Based on the techniques can identify what are the problems that customer experiencing, what the customer needs to be experience, goals needs to be achieved, and project team can deliver right solution to make the customer happy.

Table 4.3 Open and Axial coding for human-centered approach

Selective Coding	Axial Coding	Open Coding
Human-centered approach	Identifying customer's real needs	<ul style="list-style-type: none"> <li>• Most of the time BA gives requirement and he clarified with the customer and the end-user</li> <li>• PO give the requirement and he clarify those with end-users.</li> <li>• Customer journey to understand how user going use the system, experience and for the clarifications.</li> <li>• Prototype, POC, Wireframes, and UX to fine tune the requirement.</li> <li>• Story mapping for requirement defined in the backlog.</li> <li>• Use-cases for understand the system and what is it about and how user interact with the system.</li> <li>• Scenario based for understanding requirement based on different scenarios.</li> <li>• Build up minimal viable product.</li> <li>• Presentation to get feedbacks.</li> <li>• Customer profiling based on the background check. It is easy to understand what suite to the customer.</li> <li>• Workflows to identify how the system flow is going.</li> <li>• Heat map is something like mouse click points which is most impotent to increase customer experience.</li> <li>• A/B testing put two interfaces to customer testing to choose a one interface.</li> <li>• Persona techniques to identify user characteristics.</li> <li>• Prefer to use the customer journey mapping more than the scenario based approach.</li> </ul>
	Practicing human-centered approach	<ul style="list-style-type: none"> <li>• Workshops with discussions.</li> <li>• Product Owner communication with end users to get their feedbacks.</li> <li>• BA explains current understanding.</li> <li>• PO explain the requirement that he expected.</li> <li>• Convert high level vision into operations.</li> <li>• End users integration through UAT.</li> </ul>
	Stakeholders of human-centered approach	<ul style="list-style-type: none"> <li>• Stakeholders of the project such as Business Owner, Product Owner, Scrum Master, Business Analysis, Project Team, End-user representatives from the customer's side.</li> <li>• Customer's technical people to communicate with the development team.</li> </ul>

	Customer expectations vs deliverable	<ul style="list-style-type: none"> <li>• Fine-tune, Clarify, and finalize the collected requirements and wider aspects of the requirements.</li> <li>• Understand user experience of the domain and end-user background and what he really need.</li> <li>• Identify the appropriate technologies for the requirement.</li> <li>• Task prioritization must have and nice to have features.</li> <li>• Understand the customer's business objective and what customer is going to be achieved.</li> <li>• Sign off features with the client.</li> <li>• Can understand customer expectations and can ask right question from right people.</li> <li>• Help to understand all the impacted area of the application.</li> <li>• Reduce rework because of different understanding about the requirement.</li> <li>• Avoid unrealistic estimations.</li> <li>• Understand how user interaction with the system.</li> <li>• Provide very little things as well to satisfy customer.</li> <li>• Reduce the communication gap between intermediaries (PO, BA, and End-user).</li> </ul>
--	--------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 4.2.2. Thinking by doing / Action based

According to the study done by the grounded theory, thinking by doing is an action based thinking which is important to express the idea as tangible. Thinking solutions are very challenging thing in specially software development. Because it is important to think innovative ideas with the competitive world instead of providing general solutions.

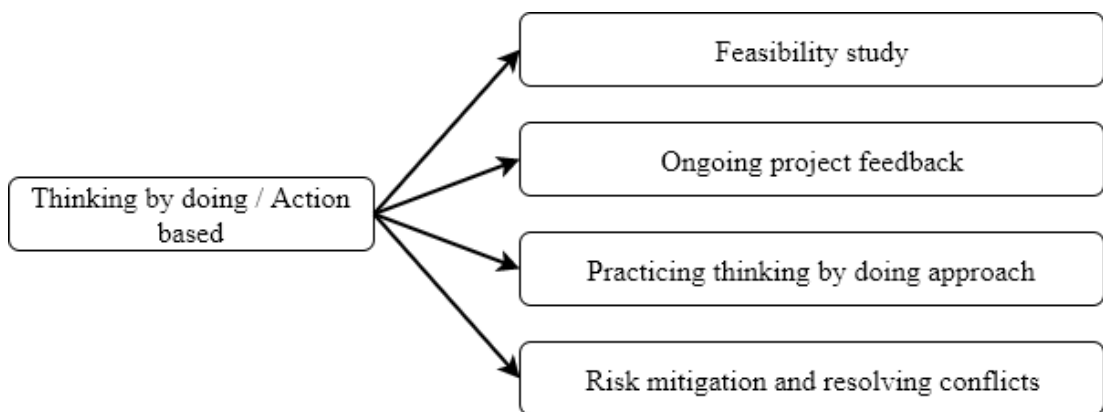


Figure 4.3 Dimensions related to thinking by doing

#### Feasibility study

As per the finding from the interviewees, software engineers intend to create proof of concepts (POC), prototypes and wireframes as feasibility study to determine the idea

can be implemented. As stated by a tech lead in a mid-size software development company:

*“Creating prototype and do POCs before going to actual implementation. Because the customer wants to know what will he getting in 6 months”*

Moreover, a Technical Specialized from mid-size company further explains:

*“Do POCs to it can be done what we suggested to the customer and with prototypes can learn something both project team and the customer, can take decisions in short frame of time to undertake whether it is good enough”*

Therefore, thinking by doing is a kind of feasibility study which make the idea as tangible and confident about the idea is possible to implement.

### **Ongoing project feedbacks**

As per the study, this approach is important to get ongoing project feedback which is important to drive the project right direction. As mentioned by engineers, creating wireframes to user interfaces and viewpoints of the application, with using prototypes with adjustments, direct the requirement to right direction. As stated by a senior project manager in a mid-sized company:

*“You can’t get very clear requirements from the client. Therefore, there should be a way to make sure the given requirement is correct”*

He further explains the importance about the feedback before going for actual implementation as:

*“We get the more confident that we are developing weather it is the right feature and the right way to develop it. Then it will reduce re-work and deliver the features fast”*

### **Practicing thinking by doing approach**

As per the analysis, there are several ways to practicing thinking by doing approach for; Show outcome and clean path about what needs to be done, Early failures rather than going forward and fails, make easy for communication, and Avoid misguiding requirements cause to project go wrong. As stated by Tech Lead of a mid-sized company:

*“With prototypes, reduce the risk factor, reduce the buffer time. Customer takes the responsibility after he say yes to the prototype”*

Moreover, explained by a BA in the same company:

*“Prototypes, make satisfy customer what the implementation is going to be implemented. Customer is confident what we can understand what they really want.”*

Confident of the implementation going to be implemented is very important to satisfy customer. Based on the study, with using this approach, the project team can increase the confident level of the customer before deliver the actual product.

### **Risk mitigation and resolving conflicts**

As per the answers given by the interviewees, risk mitigation and resolve conflicts are the biggest benefit that can take by using this DT practice. If the project team can mitigate the risk with resolving conflicts, project can drive smoothly. As per the analysis reduce the risk such as; Resolve conflicts with the requirement and the understanding gray areas, develop right feature at once without going with scrum iterative by iterative, resolve technical challenges to make sure it can be implemented and identify limitations in very early stage, customer expectations before going for actual implementation, and set responsibility to the customer for the given requirement.

As stated by Senior BA of a mid-sized company:

*“Customers mindset they learn us from our solution. After that they get better understand what they need. Then ask for changes.”*

As described by findings of this study, by using thinking by doing approach, the project drives to the right direction with minimizing risk of delivering the wrong feature iterative by iterative in scrum project management. With spending one or two iterations, can be deliver the right functionality to the customer with increasing his satisfaction of the final product.

Table 4.4 Open and Axial coding for thinking by doing DT practice

Selective Coding	Axial Coding	Open Coding
Thinking by doing / Action based	Feasibility study	<ul style="list-style-type: none"> <li>• POC to understand the feasibility, limitations, effort, performance, technology and make sure that can be implemented.</li> <li>• Prototype to show what we understood and what is going to be implemented.</li> <li>• Wireframe and process diagrams give better outcome.</li> <li>• Mockups for understanding screen functionality.</li> <li>• Taking feedbacks based on the POC, Prototype and mockups to finalize the requirement.</li> <li>• Some cases without going for POC or Prototype can give a best solution.</li> </ul>
	Ongoing project feedback	<ul style="list-style-type: none"> <li>• Wireframes to show the user interface and viewpoints to take customer's feedback.</li> <li>• PSD to show how the UI colors etc.</li> <li>• Mockups to describe the user interface very detail.</li> <li>• Prototypes make adjustments and direct to the right direction and understand the functionality of unknown areas.</li> <li>• Prototypes to understand how to do the implementation correctly on unknown areas.</li> <li>• Accurate estimations.</li> <li>• To test different approaches.</li> <li>• Customer engagement through Prototypes and POCs.</li> </ul>
	Practicing thinking by doing approach	<ul style="list-style-type: none"> <li>• Show outcome and clean path about what needs to be done.</li> <li>• Risk mitigation with avoid misguiding requirements to project go wrong.</li> <li>• Easy to make decisions.</li> <li>• Get client's sign off.</li> <li>• Understand real need at the early stage and feasibility of the requirement.</li> <li>• Early failures rather than going forward and fails.</li> <li>• Can resolve a lot of difficulties.</li> <li>• POC get to understand what will be delivered by us.</li> <li>• Convince the client to select us.</li> <li>• Make easy for communication.</li> <li>• Prototypes helps to get the project from the sales pipeline.</li> <li>• More paths to identify different options.</li> <li>• When developers are not experts on that area.</li> <li>• When we don't have clear picture what to be implemented.</li> <li>• To take a technical decisions.</li> <li>• Understand complex UI functionalities by using wireframes and mockups</li> <li>• Identify limitations.</li> </ul>



	Risk mitigation and resolving conflicts	<ul style="list-style-type: none"> <li>• Resolve conflicts with the requirement and the understanding ambiguous and gray areas.</li> <li>• To develop right feature at once with better understanding what they need.</li> <li>• When not confident about the expected implementation and check the expectations before going for actual implementation.</li> <li>• Technical challenges to make sure it can be implemented and identify limitations.</li> <li>• Feedbacks are important with trial ones before actual implementation.</li> <li>• Take major decisions.</li> <li>• Client get to know what is going to be implemented.</li> <li>• Identify must have and nice to have features.</li> <li>• When the customer is not confident, can show we can achieve it.</li> <li>• Resolve customer's doubts what we can understand and what they want.</li> <li>• Sometimes customer learn from us based on our solution.</li> <li>• For larger projects make the life easier.</li> <li>• Proper estimations with reducing variations.</li> <li>• Set responsibility to the customer for the given requirement.</li> <li>• Early identifying impacted areas of the application</li> <li>• Customer engagement through POCs and Prototypes.</li> </ul>
--	-----------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4.2.3. Visualizing

According to Straussian's Grounded theory the above analysis on visualization can be summarized as illustrated in below Figure 4.4. Visualization can be break downed in to three aspects such as; testifying ideas by visualization approaches, Stakeholder perception on visualized idea, and Practice of idea visualizing.

Thinking by doing DT practice and visualizing are going together. By thinking by doing make the idea as tangible, then visualize it to the customer. Based on the study done by qualitative analysis, stakeholder perception on visualize idea helps to implement the right feature. As stated by a tech lead in a mid-sized software development company:

*“Picture is a good tool to show something very clearly and convince the idea”*

This is important because of the customer can see what the development team trying to convince and feedback are very effective to understand the real expectation of the customer.

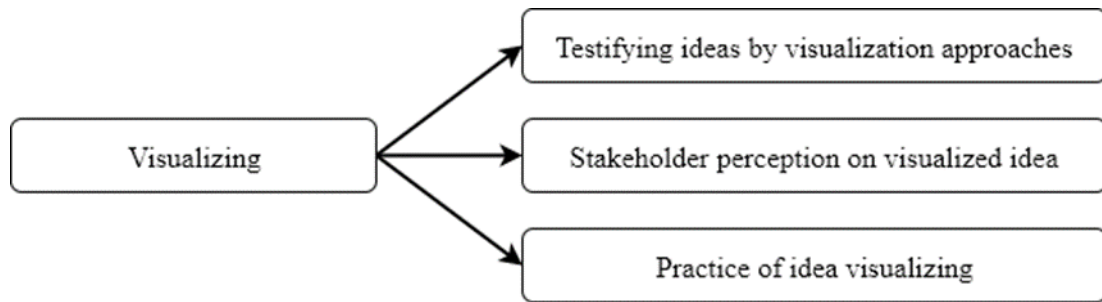


Figure 4.4 Dimensions related to Visualizing

### **Testifying ideas by visualization approaches**

Visualizing can be used to testifying idea such as; functionality, how the system flow, understanding the feature by project team, and when both parties don't have an idea about the way it needs to be implemented. As stated by a senior BA in a mid-sized company:

*“Flowcharts help them to visually show the impact their existing processes. Help to understand how it help us to customer to make their life easy”*

Moreover, a tech lead from a mid-sized company further explains:

*“Mapping with existing system take the base and help to do some visualization and get feedback based on it”*

Prototypes, POC, UX designs, Mockups, Wireframes are known to be widely used techniques for visualization to capture the real feature requirement. Many approaches are used however the teams does not have a clear idea on which is the correct approach for the specific project. Project team use to develop one or more approaches based on the prior experience and their own comfort zone. However, this is not the expectation of DT approach. Identifying the correct approach for visualization still remains a question for many project teams. As stated by a Tech Lead in a mid-sized company:

*“With visualizing existing system, take a base and asking what else you are expecting “*

As stated by a technical project manager in a mid-sized company:

*“Sometimes do POCs or Prototypes only if client is requested to do”*

In the other hand having wireframes is a well know approach to address above issue. Some team members are willing to build wire frames rather than developing Prototypes, POC.

### **Stakeholder Perception**

To implement the right feature, project teams need to understand the right requirement. As per the findings from interviews, stakeholder perception can be identified such; user experience of the given prototype, make sure both customer and the project team are in the same page with mapping project team idea with the customer’s idea, and understanding customer expectation. As stated by a BA in a mid-sized company:

*“Visualizing helps to reduce the cost before go for actual implementation and demo it to the client iteratively with applying changes based on the feedback given”*

### **Practicing of idea visualizing**

As per the analysis, there are several ways to practicing of idea visualizing with agile such; To get end-user feedbacks with visualizing the idea, to get approvals for features, clarification for the customer expectation, situations both customer and the team don't know what to be implemented, and the project team want to show the understanding to the customer. As stated by a technical project manager in a mid-sized company:

*“Without seen that, customer doesn't feel what going to be implemented”*

Moreover, senior BA in a mid-sized company further expressed:

*“Situations both customer and the team don't know what to be implemented for the given idea. If we go with actual implementation with scrum iterations, it will take many iterations to show the implemented solution and apply changes with getting feedbacks”*

Visualization is very useful when it's difficult to convince something to the customer. As stated by a Tech Lead in a mid-sized company:

*“Visualization is used when it’s difficult to convince to the client through verbal communication.”*

According to the study, interviewees perception was to show the customer that we understand as their need and the objective that customer is going to be achieved.

Table 4.5 Open and Axial coding for Visualizing

Selective Coding	Axial Coding	Open Coding
Visualizing	Testifying ideas by visualization approaches	<ul style="list-style-type: none"> <li>• Prototypes, POC, UX designs, Mockups, wireframes.</li> <li>• Understand the functionality to the user.</li> <li>• Clickable HTML pages, clickable wireframes to show the system flow.</li> <li>• Flow charts, Process diagrams to visualize the flow.</li> <li>• Visualize real world existing examples and ask for changes.</li> <li>• Show existing implementation and ask feedbacks.</li> <li>• Based on the feedback do changes and show it to client and ask recommendations.</li> <li>• Activity diagrams and sequence diagrams to visualize the flow to the customer.</li> <li>• Show performance statistics taken through POCs.</li> <li>• Visualize the implementation going to be.</li> </ul>
	Stakeholder perception on visualized idea	<ul style="list-style-type: none"> <li>• Understand the solution going to be implemented.</li> <li>• Get user experience.</li> <li>• Make sure both dev team and client are in the same page.</li> <li>• The idea going to be implemented is clear.</li> <li>• Customers mindset they learn from seen something.</li> <li>• Reduce the cost with showing actual implementation sprint by sprint.</li> <li>• Can clearly mention our idea and map with customer's idea.</li> <li>• Can understand the expectations for what extend.</li> <li>• Easy to communicate the idea to the customer and can get quick feedbacks.</li> <li>• Customer can understand the solution at glance.</li> <li>• Picture is a good tool to show something very clearly.</li> <li>• To remove negative view point.</li> <li>• Reduce the communication gap between customer, end-user and the project team.</li> <li>• Can stop the wrong implementation in early stages.</li> </ul>

	Practice of idea visualizing	<ul style="list-style-type: none"> <li>• To get end-user feedbacks with visualizing the idea.</li> <li>• When having unclear requirements and unclear user expectation.</li> <li>• Take the team onboard with expectations.</li> <li>• Side by side testing.</li> <li>• To get approvals for features.</li> <li>• Customer is not confident about the implementation.</li> <li>• To get an idea what is happening in the system.</li> <li>• Situations both customer and the team don't know what to be implemented.</li> <li>• Reduce the cost by going for actual implementation and getting feedback.</li> <li>• Understand the knowledge gap between project team and the customer.</li> <li>• Think more about the requirement.</li> <li>• When the solution is very complex.</li> <li>• For easy communication and get feedbacks.</li> <li>• To understand most suite for the customer.</li> <li>• Low fidelity to verify the idea communicated correctly.</li> <li>• HI fidelity get the conformation about the UI.</li> <li>• Reduce the communication gap between customer, end-user and the project team.</li> <li>• Through prototypes can understand the mismatching requirements with the customer.</li> <li>• When difficult to explain the understanding verbally and to take feedbacks.</li> </ul>
--	------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 4.2.4. Synthesis of divergent and convergent

According to Strassman's Grounded theory synthesis of divergent and convergent approach is very useful for come up with best possible solution and it can be summarized as illustrated in below figure 4.5. Synthesis of divergent and convergent approach can be break downed in to 3 aspects such as; idea generation, evaluating ides for best approach, and practicing divergent and convergent process.

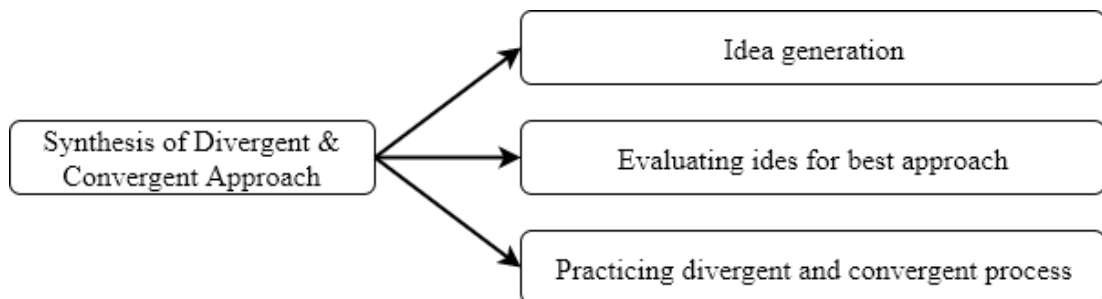


Figure 4.5 Dimensions related to Synthesis of divergent and convergent

## **Idea generation**

Using divergent approach create ideas by exploring many possible solutions. Then using convergent approach combine the best ideas and techniques to provide the best possible solution for the given problem.

As per the findings from interviews, synthesis of divergent and convergent approach is used for idea generation because of such; first idea is not always the best, not confident with having only one idea, to understand type of solution that can be given for a business case. As stated by a senior BA of a mid-sized company:

*“Come up with the business case to the technical team and asking kind of solutions that can be given”*

Normally project team use divergent approach to show different possibilities of the implementation. As stated by a Tech Lead in a mid-sized company:

*” Going for multiple options to explain the different possibilities of the implementation.”*

Some situations, project stakeholders are not going for generating multiple ideas as for solutions. As stated by a project manager in a large-sized company:

*“If the team don't have enough time for verify multiple ideas, do the actual implementation with initial idea.”*

## **Evaluating ideas for best approach**

It shouldn't be a one solution. Several solutions should present to the customer to identify what they really want.

Hereafter, after generating several ideas by using divergent choose the best possible solution with merging several solutions and considering pros and cons. As stated by a BA in a mid-sized company:

*“Different ideas combine into one based on pros and cons”*

Moreover, stated by a tech lead in the same company:

*“Not giving the technically best solution. Give most suitable solution for the business requirement”*

The outcome of the convergent approach is too confident about the solution given to customer to the requirement as he expected.

**Practicing divergent and convergent process**

This study further analyzed how these practices are using by project team. As described by interviewees; evaluate a problem, when there is no idea what is the best solutions, to make innovations, and for large projects to choose the best possible solution. As stated by a tech lead of a mid-sized company:

*“Use divergent and convergent approach when the requirement and the technology are not straightforward”*

Moreover, practicing this divergent and convergent approach when having ideas which are in the same level of intensity. As stated by a Project manager in a mid-sized company:

*“When there are multiple ideas which are similar and hard to choose one with looking at them”*

Table 4.6 Open and Axial coding for Synthesis of Divergent and Convergent

Selective Coding	Axial Coding	Open Coding
Synthesis of divergent and convergent approach	Idea generation	<ul style="list-style-type: none"> <li>• First idea is not always best.</li> <li>• Not confident with the initial idea.</li> <li>• Generate several solutions with different approaches.</li> <li>• Solution with different technologies.</li> <li>• Need several ideas for some situations to provide better solution.</li> <li>• To understand type of solution that can be given for a business case.</li> <li>• To think out of the box.</li> <li>• Go for multiple solution if the team don't have much experience of the area.</li> <li>• To get different solutions from the experts.</li> <li>• Multiple options to explain multiple possibilities.</li> <li>• Sometimes discuss multiple ideas within the team.</li> <li>• Not going forward if we have direct answer.</li> </ul>

	Evaluating ideas for best approach	<ul style="list-style-type: none"> <li>• Evaluate multiple ideas and take a decision.</li> <li>• Evaluate information and capabilities to choose the ideal solution.</li> <li>• Based on the feedbacks and merging some parts from other ideas, choose the best.</li> <li>• Choose best answer based on the brainstorm sessions.</li> <li>• Ignore the past pit-falls based on the experience and choose a solutions.</li> <li>• Combine multiple ideas into one based on pros and cons.</li> <li>• Based on the strengths and weaknesses decide what is the best.</li> <li>• Not giving the technically best solution. Give most suitable solution for the business requirement.</li> <li>• Based on the business need and user experience choose one.</li> <li>• Team members do POCs come up with solutions. Based on that choose the best for the requirement.</li> <li>• Choose the best based on technology, performance, easy to implement etc.</li> <li>• Fish born method with voting choose the best.</li> </ul>
	Practicing divergent and convergent process	<ul style="list-style-type: none"> <li>• Select the technology.</li> <li>• Evaluate a problem.</li> <li>• Go for the best solution.</li> <li>• Suite for service based projects.</li> <li>• Research and development.</li> <li>• Not clear about the best way.</li> <li>• When there is no idea what is the best solutions.</li> <li>• For large projects to choose the best possible solution.</li> <li>• Provide solution for a global regional limitations.</li> <li>• To make innovations.</li> <li>• Customer decide the technology and ask for solutions.</li> <li>• Customer like more options expect to choose best fit to them.</li> <li>• Sometimes first idea go to implementation. While developing facing troubles such situations.</li> </ul>

#### 4.2.5. Collaborative work style

According to Strassman’s Grounded theory, collaborative workstyle is important to work with a group of people get decisions with sharing ideas. It can be summarized as illustrated in below figure 4.6.

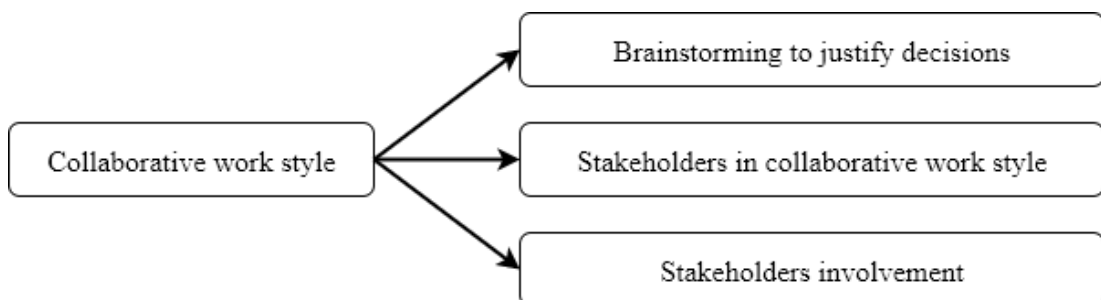


Figure 4.6 Dimensions related to Collaborative work style



## Brainstorming to justify decisions

Based on the study, brainstorming can be used to justify decisions; get ideas, evaluate ideas, pros and cons to make decisions, and clarification with expertize ideas. As stated by a senior PM in a mid-sized company:

*“Brainstorm to get the best idea then go with evaluation process to make decisions”*

Moreover, stated by a project manager:

*“Multiple brains are smarter than a single brain”*

Table 4.7 Open and Axial coding for Collaborative work style

Selective Coding	Axial Coding	Open Coding
Collaborative work style	Brainstorming to justify decisions	<ul style="list-style-type: none"> <li>• Evaluation.</li> <li>• Multiple brains smarter than single brain.</li> <li>• Brainstorm to get expert ideas.</li> <li>• Pros and cons to make decisions.</li> <li>• Brainstorming the solution to be implemented.</li> <li>• Brainstorming with the team for clarification.</li> <li>• Developers think only technical perspective, BA focusing only business aspect therefore, only developers solution might be too complicated to customer.</li> <li>• Collaborate to maintain harmony with tech teams.</li> <li>• When tight deadline project rather than take own decision take collective knowledge.</li> <li>• Spread the risk with the team without taking own risk.</li> <li>• Estimate the accurate work effort.</li> <li>• Groom the requirement.</li> <li>• Task breakdown.</li> <li>• Resolve code level issues and requirement related issues.</li> <li>• To get to know the unknown areas.</li> <li>• Get another perspective about the requirement.</li> </ul>
	Stakeholders in collaborative work style	<ul style="list-style-type: none"> <li>• Stakeholders of the project such as Product Owner, Scrum Master, Business Analysis, Project Team, Tech experts and Customer.</li> <li>• Technical specialize people to get consultancy.</li> <li>• Developers tell alternative approaches and feasibility of implementation.</li> <li>• QA share the impacted areas.</li> <li>• PO share the business needs.</li> <li>• Customer's technical team.</li> <li>• Onsite team.</li> </ul>

	Stakeholders involvement	<ul style="list-style-type: none"> <li>• Clarify unclear requirements.</li> <li>• Understand current process.</li> <li>• Difficult to understand what BA provided.</li> <li>• Make development fast.</li> <li>• Resolve practical issues while development.</li> <li>• Discuss team understanding and to get clarified.</li> <li>• Conflict and confuse situations.</li> <li>• Feedback for the presentations.</li> <li>• Conversations for finalize the requirement.</li> <li>• No need to blame individuals about wrong understanding of requirements.</li> <li>• Avoid missing requirement to make unhappy customer.</li> <li>• Domain expert consultants.</li> <li>• Technical people to get insights from the customer.</li> <li>• When needs clarifications and to get confirmation.</li> <li>• Discuss the requirements that clarifications are needed.</li> <li>• To take technical decisions.</li> <li>• Customer also have some ideas which are useful to find a solution.</li> </ul>
--	--------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Further mentioned by the interviewees, it helps to avoid conflicts with different expertise employees such; developers think only technical perspective, and BA focusing only business aspect. Furthermore, stated by a senior BA in a mid-sized company:

*“Solution provided by the dev team only, too complicated to the customer”*

Brainstorming is not work together. It needs to be used to justify a decision with evaluating with other’s recommendations. As stated by a tech lead of a mid-sized company:

*“When there is very tight deadline project rather than take own decision take collective knowledge”*

They are applying brainstorming only for tight deadlines. But as a practice, this needs to be used whenever an employee or team needs to evaluate a decision and before it executes.

### **Stakeholders in collaborative work style**

According to the study, stakeholders are important for collaborative work. All the stakeholders that required; senior members of the team, specialize people of the area, and owner to make a decision at the end. Otherwise the brainstorming is not useful. As stated by a technical specialist in a mid-sized company:

*“Based on the scenario the relevant people should be involved”*

No need every one for each and every time. That’s needs to be decide based on the scenario to be addressed.

Traditionally blaming to individual if something goes wrong. With brainstorming, everyone should take the responsibility about the final decision. Because, everyone involved to get the decision. As stated by a senior BA in a mid-sized company this is important to meet the customer needs:

*“Customer is not happy the requirement is not able to 100% meet their needs. Very concern about very little details also”*

### **Stakeholder involvement**

As per the analysis of the interview data based on the Straussian’s Grounded theory, stakeholder involvement is very important to understand the right requirement and to provide the right solution such; clarify unclear requirements, understand current process, and resolve practical issues while development. As stated by a technical specialist in a mid-sized company:

*“Sometimes it’s difficult to understand the requirement that BA provided”*

As mention by interviewees, they are involving domain experts and technical experts for brainstorming to clarify gray areas.

### **4.3. Summary**

The main focus of this research to identify the usage of DT practices with the Agile project management and identify effective use of DT practices in the software development industry. Data collected through the semi-structured interviews and analyzed by using Straussian’s Grounded theory approach. Sample data analysis and presentation were discussed in this chapter. Research result was written based on the data analysis. In addition to that identified the most popular Agile methodology in the Sri Lankan software development industry and identified the customer’s pain-points from the project stakeholders point of view to be addressed to satisfy the customers.

## **5. RECOMMENDATIONS AND CONCLUSION**

This chapter summarize the work done in above chapters and provides the conclusions and recommendations of the research work. Section 5.1 provides the conclusion of the research work, Section 5.2 list the Observations made during the research work, finding from the research data analysis, and the recommendations for management in the software development companies. Section 5.3 list down the research limitations and Section 5.4 describes the future works.

### **5.1 Conclusion**

The purpose of the current study was to identify the usage of design thinking practices with Agile project management and identify how effectively user DT practices with Agile. DT elements; Human-centered approach, Thinking by doing, Visualizing, Synthesis of Divergent and Convergent approach, and Collaborative work style stated in the literature review by Hassi & Laakso (2011) were identified as DT practices that can apply with software development projects. With conducting the interviews, identified the purpose of using DT practices in Agile project management. Then the recommendation focus on how effectively apply those practices to improve customer expectations on Agile project management.

Since this is new area to be studied, this study focus on mid-sized and large-sized service based software development companies in Sri Lanka. Because small-sized or startup companies are not following this kind of processes and practices at the initial stage. Product based software development companies also not considered because there are some difficulties to customize products for specific customer needs. Interviewees are senior level people of those companies; Business Analysis, Project Managers, Architects, and Tech Lead who are adding major contribution to the decision making process. When gathering the data focused on awareness of techniques of each and every DT practice, participants to practice these techniques, usage of techniques, and advantages and benefits that they have gained with applying with agile methodologies to provide customer expected solution. This was the most challenging part of this study. Project stakeholders use these techniques which

related to DT practices but they don't know theoretically that they are using DT practices.

This study first identified the agile methodology that the interviewees use for software development projects. Scrum is the most used (71%) agile practice. Kanban, XP, and Lean are the other agile practices. This study also identified the customer-pain points from the interviewee's point of view which required to address to make the customer satisfaction. According to the data, identified the situations that customer is worrying mostly while developing a product.

The first element is the "human-centered approach". During the study, most of the project stakeholders use "human-centered approach" in the requirement gathering to understand the real problem of the customer. Many situations, the customer has the idea only. Don't know how it needs to be implemented. With using this approach, project stakeholders could understand the real need of the customer. Therefore, "human-centered approach" positively effecting to the satisfy customer expectations and the project deliverables.

Second element is the "thinking by doing approach". According to study this approach is used to verify the feasibility of the implementation going to be implemented. With applying thinking by doing practices, project team is more confident with solution that expecting to deliver to the customer. Customer also confident about the project team that they can deliver the expected solution. Both customer and the project team can identify the limitations of the implementation going to be and if there any changes needs to be applied or go for an alternative solution, can be decide at early stage of development. Therefore, "thinking by doing approach" is positively effect to the customer expectation.

Third element is the "visualizing". Based on the study this approach is used after "thinking by doing approach". This is for testifying the pilot solutions with the customer and end-users that team build based on the "thinking by doing approach". Then customer can see the idea as tangible and he can better understand about the solution is going to be implemented, he intends to think about it from different aspects, and can give effective feedbacks for changes before start the actual work.

Forth element is the “Synthesis of divergent and convergent approach”. This approach for idea generations, evaluate the ideas and choose the most suitable solution for the problem. This practice involving to choose the most suitable solution from many aspects; business problems, technical problems, identify limitations, innovation with thinking out of the box, and more options to choose one for the customer from several solutions. Therefore, “synthesis of divergent and convergent approach” is positively effecting to satisfy customer expectations of the project.

Fifth and last element is the “collaboration work style”. This is used for brainstorming the work with the team by involving internal and external stakeholders. This approach is coming under Agile practices as well. Project team involving to discuss internal problems; resolve practical issues with development, make development fast, resolve conflict and confuse situations, get opinion from technical experts, and to take technical decisions. Customer involving for situations; clarify unclear requirements, when difficult to understand what BA is provided, verify the team understanding with the customer, and avoid missing requirements that make customer unhappy. At the end, project stakeholders trying to make the customer happy. Therefore, “collaboration work style” positively effecting to satisfy customer expectations of the project.

Based on this study it can be concluded that DT practices such as human centered approach/empathy, thinking by doing or action based, visualizing, synthesis of diverging and converging, and collaborative work style are helpful in enhancing the customer expectation. At the same time, we identified that organizations use DT techniques without having a deeper understanding.

## **5.2 Observations, findings and recommendations**

During the data collection and data analysis series of observations and findings are being discovered and recommendations are derived based on the observations and findings.

## Observations

While conducting the interviews for the specific questions, following observations was identified as current feeling of Agile, DT practices, and customer expectations as mentioned in Table 5.1.

Table 5.1 Observations identified when conducting interviews

#	Description
1	Organizations use techniques in DT practices without having theoretical knowledge on DT.
2	Many project teams are unaware that they are using techniques in DT practices on a daily basis.
3	Lack of understanding of agile principles and DT principles.
4	Techniques used to capture the requirements are customer journey mapping, prototype, proof of concept, wireframes, user experience (UX) design, story-mapping, use-cases, scenario identification, workflows, customer profiling, heat maps, A/B testing, and persona techniques.
5	Common practices such as collaboration are available for both agile and DT practice.
6	Most of the time vendor not engaged with the end user and communicated only through product owner, which is not acceptable.

## Findings

I categorized the best practices to satisfy customer expectations in agile projects into five groups as customer's real need identification, transforming customer's real needs into pilot solutions, visualizing the pilot solution for customer feedback, idea generation for the pilot solution, and brainstorming. Each of these findings are described in Table 5.2.

Table 5.2 Best practices to satisfy customer expectations

#	Description
1	<p><b>Customer’s real need identification</b></p> <ul style="list-style-type: none"> <li>• Accepting requirement only from the business analyst is not recommended.</li> <li>• Before starting actual implementation, it is recommended to spend a good amount of time (e.g., 6 weeks) to capture customer need.</li> <li>• Adhering to DT techniques such as customer journey mapping, user story mapping, user experience design, scenario identification, heat maps, and customer profiling.</li> <li>• Practice HCA with workshops and discussions.</li> </ul>
2	<p><b>Transforming customer need into pilot solutions</b></p> <ul style="list-style-type: none"> <li>• Proof of concept (POC), prototype, persona identification and wireframes are used determine the idea.</li> <li>• Generate questions to the customer and get feedback during sprint works, to drive the project into right direction.</li> <li>• Pilot solutions help to indicate failures early rather than leading to an unexpected failure.</li> <li>• Ensure project team is solving the right solution and validate before expensive development or design occurs.</li> </ul>
3	<p><b>Visualizing the pilot solution for customer feedback</b></p> <ul style="list-style-type: none"> <li>• Understand stakeholder perception on idea visualization helps to implement the right feature.</li> <li>• Get feedback by asking questions on the outcome of the planned solution, based on pilot solutions such as proof of concepts, prototypes, and mockup.</li> <li>• Testifying the idea with the end-user.</li> <li>• Adjust customer mindset to think more about their own requirements and planned solution.</li> </ul>
4	<p><b>Idea generation for the pilot solution</b></p> <ul style="list-style-type: none"> <li>• Divergent approach is used for idea generation, as the first idea is not always the best idea.</li> <li>• Identify the most suitable solution for customer given business context.</li> <li>• Have confidence in the development team.</li> </ul>
5	<p><b>Brainstorming</b></p> <ul style="list-style-type: none"> <li>• Brainstorming can be used to justify decisions, generate ideas, evaluate ideas, pros and cons to make decisions, and clarification of expertise ideas.</li> <li>• Spread responsibility with everyone to about the final decision.</li> <li>• Stakeholder involvement is essential to understand the right requirement.</li> <li>• Involve domain and technical experts to clarify gray areas.</li> <li>• Avoid conflicts among project stakeholders due to different understandings.</li> </ul>



## Recommendations

Based on the observations and findings, I propose a framework in Figure 5.1, which could be used to improve the customer satisfaction. This framework describes how to use five DT practices to improve the customer satisfaction. There are four techniques in HCA, thinking by doing category and three factors in visualizing, synthesis of the diverging and converging, and collaborative work style category. All the factors are briefly described in Table 5.3. It explains how you can use the respective factor to improve customer satisfaction.

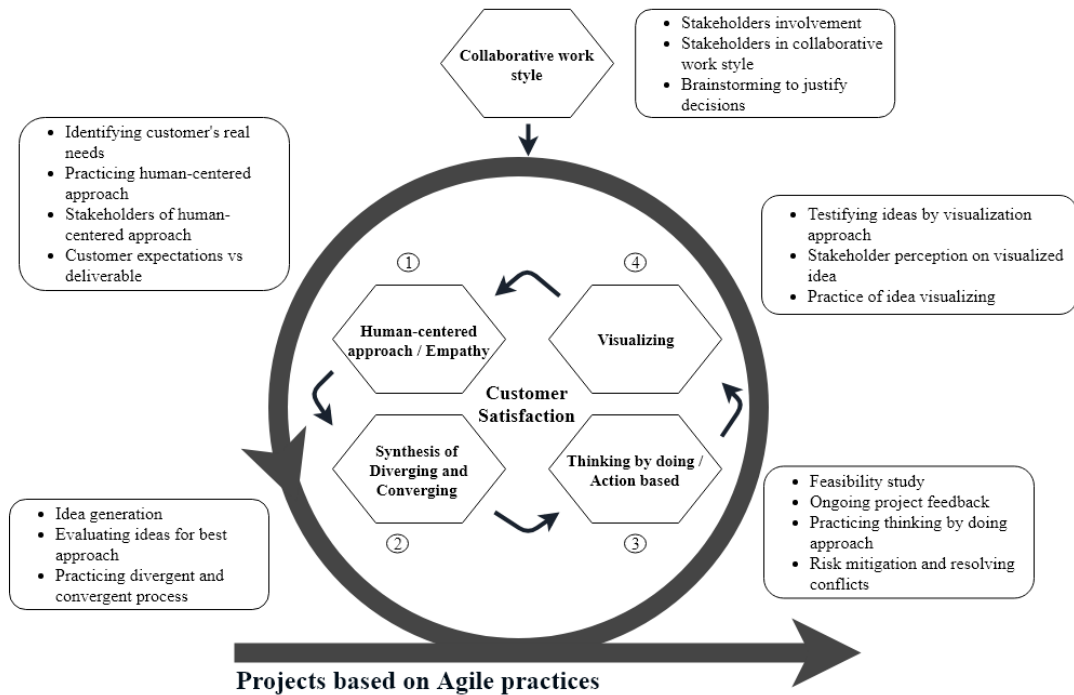


Figure 5.1 Framework to enhance customer satisfaction using DT practices in agile practices

Initial four recommendations are based on the proposed framework and others are based on the observations and findings.

- *As the first step, apply HCA in the requirement gathering phase to identify customer real needs.*

Understanding correct requirement and what the customer expecting to see in his product are the most critical part of the project. Therefore, it is important to allocate considerable time to understand correct requirement with applying HCA techniques.

- *As the second step, generating ideas by using Divergent and Convergent approach.*

After identifying the real need of the customer, it is important to identify the best possible solution that is most suite for the business needs. Therefore, project team needs to be generated multiple ideas come up with several best solutions based on pros and cons.

- *As the third step, convert complex problems into pilot solutions based on generated ideas.*

In this step needs to be verified the feasibility of selected solutions? With this, it will increase the confident level of both customer and the project team. Can be identified failure very early rather unexpected failures.

- *As the forth step, visualize the pilot solutions for end-user testing and for their feedback.*

In this step, get the customer and end-user feedbacks for the pilot solution. With this customer can imagine what will be the end result of the product and he will intend to think more to verify his/her requirement is correct.

- *Conduct DT workshops and establish DT coaches.*

According to the observations while conducting the interviews, awareness about the DT practices are minimum within the project stakeholders. Therefore, first need to make the awareness about the DT and give practical trainings to the stakeholders of the project to encourage to use it.

- *Combine Agile with DT with adding DT tasks to backlog and grooming them.*  
Based on the findings, DT items also needs to be added to the Agile sprints. Those should be considered as tasks in the project.
- *Make a DT culture in the organization for more customer centric and collaboration.*  
As make the Agile culture, organizations need to promote DT culture as well. Then, project stakeholders will be adopted to use DT practices by default.
- *Use DT as a tool for better decision-making around the development.*  
Encourage project stakeholders to use DT practices for better decision making while developing a software development project.

Table 5.3 DT practices and factors

<b>DT Practices</b>	<b>Factors</b>
Human-centered approach (HCA)	Identifying customer's real needs
	Practicing human-centered approach
	Stakeholders of human-centered approach
	Customer expectations vs deliverable
Thinking by doing / Action based thinking	Feasibility study
	Ongoing project feedback
	Practicing thinking by doing approach
	Risk mitigation and resolving conflicts
Visualizing	Testifying ideas by visualization approaches
	Stakeholder perception on visualized idea
	Practice of idea visualizing
Synthesis of Divergent and Convergent approach	Idea generation
	Evaluating ides for best approach
	Practicing divergent and convergent process
Collaborative work style	Brainstorming to justify decisions
	Stakeholders in collaborative work style
	Stakeholders involvement

### **5.3 Research limitations**

Following research limitations can be identified during this study, this research focus only on the service based software development companies where the number of employees are above 50, and also only the companies providing IT services which is accountable to around 26 companies. Hence many of the companies for the selected 26 are new to the DT there is a limitation of selecting the companies who practice DT.

Whereas researcher did a pilot study to identify the companies know practice DT and identified and interviewed 10 companies. Whereas it is not completely covering the entire population which might have different views on DT. There might be more companies who practice DT form the remaining however due to difficulties in reaching those researcher has conducted the study with only 10 companies.

Moreover, DT is not well established among Sri Lankan software development companies. Many companies do not have the desired knowledge of DT even though they practice DT. Therefore, there might have gray areas which couldn't understand how to practice DT properly.

The study mainly focusses on the top level of the company such as; project managers, business analyst, tech leads, and architect who involving to requirement gathering, decision making, design, development, and testing.

When considering the offshore factor of software development, customer interaction is high even the vendor in another country. Most of the times BA is from the customer's side or BA or technical person will be placed in customer side to reduce the communication gaps.

#### **5.4 Future work**

The proposed framework can be improved further by applying it on real world scenario with adding DT tasks to the agile backlog and grooming them. As an example can be measure the performance of this framework with adding tasks that end-user interaction is needed into earlier sprints before starting the actual implementation. If we add those tasks in the same sprint, will have to hold development till end-user feedback receive.

As a second step, proposed framework can be improved with other two categories, cognitive approach and the Mindset identified by the (Hassi & Laakso, 2011).

Furthermore, there researcher was unable to identify the opinion on QA professionals since testing also considering in DT process.

## REFERENCES

- Agarwal, N., & Rathod, U. (2006). Defining “success” for software projects: An exploratory revelation. *International Journal of Project Management*, 24(4), 358–370. <https://doi.org/10.1016/j.ijproman.2005.11.009>
- Al-ahmad, W., Al-fagih, K., & Khanfar, K. (2009). A Taxonomy of an IT Project Failure : Root Causes. *International Management Review*, 5(1), 93–104. [https://doi.org/10.1016/S0263-7863\(98\)00047-7](https://doi.org/10.1016/S0263-7863(98)00047-7)
- Begel, A., & Nagappan, N. (2007). Usage and Perceptions of Agile Software Development in an.
- Brooks, F. P. (2010). *The Design of Design*.
- Clark, K., & Smith, R. (2008). Unleashing the Power of Design Thinking Unleashing the Power of Design Thinking. *Leadership*, (617).
- Coram, M., & Bohner, S. (2005). The Impact of Agile Methods on Software Project Management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, 363–370. <https://doi.org/10.1109/ECBS.2005.68>
- Crawford, B., De La Barra, C. L., & Letelier, P. (2008). Communication and creative thinking in agile software development. *IFIP International Federation for Information Processing*, 277, 205–216. [https://doi.org/10.1007/978-0-387-09697-1\\_17](https://doi.org/10.1007/978-0-387-09697-1_17)
- Dorst, K. (2011). The core of “design thinking” and its application. *Design Studies*, 32(6), 521–532. <https://doi.org/10.1016/j.destud.2011.07.006>
- Dunne, D., & Martin, R. (2006). Design thinking and how it will change management education: An interview and discussion. *Academy of Management Learning and Education*, 5(4), 512–523. <https://doi.org/10.5465/AMLE.2006.23473212>
- Faily, S., & Flechais, I. (2011). Persona Cases: A Technique for Grounding Personas. *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages 2267-2270*, 2267–2270. <https://doi.org/10.1145/1978942.1979274>
- Feeler, W. (2012). Being there: A grounded-theory study of student perceptions of instructor presence in online classes. *ProQuest Dissertations and Theses*, 225. Retrieved from [http://search.proquest.com/docview/1266830430?accountid=35812%5Cnhttp://av4kc7fg4g.search.serialssolutions.com/?ctx\\_ver=Z39.88-2004&ctx\\_enc=info:ofi/enc:UTF-8&rfr\\_id=info:sid/ProQuest+Dissertations+&+Theses+Full+Text&rft\\_val\\_fmt=info:ofi/fmt:kev:mtx:disse](http://search.proquest.com/docview/1266830430?accountid=35812%5Cnhttp://av4kc7fg4g.search.serialssolutions.com/?ctx_ver=Z39.88-2004&ctx_enc=info:ofi/enc:UTF-8&rfr_id=info:sid/ProQuest+Dissertations+&+Theses+Full+Text&rft_val_fmt=info:ofi/fmt:kev:mtx:disse)
- Frye, U., & Inge, T. (2013). The Integration of Design Thinking and Lean Software

Development from the Perspective of Product Owners and Scrum Masters, 64.

- Gurusamy, K., Srinivasaraghavan, N., & Adikari, S. (2016). An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation. [https://doi.org/10.1007/978-3-319-40409-7\\_4](https://doi.org/10.1007/978-3-319-40409-7_4)
- Hassi, L., & Laakso, M. (2011). Conceptions of Design Thinking in the Design and Management Discourse. *Proceedings of IASDR2011*, 1–10.
- Holloway, M. (2009). How tangible is your strategy? How design thinking can turn your strategy into reality. *Journal of Business Strategy*, 30(2/3), 50–56. <https://doi.org/10.1108/02756660910942463>
- Hussain, A., Mkpojiogu, E. O. C., & Kamal, F. M. (2016). International Review of Management and Marketing The Role of Requirements in the Success or Failure of Software Projects. *International Review of Management and Marketing International Soft Science Conference*, 6(7), 306–311. Retrieved from [www.econjournals.com](http://www.econjournals.com)
- Jeeva Padmini, K. V., Perera, I., & Dilum Bandara, H. M. N. (2016). Applying agile practices to avoid chaos in User Acceptance Testing: A case study. *2016 Moratuwa Engineering Research Conference (MERCCon)*, 96–101. <https://doi.org/10.1109/MERCCon.2016.7480122>
- Kaur, R., & Sengupta, J. (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, 2(2), 1–4. Retrieved from <http://arxiv.org/abs/1306.1068>
- Lucena, P., Braz, A., & Tizzei, L. (2016). IBM Design Thinking Software Development Framework. *Conference: 7th Brazilian Workshop on Agile Methods (WBMA '2016)*, (November).
- Memmel, T., Gundelsweiler, F., & Reiterer, H. (2007). Agile Human-Centered Software Engineering. *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but Not as We Know It - Volume 1*, 167–175. <https://doi.org/10.1177/004057368303900411>
- Mickahail, B. (2015). Corporate Implementation of Design Thinking for Innovation and Economic Growth. *Journal of Strategic Innovation and Sustainability*, 10(2), 67.
- Mintz, S. (2017). Design Thinking. Retrieved from <https://www.insidehighered.com/blogs/higher-ed-gamma/design-thinking>
- Mundra, A., Misra, S., & Dhawale, C. A. (2013). Practical scrum-scrum team: Way to produce successful and quality software. *Proceedings of the 2013 13th International Conference on Computational Science and Its Applications, ICCSA 2013*, 119–123. <https://doi.org/10.1109/ICCSA.2013.25>
- Pourdehnad, J., Wexler, E. R., & Wilson, D. V. (2011). Systems & Design Thinking: A Conceptual Framework for Their Intergration, 1–16. Retrieved from [http://repository.upenn.edu/od\\_working\\_papers%0Ahttp://repository.upenn.edu/](http://repository.upenn.edu/od_working_papers%0Ahttp://repository.upenn.edu/)

- Razzouk, R., & Shute, V. (2012). What Is Design Thinking and Why Is It Important? *Review of Educational Research*, 82(3), 330–348. <https://doi.org/10.3102/0034654312457429>
- Rosson, M. B., & Carroll, J. M. (2002). Scenario-Based Design. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 1032–1050. <https://doi.org/10.1016/j.jbi.2011.07.004>
- Saiedian, H., & Dale, R. (2000). Requirements engineering: Making the connection between the software developer and customer. *Information and Software Technology*, 42(6), 419–428. [https://doi.org/10.1016/S0950-5849\(99\)00101-9](https://doi.org/10.1016/S0950-5849(99)00101-9)
- Strauss, A., & Corbin, J. (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Basics of Qualitative Research Grounded Theory Procedures and Techniques* (Vol. 3). <https://doi.org/10.4135/9781452230153>
- Thienen, J. von, Noweski, C., Meinel, C., & Rauth, I. (2011). Design Thinking. *American Journal*, 81–99. <https://doi.org/10.1145/2535915>
- Vetterli, C., Brenner, W., Uebernickel, F., & Petrie, C. (2013). From palaces to yurts: Why requirements engineering needs design thinking. *IEEE Internet Computing*, 17(2), 91–94. <https://doi.org/10.1109/MIC.2013.32>
- Vetterli, C., Uebernickel, F., Brenner, W., Häger, F., Kowark, T., Krüger, J., ... Sikkha, V. (2013). Jumpstarting Scrum with Design Thinking. *Design Thinking Research: Building Innovators*, 2(2), 1–60. <https://doi.org/10.1007/978-3-319-06823-7>
- Waks, S., Trotskovsky, E., Sabag, N., & Hazzan, O. (2011). Engineering thinking: The experts' perspective. *International Journal of Engineering Education*, 27(4), 838–851.



## **APPENDIX A: INTEVIEW QUESTIONS**

### **Geographical Information:**

1. Name
2. Designation
3. Gender
4. Experience in Agile projects
5. No of Agile projects engaged with:

### **Common interview questions**

1. Could you explain your role? Focus on day-today activities and responsibilities of the project. Focus to identify what he is doing in the project.
2. What type of agile methodology you have experienced with (Scrum, XP, TDD, and Hybrid)?

### **Human-centered approach / Empathy**

1. What are the type of techniques you used to get to know about your customer's real needs (expectations) when you are gathering requirements? For how long? If not what are the techniques, you are using?
2. How are you practicing the above mention technique? Who are the participants?  
  
<Ex: Product owner, Dev team, Scrum master, BA >
3. Did you achieve any advantages using above mention techniques to understand the product value and customer expectations?
4. What is your recommendation for others to use those techniques in requirement gathering? If so, why?

### **Thinking by doing / Action based**

1. Are your team intending to create prototypes for the requirements and for the initial idea and getting feedback before going for actual implementation?
2. Is it good approach to implement what you are thinking instead of just thinking?
3. Could you get cost benefit with using prototypes instead of going for actual implementation and getting feedback?
4. How long and what kind of situation you are using this technique (Ex: Both product owner or dev team don't know what to implement)

### **Visualizing**

1. Are you using prototypes or graphical format (UX Design) to visualize your idea as tangible?
2. Is it good approach to show what will be the end result by using prototype or UX design or POC?
3. Is your customer testing the prototype with end-user to get end-user experience of the product?
4. How long and what kind of situations that you have been using these techniques in your project?

### **Synthesis of Divergent and Convergent approach**

1. Are you using divergent approach to come up with multiple alternative ideas instead of taking initial idea as the best?
2. How about using the convergent approach used to identify patterns and relationships for moving towards with the best solution?

3. How long and what kind of situation you have been applied these techniques and could you deliver best solution to the customer with making him happy?  
How?

### **Collaborative work style**

1. Are you brainstorming the work with your team?  
Ex: KJ method (group consensus technique)
2. Are you involving different stakeholders such as development team, marketing, customer and public to get ideas and decisions?
3. Have you involved your customer to your work to help you understand the problem?
4. How long and what kind of situations you have been using this above approaches in your project and have provided customer expected solution?

### **Satisfy customer expectation projects based on agile**

1. Have you considered the pain points (understand real problem, maximize the value of the product, competitive advantage or user experience etc.) of the customer journey in your project?
2. Have you considered that both internal (time, cost and scope) and external (customer and the end users expectations and social) factors should be focus on to satisfy customer expectations in your project?
3. Are you making effort to get to know the client and their needs? If so, how?
4. Are you understanding ways and means to improve the customer satisfaction?  
What are they?

## **APPENDIX B: REVISED INTERVIEW QUESTIONS**

### **Geographical Information:**

1. Gender
2. Experience in Agile projects
3. No of Agile projects engaged with

### **Common interview questions:**

1. Could you explain your role? Focus on day-today activities and responsibilities of the project. Focus to identify what he is doing in the project.
2. What type of agile methodology you have experienced with (Scrum, XP, TDD, and Hybrid)?

### **Human-centered approach / Empathy**

1. What are the type of human centered approach techniques used to get to know about the customer's real needs (expectations) when you are gathering requirements?

Examples:

- Persona techniques to identify the characteristics of the end-user (age, location, education etc.) look into who using this product?
  - Scenario based approach is product related scenarios, how they use the product.
  - Use cases to understand how the end-user interact with the real world
  - Consumer journey mapping to understand the customer experience when using the product
2. How are you practicing the above mention technique? Who are the participants?

Ex: Product owner, Dev team, Scrum master, BA

3. Did you achieve any advantages using above mentioned techniques and what is your recommendations to use this techniques?

### **Thinking by doing / Action based**

1. What are the type of "thinking by doing practices" for getting feedback before going for actual implementation?

Ex: Create prototypes, POCs for the requirements and for the initial idea and getting feedback before going for actual implementation?

2. Advantage and benefits of following "thinking by doing approach" instead of directly going for actual implementation?
3. What kind of situations that can have used "thinking by doing approach"?

### **Visualizing**

1. What are the techniques that you are using to visualize the idea as tangible in your project? How they are using?

Ex: Prototypes, POC or graphical format (UX Design)

2. What are the advantages and benefits of using visualizing your idea as tangible?
3. What kind of situations that you have applied this visualizing DT practice in your project?

### **Synthesis of divergent and convergent approach**

1. Are you using divergent approach to come up with multiple alternative ideas instead of taking initial idea as the best?
2. How about using the convergent approach used to identify the best solution out of several solutions?
3. What kind of situations have you applied this divergent and convergent approaches in your project?

### **Collaborative work style**

1. Are you brainstorming the work with your team? The situations that brainstorming is necessary?
2. Who are the participants for practicing collaborative work style? How they are involving?
3. Have you involved your customer to your work to help you understand the problem and what kind of situations?

### **Satisfy customer expectation projects based on agile**

1. According to your understanding, what are the pain points of the customer journey

Ex: understand real problem, maximize the value of the product, competitive advantage or user experience etc.

## **APPENDIX C: MEMO WRITING OF GROUNDED THEORY**

### **Human-centered approach**

HCA is using to identify customer's real need. Many situations requirements given by BA is not enough for identifying customer's real needs. There are several techniques that project stakeholders are using to identify the customer's real needs; customer journey mapping, prototypes, POCs, wireframes, use-cases, scenarios, customer profiling, heat maps, AV testing, and persona technique. Not only the BA or PO, almost every stakeholders of the project needs to be participated to success this practice. The main objective of practicing HCA is; fine-tune the requirements, understand user experience, identify appropriate technologies, understand customer's business objective, reduce rework, and sign off the features.

### **Thinking by doing approach**

Thinking by doing is a feasibility study about the customer requirement can be implemented or not, look at the problem from different aspects, and to identify limitations. There are several techniques used by project stakeholders to verify that requirement can be implemented; POC, Prototypes, Wireframes, and Mockups. This approach is very important to get ongoing project feedbacks, identify failures early, and for accurate estimations. When practicing this approach, it will increase the confidence level of the project stakeholders, risk can be mitigated, early failures, more paths to identify different options, and make effective communication with the client.

### **Visualizing**

Visualizing approach is followed by the thinking by doing approach. This is used to testifying the idea with the customer and get feedback for identifying the problem correctly. Stakeholder perception is very important with the visualization. Customer can understand the solution going to be implemented after seen POC, Prototype or

clickable wireframe. Customer starts to think deeply about the idea and will give effective feedbacks for understanding the solution at glance. This is very useful to understand the expectations and limitations also.

### **Synthesis of divergent and convergent approach**

Divergent approach is used for idea generation. Because, most of the situations the first idea is not always best and confidence level also low. It is important to generate several ideas to come up with the best possible solution. Then, convergent approach is used to evaluate multiple ideas and choose the best possible idea. It will consider several factors; based on customer feedback, pros and cons, ignore pit-falls based on the previous experience, identify the strengths and weaknesses, and most suitable solution for the business requirement.

### **Collaborative work style**

Collaborative work style is brainstorming to justify decisions. Collaborative working culture helps to get expert ideas, clarifications, get ideas from different aspects, and spread the risk among the team rather than taking risk by one person. Generally, senior project stakeholders are involving for brainstorming sessions. It is important to engage different stakeholders; DEV, QA, BA, and PO to clarify unclear requirement from different angles, resolve practical issues, take feedbacks, avoid missing requirements, and to take expert consultations. This is more productive than individual ideas and customer's ideas also helps to find a solution.



## APPENDIX D: CODING BASED ON GROUNDED THEORY

Selective Coding	Axial Coding	Open Coding
Human-centered approach	Identifying customer's real needs	<ul style="list-style-type: none"> <li>• Most of the time BA gives requirement and he clarified with the customer and the end-user</li> <li>• PO give the requirement and he clarify those with end-users.</li> <li>• Customer journey to understand how user going use the system, experience and for the clarifications.</li> <li>• Prototype, POC, Wireframes, and UX to fine tune the requirement.</li> <li>• Story mapping for requirement defined in the backlog.</li> <li>• Use-cases for understand the system and what is it about and how user interact with the system.</li> <li>• Scenario based for understanding requirement based on different scenarios.</li> <li>• Build up minimal viable product.</li> <li>• Presentation to get feedbacks.</li> <li>• Customer profiling based on the background check. It is easy to understand what suite to the customer.</li> <li>• Workflows to identify how the system flow is going.</li> <li>• Heat map is something like mouse click points which is most impotent to increase customer experience.</li> <li>• A/B testing put two interfaces to customer testing to choose a one interface.</li> <li>• Persona techniques to identify user characteristics.</li> <li>• Prefer to use the customer journey mapping more than the scenario based approach.</li> </ul>
	Practicing human-centered approach	<ul style="list-style-type: none"> <li>• Workshops with discussions.</li> <li>• Product Owner communication with end users to get their feedbacks.</li> <li>• BA explains current understanding.</li> <li>• PO explain the requirement that he expected.</li> <li>• Convert high level vision into operations.</li> <li>• End users integration through UAT.</li> </ul>

	Stakeholders of human-centered approach	<ul style="list-style-type: none"> <li>• Stakeholders of the project such as Business Owner, Product Owner, Scrum Master, Business Analysis, Project Team, End-user representatives from the customer's side.</li> <li>• Customer's technical people to communicate with the development team.</li> </ul>
	Customer expectations vs deliverable	<ul style="list-style-type: none"> <li>• Fine-tune, Clarify, and finalize the collected requirements and wider aspects of the requirements.</li> <li>• Understand user experience of the domain and end-user background and what he really need.</li> <li>• Identify the appropriate technologies for the requirement.</li> <li>• Task prioritization must have and nice to have features.</li> <li>• Understand the customer's business objective and what customer is going to be achieved.</li> <li>• Sign off features with the client.</li> <li>• Can understand customer expectations and can ask right question from right people.</li> <li>• Help to understand all the impacted area of the application.</li> <li>• Reduce rework because of different understanding about the requirement.</li> <li>• Avoid unrealistic estimations.</li> <li>• Understand how user interaction with the system.</li> <li>• Provide very little things as well to satisfy customer.</li> <li>• Reduce the communication gap between intermediaries (PO, BA, and End-user).</li> </ul>
Thinking by doing / Action based	Feasibility study	<ul style="list-style-type: none"> <li>• POC to understand the feasibility, limitations, effort, performance, technology and make sure that can be implemented.</li> <li>• Prototype to show what we understood and what is going to be implemented.</li> <li>• Wireframe and process diagrams give better outcome.</li> <li>• Mockups for understanding screen functionality.</li> <li>• Taking feedbacks based on the POC, Prototype and</li> </ul>

		<p>mockups to finalize the requirement.</p> <ul style="list-style-type: none"> <li>• Some cases without going for POC or Prototype can give a best solution.</li> </ul>
	Ongoing project feedback	<ul style="list-style-type: none"> <li>• Wireframes to show the user interface and viewpoints to take customer's feedback.</li> <li>• PSD to show how the UI colors etc.</li> <li>• Mockups to describe the user interface very detail.</li> <li>• Prototypes make adjustments and direct to the right direction and understand the functionality of unknown areas.</li> <li>• Prototypes to understand how to do the implementation correctly on unknown areas.</li> <li>• Accurate estimations.</li> <li>• To test different approaches.</li> <li>• Customer engagement through Prototypes and POCs.</li> </ul>
	Practicing thinking by doing approach	<ul style="list-style-type: none"> <li>• Show outcome and clean path about what needs to be done.</li> <li>• Risk mitigation with avoid misleading requirements to project go wrong.</li> <li>• Easy to make decisions.</li> <li>• Get client's sign off.</li> <li>• Understand real need at the early stage and feasibility of the requirement.</li> <li>• Early failures rather than going forward and fails.</li> <li>• Can resolve a lot of difficulties.</li> <li>• POC get to understand what will be delivered by us.</li> <li>• Convince the client to select us.</li> <li>• Make easy for communication.</li> <li>• Prototypes helps to get the project from the sales pipeline.</li> <li>• More paths to identify different options.</li> <li>• When developers are not experts on that area.</li> <li>• When we don't have clear picture what to be implemented.</li> <li>• To take technical decisions.</li> <li>• Understand complex UI functionalities by using</li> </ul>

		<p>wireframes and mockups</p> <ul style="list-style-type: none"> <li>• Identify limitations.</li> </ul>
	Risk mitigation and resolving conflicts	<ul style="list-style-type: none"> <li>• Resolve conflicts with the requirement and the understanding ambiguous and gray areas.</li> <li>• To develop right feature at once with better understanding what they need.</li> <li>• When not confident about the expected implementation and check the expectations before going for actual implementation.</li> <li>• Technical challenges to make sure it can be implemented and identify limitations.</li> <li>• Feedbacks are important with trial ones before actual implementation.</li> <li>• Take major decisions.</li> <li>• Client get to know what is going to be implemented.</li> <li>• Identify must have and nice to have features.</li> <li>• When the customer is not confident, can show we can achieve it.</li> <li>• Resolve customer's droughts what we can understand and what they want.</li> <li>• Sometimes customer learn from us based on our solution.</li> <li>• For larger projects make the life easier.</li> <li>• Proper estimations with reducing variations.</li> <li>• Set responsibility to the customer for the given requirement.</li> <li>• Early identifying impacted areas of the application</li> <li>• Customer engagement through POCs and Prototypes.</li> </ul>
Visualizing	Testifying ideas by visualization approaches	<ul style="list-style-type: none"> <li>• Prototypes, POC, UX designs, Mockups, wireframes.</li> <li>• Understand the functionality to the user.</li> <li>• Clickable HTML pages, clickable wireframes to show the system flow.</li> <li>• Flow charts, Process diagrams to visualize the flow.</li> <li>• Visualize real world existing examples and ask for changes.</li> <li>• Show existing implementation and ask feedbacks.</li> </ul>

		<ul style="list-style-type: none"> <li>• Based on the feedback do changes and show it to client and ask recommendations.</li> <li>• Activity diagrams and sequence diagrams to visualize the flow to the customer.</li> <li>• Show performance statistics taken through POCs.</li> <li>• Visualize the implementation going to be.</li> </ul>
	Stakeholder perception on visualized idea	<ul style="list-style-type: none"> <li>• Understand the solution going to be implemented.</li> <li>• Get user experience.</li> <li>• Make sure both dev team and client are in the same page.</li> <li>• The idea going to be implemented is clear.</li> <li>• Customer's mindset they learn from seen something.</li> <li>• Reduce the cost with showing actual implementation sprint by sprint.</li> <li>• Can clearly mention our idea and map with customer's idea.</li> <li>• Can understand the expectations for what extend.</li> <li>• Easy to communicate the idea to the customer and can get quick feedbacks.</li> <li>• Customer can understand the solution at glance.</li> <li>• Picture is a good tool to show something very clearly.</li> <li>• To remove negative view point.</li> <li>• Reduce the communication gap between customer, end-user and the project team.</li> <li>• Can stop the wrong implementation in early stages.</li> </ul>
	Practice of idea visualizing	<ul style="list-style-type: none"> <li>• To get end-user feedbacks with visualizing the idea.</li> <li>• When having unclear requirements and unclear user expectation.</li> <li>• Take the team onboard with expectations.</li> <li>• Side by side testing.</li> <li>• To get approvals for features.</li> <li>• Customer is not confident about the implementation.</li> <li>• To get an idea what is happening in the system.</li> <li>• Situations both customer and the team don't know what to be implemented.</li> <li>• Reduce the cost by going for actual implementation</li> </ul>

		<p>and getting feedback.</p> <ul style="list-style-type: none"> <li>• Understand the knowledge gap between project team and the customer.</li> <li>• Think more about the requirement.</li> <li>• When the solution is very complex.</li> <li>• For easy communication and get feedbacks.</li> <li>• To understand most suite for the customer.</li> <li>• Low fidelity to verify the idea communicated correctly.</li> <li>• HI fidelity get the conformation about the UI.</li> <li>• Reduce the communication gap between customer, end-user and the project team.</li> <li>• Through prototypes can understand the mismatching requirements with the customer.</li> <li>• When difficult to explain the understanding verbally and to take feedbacks.</li> </ul>
Synthesis of divergent and convergent approach	Idea generation	<ul style="list-style-type: none"> <li>• First idea is not always best.</li> <li>• Not confident with the initial idea.</li> <li>• Generate several solutions with different approaches.</li> <li>• Solution with different technologies.</li> <li>• Need several ideas for some situations to provide better solution.</li> <li>• To understand type of solution that can be given for a business case.</li> <li>• To think out of the box.</li> <li>• Go for multiple solution if the team don't have much experience of the area.</li> <li>• To get different solutions from the experts.</li> <li>• Multiple options to explain multiple possibilities.</li> <li>• Sometimes discuss multiple ideas within the team.</li> <li>• Not going forward if we have direct answer.</li> </ul>
	Evaluating ideas for best approach	<ul style="list-style-type: none"> <li>• Evaluate multiple ideas and take a decision.</li> <li>• Evaluate information and capabilities to choose the ideal solution.</li> <li>• Based on the feedbacks and merging some parts from other ideas, choose the best.</li> </ul>

		<ul style="list-style-type: none"> <li>• Choose best answer based on the brainstorm sessions.</li> <li>• Ignore the past pit-falls based on the experience and choose a solution.</li> <li>• Combine multiple ideas into one based on pros and cons.</li> <li>• Based on the strengths and weaknesses decide what is the best.</li> <li>• Not giving the technically best solution. Give most suitable solution for the business requirement.</li> <li>• Based on the business need and user experience choose one.</li> <li>• Team members do POCs come up with solutions. Based on that choose the best for the requirement.</li> <li>• Choose the best based on technology, performance, easy to implement etc.</li> <li>• Fish born method with voting choose the best.</li> </ul>
	Practicing divergent and convergent process	<ul style="list-style-type: none"> <li>• Select the technology.</li> <li>• Evaluate a problem.</li> <li>• Go for the best solution.</li> <li>• Suite for service based projects.</li> <li>• Research and development.</li> <li>• Not clear about the best way.</li> <li>• When there is no idea what is the best solutions.</li> <li>• For large projects to choose the best possible solution.</li> <li>• Provide solution for a global regional limitation.</li> <li>• To make innovations.</li> <li>• Customer decide the technology and ask for solutions.</li> <li>• Customer like more options expect to choose best fit to them.</li> <li>• Sometimes first idea goes to implementation. While developing facing troubles such situations.</li> </ul>
Collaborative work style	Brainstorming to justify decisions	<ul style="list-style-type: none"> <li>• Evaluation.</li> <li>• Multiple brains smarter then single brain.</li> <li>• Brainstorm to get expert ideas.</li> </ul>

		<ul style="list-style-type: none"> <li>• Pros and cons to make decisions.</li> <li>• Brainstorming the solution to be implemented.</li> <li>• Brainstorming with the team for clarification.</li> <li>• Developers think only technical perspective, BA focusing only business aspect therefore, only developer's solution might be too complicated to customer.</li> <li>• Collaborate to maintain harmony with tech teams.</li> <li>• When tight deadline project rather than take own decision take collective knowledge.</li> <li>• Spread the risk with the team without taking own risk.</li> <li>• Estimate the accurate work effort.</li> <li>• Groom the requirement.</li> <li>• Task breakdown.</li> <li>• Resolve code level issues and requirement related issues.</li> <li>• To get to know the unknown areas.</li> <li>• Get another perspective about the requirement.</li> </ul>
	Stakeholders in collaborative work style	<ul style="list-style-type: none"> <li>• Stakeholders of the project such as Product Owner, Scrum Master, Business Analysis, Project Team, Tech experts and Customer.</li> <li>• Technical specialize people to get consultancy.</li> <li>• Developers tell alternative approaches and feasibility of implementation.</li> <li>• QA share the impacted areas.</li> <li>• PO share the business needs.</li> <li>• Customer's technical team.</li> <li>• Onsite team.</li> </ul>
	Stakeholders involvement	<ul style="list-style-type: none"> <li>• Clarify unclear requirements.</li> <li>• Understand current process.</li> <li>• Difficult to understand what BA provided.</li> <li>• Make development fast.</li> <li>• Resolve practical issues while development.</li> <li>• Discuss team understanding and to get clarified.</li> <li>• Conflict and confuse situations.</li> </ul>



		<ul style="list-style-type: none"> <li>• Feedback for the presentations.</li> <li>• Conversations for finalize the requirement.</li> <li>• No need to blame individuals about wrong understanding of requirements.</li> <li>• Avoid missing requirement to make unhappy customer.</li> <li>• Domain expert consultants.</li> <li>• Technical people to get insights from the customer.</li> <li>• When needs clarifications and to get confirmation.</li> <li>• Discuss the requirements that clarifications are needed.</li> <li>• To take technical decisions.</li> <li>• Customer also have some ideas which are useful to find a solution.</li> </ul>
Project success and customer satisfaction	Pain points in satisfying customer expectations	<ul style="list-style-type: none"> <li>• Unable to prioritize what customer really wants</li> <li>• Compare and update the product features against competitor 's product without considering its applicability</li> <li>• Not focus on feature applicability for tomorrow market</li> <li>• Dealing with fix bid projects (budget, cost and time are fixed)</li> <li>• After completing the implementation only user realize feature is useless.</li> <li>• Requirement changes in a limited budget.</li> <li>• Dealing with very new technology.</li> <li>• Product develop without realizing target market.</li> <li>• Customer always expect to work only with domain experts and subject matter experts.</li> <li>• Micro manages by the customer</li> <li>• Lack of trust on vendor</li> <li>• ROI for the client for his investment.</li> </ul>
	Techniques to understand customer expectation	<ul style="list-style-type: none"> <li>• Conduct workshops at customer end to understand customer and end-user wish list</li> <li>• Develop roadmaps for implementation</li> <li>• Conduct reviews and demos (proof of concepts,</li> </ul>

		<p>prototypes)</p> <ul style="list-style-type: none"> <li>• Conduct frequent formal and casual discussions</li> <li>• Hire an end user to understand his/her pain points.</li> <li>• Gather and analyses customer pinpoints</li> <li>• Study the customer during pre-sales</li> <li>• Deliver small frequent deliverables with expected features</li> <li>• Communicate the blockers early as possible</li> <li>• Conduct market validation before starting product implementations</li> <li>• Educate client on simplicity on product implementation</li> <li>• Derive business case and form the foundation to the solution</li> <li>• Identify the non-functional requirements</li> <li>• Keep update customer about product implementation progress (Ex. Burn down charts, show and tells)</li> <li>• Study the drawbacks of the existing process</li> </ul>
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------