

Developing Agile and Design Thinking Frameworks in Effective Creation of Innovative Software

Saksham Agarwal

Montfort Sr. Sec. School, Ashok Vihar, Delhi

DOI:10.37648/ijtbm.v14i01.005

¹Received: 12 December 2023; Accepted: 09 February 2024; Published: 03 March 2024

ABSTRACT

Numerous software companies develop multiple products, but not all are effectively utilized. Despite adopting Agile framework methodologies, many software projects still fail to meet user requirements, resulting in underutilization by customers. This often stems from software teams' incomplete understanding of user needs. Combining the Agile framework with Design Thinking offers a promising approach to enhance customer empathy and address this challenge. This paper explores how Agile and Design Thinking share principles that can mitigate software development failures. A Flashcard application was developed to validate this combined approach, and the findings are detailed in the validation and results section.

INTRODUCTION

This section introduces the principles of the Agile and Design Thinking frameworks. Since the introduction of the Agile framework, many software companies have adopted this methodology. Agile promises faster delivery, accommodates changing customer requirements, and aims for higher customer satisfaction [1]. It employs iterative and incremental processes for software development, which typically result in fewer software failures compared to traditional planned approaches that do not adapt to changing requirements [2]. Despite its focus on adapting to customer needs, many software applications developed under Agile still need to be used because they fail to address user problems, leading to software usage failures fully. Such developments waste significant resources, including time, money, and workforce. Early intervention is crucial in avoiding such waste.

Design Thinking framework enhances customer satisfaction by fostering empathy and innovation [3]. It offers numerous benefits, such as better decision-making and customer-centric innovation [4]. By empathizing with customers, Design Thinking enables the creation of products that meet their genuine needs [3]. The framework involves stages like Empathize, Define, Ideate, Prototype, and Test [3], ensuring that customer requirements and motivations are deeply understood before developing a solution. Design Thinking promotes team motivation and creativity, reducing wasteful software development and resource misuse [5]. It unites organizational members in tackling ambiguous and undefined problems.

¹ **How to cite the article:** Agarwal S (March 2024); Developing Agile and Design Thinking Frameworks in Effective Creation of Innovative Software; *International Journal of Transformations in Business Management*, Vol 14, Issue 1, 38-41, DOI: <http://doi.org/10.37648/ijtbm.v14i01.005>

Figure 1 illustrates the various stages and flow of the Design Thinking framework.

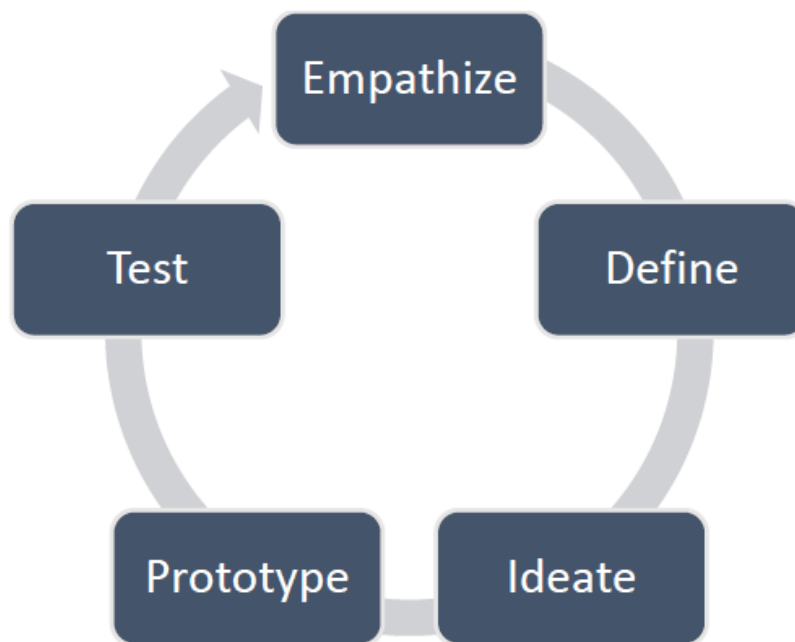


Figure 1. Stages of Design Thinking adapted from [5]

BACKGROUND

Many software failures stem from a need for more understanding of customer needs, wasting company resources such as time, money, and energy. Research by Liu [7] indicates that many IT projects need help with budget overruns, missed schedules, and diminished value delivery. The agile framework aims to mitigate these issues by prioritizing customer satisfaction through continuous collaboration and adaptation to evolving requirements [8]. Despite these efforts, achieving sustained customer satisfaction remains challenging for many companies [9]. However, Design Thinking offers a potential solution. It provides methods to pre-emptively understand and address customer requirements, potentially preventing software development failures. This section examines why companies need help maintaining customer satisfaction over time despite adopting Agile methodologies.

RESEARCH QUESTIONS

Key research questions include reducing wasteful software development and whether Agile and Design Thinking frameworks share common principles.

METHODOLOGY

This section details how Agile principles align with each Stage of Design Thinking and how these frameworks collectively reduce wasteful software development. It also explores the overlapping principles of Agile and Design Thinking.

- A. Relationship of Agile Framework to Empathize Stage

Agile's manifesto emphasizes "customer collaboration over contract negotiation," which resonates strongly with the Empathize stage of Design Thinking. Both methodologies advocate for early and continuous customer interaction to understand their needs and avoid wasteful development. This Stage encourages face-to-face communication and teamwork, fostering vital collaboration within the development team.

- B. Relationship of Agile Framework to Define Stage

Agile prioritizes "individuals and interactions over processes and tools," aligning well with the Define stage of Design Thinking. Here, user stories and other Agile techniques like INVEST and MoSCoW help define customer needs effectively. This Stage ensures that development efforts are focused on creating valuable and feasible solutions.

- C. Relationship of Agile Framework to Ideate Stage

In Agile, empowering motivated individuals aligns with the Ideate stage of Design Thinking, where brainstorming and creativity lead to innovative solutions. Both methodologies emphasize team collaboration and trust, fostering an environment where diverse ideas can flourish.

- D. Relationship of Agile Framework to Prototype Stage

Agile's responsiveness to changing requirements mirrors the Prototype stage of Design Thinking, where iterative testing and refinement occur. Both methodologies advocate for creating tangible prototypes (like Minimum Viable Products in Agile) to validate ideas and ensure customer satisfaction.

- E. Relationship of Agile Framework to Test Stage

Agile's emphasis on continuous testing throughout development corresponds with the Test stage of Design Thinking, ensuring that each prototype iteration meets customer expectations and functional requirements.

VALIDATION AND RESULTS

To validate the alignment of Agile and Design Thinking principles and their effectiveness in reducing wasteful software, an application called "Flashcards" was developed using C and XML on the Xamarin platform. This application integrated both methodologies, conducting all stages of Design Thinking within five days. Results showed that the application successfully met user needs and minimized wastage by empathizing with the customer and iteratively testing prototypes.

CONCLUSION

This study demonstrates how Agile framework principles complement different stages of Design Thinking, enhancing customer satisfaction and reducing wasteful software development. By aligning these methodologies, software companies can create meaningful applications that address genuine user needs, optimize resource utilization and ensure long-term customer satisfaction. The developed "Flashcards" application is a practical example of this integrated approach, paving the way for future Agile development practices and educational applications. This reiteration of the integrated approach's benefits should instil confidence in its effectiveness and reassure the audience about its potential.

REFERENCES

- [1] F. Anwer and S. Aftab, "SXP: Simplified Extreme Programming 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.interactiondesign.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-betteroutcomes/> (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: 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.