Developing Software Using Agile and Design Thinking Framework

Mahrukh Sameen Mirza
Software Engineering
University of Houston-Clear Lake
Houston, USA
MirzaM4179 @uhcl.edu

Soma Datta
Software Engineering
University of Houston-Clear Lake
Houston, USA
datta@uhcl.edu

Abstract—There are several software companies that develop several products. Are all of them used? Even after software companies following Agile framework methodology to develop their software's, many software's still fail to meet the requirements and eventually customers end up not using them. This is usually because the software teams fail to completely understand the user requirements. Agile framework and Design Thinking framework can be used together to gain customer empathy and to overcome this problem. This study discusses how Agile framework and Design Thinking supports the same principles and can be used to reduce software development failures. A Flashcard application was developed to validate Agile framework Design Thinking framework. The results are discussed in the validation and result section.

Keywords—Agile framework, Design thinking, software development, communication, prototype

1. INTRODUCTION

This section introduces the principles of Agile framework and Design thinking framework. From the introduction of Agile framework software companies have started using this methodology. Agile framework promises faster delivery, changing customer requirements, and gaining customer satisfaction [1]. Agile framework provides iterative and incremental process for developing software. When compared to planned software development, Agile framework has a smaller number of software failures because it includes changing customer requirements [2]. Agile framework is a people process, although it includes changing customer requirements, many software developed are still unused Customer do not use them because it does not target at solving the problem completely. This results in software use failures. Developing wasteful application results in wastage of a lot of resources including time, money, and workforce. To avoid such wastage, steps should be taken early in the development process.

Design thinking framework helps in creating great customer satisfaction [3]. Design thinking framework has several benefits associated to it including customer, better decision making, and innovation. It is

a way to gain empathy with the customer to find their needs [4]. Thus, a meaningful product can be build using design thinking. There are several stages in design thinking which includes – Empathize, Design, Ideate, Prototype and Test [3]. By conducting these steps, customer requirements and motivation to use the new solution can be identified. Design thinking helps the team to stay motivated and innovative [5]. It helps to reduce wasteful software which in turn helps to reduce wastage of resources. It brings the members of the organization together to tackle ill-defined and unknown problems.

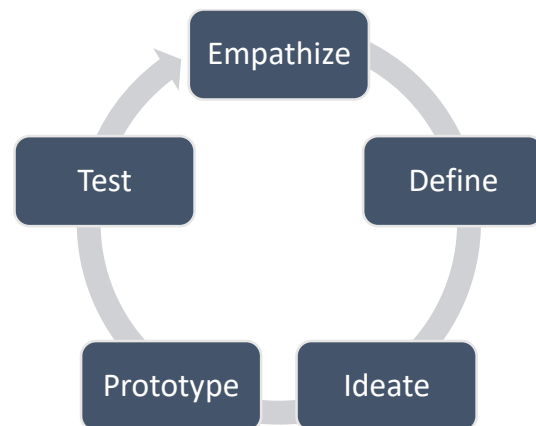Various stages of design thinking, and flow of the framework is shown in figure 1.

Figure 1. Stages of Design Thinking adapted from [5]

Empathize – this is the first stage of design thinking framework. Various methods are used to understand the customer's requirements.

Define – In this stage customer needs and problems are well defined in a formal manner. The information is analyzed and synthesized here.

Ideate – this is a fun stage where the team must think out of the box to find solutions of the problems defined in the previous stage. Here one creates as many ideas as possible.

Prototype – In this stage a minimum viable product (MVP) is created to test whether the customer is motivated to use this proposition rather than the present alternative. An MVP is created using inexpensive products.

Test – In this stage, the MVP prototype created is tested.

While Agile framework is a technique to solve a problem, design thinking is a technique to find what the problem is [6]. Do these frameworks support the same principles? Are they better together?

2. BACKGROUND

Several software's fail due to the lack of understanding the needs of the customer. Software failures result in the wastage of precious resources of the company including time, money, and energy. According to Liu [7], a recent study conducted on 5,400 IT projects revealed that 17 percent of the projects threatened the existence of the companies because of their failure. Over estimation, under estimation and not being able to satisfy the customer needs were reasons that were stated for the failure. The study reported that 45 percent of the projects were over budget, seven percent were not scheduled, and 56 percent delivered less value than expected.

Agile framework aims at customer satisfaction. Long term relationships with the customers can be built by producing customer-centric products and by completely satisfying the customer [8]. Customer satisfaction can be gained by understanding the customer requirements better and in detail. This has been identified as a central business factor [8].

Leem et. al [9] stated that continuous implementation of customer satisfaction has been hard for companies. Companies often fail to give continuous customer satisfaction to their customers. To help the companies do so, an evaluation system was developed by combining traditional software process assessment models with general customer satisfaction models. However, in this paper, the reason behind the companies failing to give continuous customer satisfaction was not specified.

Hence, research finds that one of the reasons for a software failure is not understanding the customer requirements clearly. Many software companies often end up building a product that does not satisfy the customer needs. There are companies who have successfully gained customer satisfaction but fail to maintain that over the years. Customers unsatisfied with the software often leads to losses including time, money and even the existence of the companies. It is known that many software companies now follow the Agile way of development. Agile is a framework that puts its customers first and strives to produce good customer experience. Although the companies are using Agile to develop their software, why are they not able to create good customer experience and produce software that will be used by the customers over present alternatives?

Design thinking is a framework that helps to understand customer requirements better even before the development starts. Understanding customer requirements eventually leads to customer satisfaction. Like Agile, design thinking is a people centric framework. In the methodology section, how design thinking and Agile support same principles is discussed along with how both help in reducing production of wasteful software is discussed.

3. RESEARCH QUESTIONS

How can we reduce developing wasteful software?

Do design thinking and Agile framework support the same principles?

4. METHODOLOGY

This section aims to discuss the research questions. It describes how Agile framework is the related to each stage of Design Thinking and using these frameworks wasteful software can be reduced. It also, relates the principles of design thinking and agile framework.

A. Relationship of Agile framework to Empathize stage

"Customer collaboration over contract negotiation" is one of the manifestos of Agile framework. The empathize stage of design thinking completely supports this manifesto. To develop a software that is useful to the customer and completely meets the customer requirements, regular communication with the customer is important. Empathize stage is all about interaction with the customer. Like Agile framework, empathize stage involves the customer to gain insights about his needs. Empathizing with the customer is done early in the development process to avoid production of wasteful software. Both Agile framework and empathize stage support face-to-face communication with the customer. Both believe in

bringing the team together which includes businesspeople as well as developers. Empathizing with the customer is not only the job of a designer but it is desirable for the whole team to come together. This helps to initiate communication with the customer and communication within the team. through this, teams become stronger, better, and also learn to value and respect each other. Hence, Agile framework and empathize support the same principles.

B. Relationship of Agile framework to Define stage

Individuals and interactions are prioritized over processes and tools in an Agile framework environment. User experience is created by collaborating with customers and within the team. New dimensions of a problem, innovative ideas for solving the problem and discussions can gain novel insights. The same happens in the define stage of Agile framework environment. After discovering the customer and his needs, discussions are conducted to create Point of Views and user stories. Apart from this, INVEST (Independent, Negotiable, Valuable, Estimable, Small and Testable) and MoSCoW (Must have, Should have, Could have and Won't have) can be used to write and sort user stories respectively. Together the team tries to think from the perspective of the customer. Just like Agile framework, in define stage continuous attention is given to create good design. In a way, define stage of design thinking framework completely supports the Agile framework principles.

C. Relationship of Agile framework to Ideate stage

Agile framework software development encourages customers to build software around motivated individuals. Individuals in a team can be motivated if their opinions are heard and valued. This also increases their confidence and gives a sense of contribution towards the project. In Agile framework, team members collaborate and trust each other to get the job done. Ideate stage of design thinking attempts of create a similar environment where team members can express their opinions and would not be judged. Through brainstorming, mind mapping and story boarding team members get a chance to listen each other and value each other's ideas. Both Agile framework and ideate stage value innovation and strives to create good team building skills.

D. Relationship of Agile framework to Prototype stage

Agile framework software development is responsive to changing market requirements. Market needs are never the same. Agile framework was chosen over planned software development because of its responsive nature. The prototype stage of design thinking is also responsive like the Agile framework software development. While prototyping, changes are welcomed if the customers change their requirements.

Agile framework prefers working software over comprehensive documentation. By developing minimum viable products (MVP), the team can show the customer something rather than mere documentation. Through this, the team tries to understand if the customer is really going to use proposed solution than the current alternative. Like KANO [10] model in Agile framework, that aims to gain customer satisfaction similarly MVP is used to gain customer satisfaction. Hence, prototyping stage is similar principle that Agile framework supports.

E. Relationship of Agile framework to Test Stage

Methodologies like test driven development (TDD) support testing along with development of a software. Testing is also an essential part of Agile framework. In planned software development, testing was done only after complete software was developed. But in Agile framework testing is not a phase, it is a continuous process which is done along with the development. Similarly, each prototype in design thinking is tested before the development starts. Testing is an integral part of the process in both Agile framework and design thinking.

5. VALIDATION AND RESULTS

To check if Agile framework and design thinking support the same principles and if development of wasteful software can be reduced by gaining empathy with the customer, an application ("Flashcards") was developed using C# and XML on Xamarin platform. It was developed combining both Agile framework and design thinking principles. All the stages of design thinking were carried out in five days starting from Monday to Friday.

Empathize – designing of this application started from empathize stage. Customer for this application was a graduate student at a university. Empathy maps and personas (which is a Agile framework techniques) was used to empathize after conducting a interview. The

interview was conducted using 4 W's (who, what, where and why) and 5 Y's (why is the customer, why has the customer, why has not the customer, why does not the customer and why is it a problem). By the end of the empathize stage, the problem of the customer was found out. It was that while the student found the practical courses easy, she found theory courses difficult and could not memorize them. Due to this, her grades were going down and she was disappointed.

Define – for defining the problem, user stories and child stories were created. INVEST [11] (Independent, Negotiable, Valuable, Estimable, Small and Testable) acronym was used to write the user stories and MoSCoW[12] (Must have, Should have, Could have and Won't have) was used to sort them. User story, INVEST [11] and MosCoW [12] are all Agile framework techniques

Ideate – storyboarding technique was used to carry out the ideate stage. Before and after situation along with additional notes were also created to create innovate ideas. After storyboarding, several ideas were written down in the brainstorming session. After brainstorming, three ideas were selected to be prototyped. The three ideas are as follows –

1. Write questions and answers for every unit and email the student
2. Use paper cards and let the student write questions in the front and answers on the back of the card
3. Use pictures related to the course content to help the student memorize

Prototype and test – all the above-mentioned ideas were taken to the prototype stage. The first ideas were prototyped but was not liked by the customer as this did not help her to read the course content. To prototype the second idea, a deck of cards was created using plain paper. These cards were given to the customer and she was asked to write her questions in the front of the card and answers in the back of the card while reading the course content. She was asked to use these cards for a day while learning her theory subject. She was able to add more cards and discard cards whenever needed. She was also able to edit the questions and answers on the cards. She liked what was prototyped. This did not require even one line of code and was done using readily available material.

Since this idea was most liked by the customer, we created a e-learning mobile application with flip cards in it. The application lets the user to write questions on the front of the card and write answers on the back of the card. Answers can be accessed by double tapping on the card. The user of this application was also able to remove, add and edit cards according to his/her desire. Next and previous cards could be accessed by using the next and previous buttons. The cards are saved automatically whenever a new card is created using the '+' symbol on the lower right of the application.

6. CONCLUSION

This study discusses, how Agile framework is related to different stages of design thinking. The study also discusses how both Agile framework and design thinking can together help to create good customer satisfaction and develop meaningful applications. This in turn helps in reducing the production of wasteful applications that are not used by the customer. To validate that both Agile framework and design thinking go hand in hand, an application called Flashcards was developed using C# and XML on Xamarin platform. The application uses flashcards to make learning easier. In this application, questions can be added on the front of the card and answers can be added on the back of the card. The card flips on double click. Previous and next cards can be accessed by clicking on the previous and next buttons respectively. New card can be added by clicking on the '+' button. These cards can be edited and deleted as needed by the user. Development of this application was done by following the Agile framework and design thinking principles. In the future, this methodology will be used in 'Agile development' classes and for the capstone classes.

REFERENCES

[1]   F. Anwer and S. Aftab, "SXP: Simplified Extreme Programing Process Model," *Int. J. Mod. Educ. Comput. Sci. Hong Kong*, vol. 9, no. 6, p. 25, Jun. 2017, doi: http://dx.doi.org/10.5815/ijmecs.2017.06.04.

[2]   M. Sameen Mirza, S. Datta, and S. Datta, "Strengths and Weakness of Traditional and Agile Processes - A Systematic Review," *J. Softw.*, vol. 14, no. 5, pp. 209–219, May 2019, doi: 10.17706/jsw.14.5.209-219.

[3]   B. R. Ingle, *Design Thinking for Entrepreneurs and Small Businesses Putting the Power of Design to Work*, 1st ed. 2013. Berkeley, CA: Apress, 2013.

[4]   R. Razzouk and V. Shute, "What Is Design Thinking and Why Is It Important?," *Rev. Educ. Res.*, vol. 82, no. 3, pp. 330–348, 2012, doi: 10.3102/0034654312457429.

[5]   "What is Design Thinking?," *The Interaction Design Foundation*. https://www.interaction-design.org/literature/topics/design-thinking (accessed Feb. 07, 2020).

[6]   "Design Thinking vs. Agile: Combine Problem Finding & Problem Solving," *Mendix*, Oct. 25, 2017. https://www.mendix.com/blog/design-thinking-vs-agile-

combine-problem-finding-problem-solving-better-outcomes/ (accessed Feb. 04, 2020).

[7]   S. Liu, "Engineering the Success of Software Development," *IT Prof.*, vol. 15, no. 5, pp. 4–5, 2013, doi: 10.1109/MITP.2013.76.

[8]   S. Dalal and R. Chhillar, "Empirical study of root cause analysis of software failure," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, pp. 1–7, 2013, doi: 10.1145/2492248.2492263.

[9]   C. Leem and Y. Yoon, "A maturity model and an evaluation system of software customer satisfaction: the case of software companies in Korea," *Ind. Manag. Data Syst.*, vol. 104, no. 3/4, pp. 347–354, 2004, doi: 10.1108/02635570410530757.

[10]   G. Lubinski and A. Oppitz, "Applying the KANO Model in Mobile Services World: A Report from the Frontline," Lisbon, 2012, pp. 161–164, doi: doi: 10.1109/QUATIC.2012.21.

[11]   L. Buglione and A. Abran, "Improving the User Story Agile Technique Using the INVEST Criteria," 2013, pp. 49–53, doi: 10.1109/IWSM-Mensura.2013.18.

[12]   R. Popli, N. Chauhan, and H. Sharma, "Prioritizing user stories in agile environment," 2014, pp. 515–519, doi: 10.1109/ICICICT.2014.6781336.