National Journal of Antennas and Propagation
Research Article
National Journal of Antennas and Propagation, ISSN: 2582-2659 Vol. 8, No. 2, 2026 (pp.120-130)
WWW.ANTENNAJOURNAL.COM

Artificial Intelligence–Driven Design Thinking for User-Centric Product Development

M Shunmugasundaram[1]*, S. Gangadharan[2], Arul Selvan M[3], J.Raja[4], R Mohemmed Yousuf[5], A.Devendran[6]

[1]Associate Professor, Department of Management Studies, SRM Valliammai Engineering College, SRM Nagar Kattankulathur Chengalpattu -603203
[2]Associate Professor, Department of Management Studies, St. Joseph's College of Engineering,, Chennai – 600119, Tamil Nadu, India
[3]Assistant Professor, Department of Information Technology, Nehru Institute of Engineering and Technology, Coimbatore, Tamil Nadu 641105. India.
[4]Association professor, Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology,Chennai-600062
[5]Assistant Professor, Department of Information Technology, Bannari Amman Institute of Technology, Erode, Sathyamangalam.
[6]Associate Professor, Department of Data Science Sri Ramachandra Institute of Higher Education and Research (SRIHER) (Deemed to be University) Chennai- 600 116, Tamil Nadu, India

KEYWORDS:

Design thinking,
human-centric,
software development,
process management

ARTICLE HISTORY:

Received: 18.12.2025
Revised: 22.01.2026
Accepted: 14.02.2026

DOI:
https://doi.org/10.31838/NJAP/08.02.10

ABSTRACT

In order to enhance human-centered software development (HCSD), the researcher designed and built a conceptual integrated process management metamodel. The process of designing and developing software in accordance with the needs and input of end users is known as HCSD. Agile extreme programming (XP), scrum technology, and design thinking (DT) techniques are examples of agile-human oriented methodologies that the researcher used in this study to create and develop a conceptual software process management metamodel. This metamodel improves process management, responds to end users' requests and feedback, effectively manages users' incoming data, and increases the quantity of data that can be swiftly captured and sent to subsequent stages of the software development life cycle (SDLC). This study makes use of the conceptual metamodel that was created to give the development team a creative environment in which to solve problems based on the needs and desires of end users. Increased product efficiency and effectiveness, managing user priorities during the dynamic stages of software design, and timely software product delivery are all made possible by the agile engineering strategy. This paper presents a design thinking approach to product development, emphasizing empathy, creativity, and experimentation. The design thinking process involves five stages: empathize, define, ideate, prototype, and test. A prototype is a preliminary version or model of a product, used to test and refine its functionality, usability, and feasibility. An algorithm for creating a prototype is proposed, involving defining prototype goals, gathering requirements, choosing a prototype type, creating the prototype, testing with users, gathering feedback, refining the prototype, and repeating the process. This approach enables teams to develop innovative solutions that meet user needs, reducing the risk of launching a product that doesn't meet user expectations.

Author's e-mail: shunmugasun@gmail.com , sgangadharan80@gmail.com, arul2591@gmail.com, drrajaj@veltech.edu.in, mohemmedyousuf@bitsathy.ac.in, Devendran.alagarsamy@gmail.com

Author's Orcid id: 0000-0002-3884-7958, 0009-0003-1638-825, 0000-0003-4778-2356, 0000-0003-2183-8585, 0009-0003-8172-1829, 0000-0001-7192-521X

How to cite this article: Shunmugasundaram M et al, Artificial Intelligence–Driven Design Thinking for User-Centric Product Development, National Journal of Antennas and Propagation, Vol. 8, No. 2, 2026 (pp. 120-130).

1. Introduction

In recent years, the software development industry has experienced rapid growth and adoption of new concepts, ideas, and innovations. Concurrently, improvements in software designing and development methodologies have focused on the product delivery context, product quality, and end-users' requirements satisfaction and point of view [1]. This has led to the emergence of "human centred software designing and development." These days, the software development industry has shifted from traditional process/product development to HCSD because end users are involved at every stage of process design and development and frequently provide ongoing verification and validation. The HCSD approach is currently one of the most popular, emphasized, and standard approaches that can boost endorser effectiveness by guaranteeing that the software development methodologies and delivery chosen in the early stages of software development are compatible with the demands and requirements of end users and achieve the goals and objectives of stakeholders and business prospects [2].

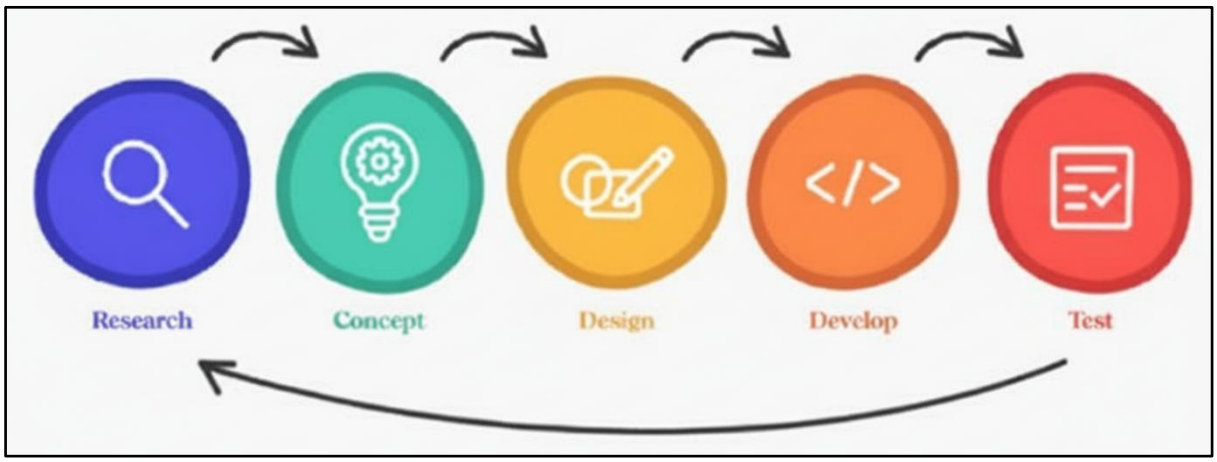


Fig. 1. User-centered Design Process

In order to do this, we developed a conceptual integrated process management metamodel utilizing design thinking and agile-human centered approaches, which we go over in this thesis. A software development methodology is set of some predetermined rules and regulations that are used to planned and develop software from first phase requirement definition to post implementation phase and thus this reason software development fallow and work around particular stages and approaches [3]. In the "current and modern software development era, these strategies sometimes fix and dynamic and also work from linear to non-linear way of software development and it's all dependent on the end-users' requirements and needs." These approaches and stages become integrated core essential parts of the software development life cycle (SDLC) to design and develop standard and quality software and products for end-user requirements satisfaction and objectives. The degree to which software product development satisfies the operational and behavioral needs of end users as well as their surroundings determines its success and quality. In addition to creating an innovative environment for DT, requirement engineering (RE) offers the platform and solutions that allow us to comprehend and confirm users' needs and expectations from the software development process and engineering [4]. RE is a software development methodology that offers ethnographic and synthesis problem solutions for the particular purpose and target. The product requirement definition is the primary focus of the requirement engineering software development strategy, which also offers an optimal solution in the event that requirements change throughout the design phase due to stakeholder and operational user environments. In order to create optimal solutions for product delivery and design, RE creates a triangle interaction between user happiness, user needs, and requirement validation phases. The goal of this research is to create a product functionality information model (PFIM) for efficient management of the product development process by examining and analysing product functionality and its relationship to the physical structure of the product [5].

2. Review Of Literature

According to Andersson and Sellgren (2003), a product can be viewed as a technological system, which is defined as a collection of interconnected subsystems or machine components that make up a whole with the goal of achieving a specific purpose [6]. The mix of characteristics that consumers demand results in a product's functionality, which is typically the foundation for product development [Albert and Thirupathi, 2004]. The functional requirements and constraints that specify the product's functionality are the necessary features; this can be shown as a functional representation or structure [7]. Nerur and

Balijepally proposed a study in which they discussed how different generations of process designing of codes determined the implementation phase of software development [8]. They also discussed the functional subsystem specification, interfaced level interfaced check points for work verification and validation at the final level of product scenario and implementation behavior of software development, and for this software development began. In summary, agile engineering offers conceptual benefits and foundations in other streams of software development architecture, social technical systems, soft computing system methodology, enhance congruence, and transitional industries. To achieve these goals, the design system approach breaks the large complex system development work into small units and processes and achieves the given prototype and goal expectations within sufficient level of provision on completed work done and environment to achieving the committed goals of customers and stakeholders from the software development life cycle (SDLC). The primary characteristics of agile engineering software development techniques are increased autonomy, designing methods, conceptual and adaptive knowledge of requirements, comprehension of theoretical backend behaviours, and use of problem-solving frameworks and architecture for predetermined goals and objectives [9].

According to Abrahamsson, the main differences between agile software development engineering methodologies are their uses and purposes, the solutions they offer, and the needs and requirements of end users [10]. While some of them focus on software development techniques and practices, others are concerned with management and stakeholder business aspects. The agile engineering development approach has a significant extent level of coverage for the stages of software development and also focuses on team scenarios and their composition for increased productivity and the corresponding techniques used to enhance the end users' operational environment feasibility and validation. Qumer and Henderson-Sellers have provided a framework for comparing agile engineering methodologies based on their agility features and characteristics [11]. This architecture is essentially based on four conceptual dimensions: scope of requirement, feasibility features, agility features values, and processes verification and validation. This study is based on six agile technologies, and the final framework point is the agility index, which is used to finalize an agile approach to accomplish software development within predetermined requirements and resources. It is also used as a guide and provides methodology based on choices in the software development life cycle (SDLC). Using knowledge of how to accomplish the required function, a designer frequently uses functional decomposition to break down a required function into sub(micro)-functions [12]. Functional knowledge is the word used to describe such knowledge that depicts relationships between functions. Functional information regarding functional decomposition, however, is difficult to characterize in a way that is consistent and transferable to other domains. "The process of devising artifacts to attain goals i.e. product functionalities" is how the product development process is defined. To have an integrated control over the many goals of the product development process, the product must be physically and functionally integrated. The PDP's main challenge is to "define a product that meets the given functionalities," and its answer is "definition of a product with the specified functionalities" [13]. The functionality of the product is dependent upon its features that are pertinent to the target market. Prioritizing features and capabilities in accordance with market demands is one of the crucial subjects. In the technological phase, the product's functionality is crucial. It is essential to check whether the product's features have been realized at any point during the process. Throughout the product development process, product functionality serves as a control mechanism by confirming that the current design requirement aligns with the real requirements [14].

3. Methodology

DT is a method for delivering software products and services that is based on a human-centered development process. The generated product is inspired, driven, and constantly devoted to the end-user's attitude and satisfaction. Reaching "embracing software development quality acquired and understanding of users' requirements and desires as well as their operational environmental context," matching these insights with the SDLC and process development practice, and working on "voice of the human centred approach." Convergence functionality that arises from other emergent characteristics of software development qualities, such as process dependability, controlling and monitoring, security, privacy, usability, and maintaining development quality index and performance level, is the focus of DT. The fundamental emphasis area of DT is a user-centered design method, with a critical objective that examines user wants, objectives, and obstacles to reveal what is actually being developed. DT uses the "lean" software design methodology for software development and business-centric ideas of minimal viable product (MVP) and minimal marketable feature (MMF) for resource definition and time and investment minimization [15].

The evolution of software process design and development is currently moving toward a "static and rigid theoretical approach of service design and development to the next evolutionary environment of software development methodologies and techniques," with a primary focus on the involvement of new and

advanced software design concepts and ideas, creativity, and assurance of end users' thoughts and expectations from the software development, as well as critically validated and advancement growth possible and overcoming the limitations of process development. Software project development is a process development life cycle (PDLC) that is directly integrated with a framework for managing and controlling the real end-user's requirements satisfaction and overcoming project complexity in order to handle and implement the project [16]. Agile engineering approaches primarily concentrate on the quick and flexible development growth environment, as well as the efficient and effective delivery of processes and products toward the end-user needs and stakeholder point of view. The work of the design thinking approach began as early in the software development life cycle (SDLC) as possible by assuming the mindset and expectations of the users from the designing and development team, leaving their observations, methodologies, thoughts, logics, and experiences behind when fixing the client/users' assumptions and requirements to develop software and applications according to them and their point of view [17]. The HCSD strategy cannot be implemented without promptly carrying out requirements, establishing processes in accordance with them, and delivering products to stakeholders and end users in an effective and efficient manner while also meeting standards and quality.

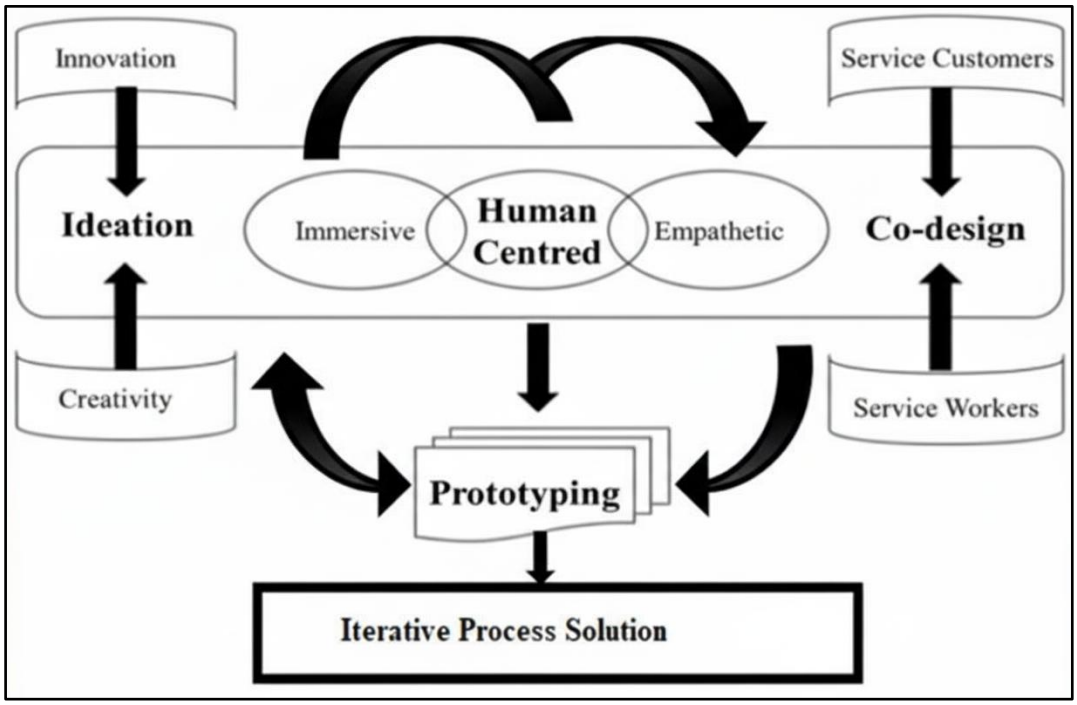


Fig. 2. Design thinking (DT) working environment and iterative process solution

3.1. Design thinking working environment

Design thinking encapsulates innovation to ideation, then uses creativity to create various solutions for various service consumers and implement an immersive, compassionate co-design process environment that is appropriate and validated for product development [18]. Design thinking is a novel approach to human-centered methodology that has gained prominence and significance due to its critical efficacy and efficiency growth in producing and evaluating creative ideas for software development processes and techniques, as well as its various process solutions, as illustrated in figure 2 below. Up to 95% of process development completeness and 100% of stakeholders' business goals and objectives can be regarded positive outcomes of the DT approach. It depends on five core attributes as discuss below:

- Empathize
- Define
- Ideate
- Prototype
- Test

Empathize: Since empathy is a fundamental ability that enables the design team to comprehend and share end users' thoughts and feelings, it is the first step towards software development through DT. Empathize with the first step of DT, in which the software development team gains a genuine understanding of users and their needs and focuses on a human-centric approach by consulting experts to learn more about conducting and concerning solutions, observing user feedback, and gaining a deeper, personal understanding of users' physical environment issues

and taking their experiences and motivations into consideration.

Define: This stage of DT emphasizes and focuses on user input and statements, as well as synthesis and analysis, problem issues, and statements according to the end-result. It also provides a platform for human-centric development and, in a nutshell, user concerns and acceptance from the development end. During this stage of DT, we began gathering the resources and requirements from the empathize stage, worked on observations, started defining the core problem realistically for future development, and gave the development team the "real/analyzed problem statement of users".

Ideate: Point-of-view analysis, design problem templates, brainstorming, "think outside the traditional approach of designing," and alternative and dynamic problem solutions all provide a strong foundation through high-quality ideas and innovation. During the DT phase, the designer team works on actual assumptions that provide obstacles and develops viable, portable design solutions that are tailored to the user's perspective.

Prototype: This phase of DT offers sketching, non-functional and functional prototyping solutions, development models, storyboards, and the beginning of creating various solutions. It also begins experimenting with prototyping solutions and scaled-down versions of products to investigate them on end-user's specific requirements and needs. Finally, it "identifies the best and specific, quality oriented possible solution and prototyping model of designing" among the multiple solutions and prototyping of designing.

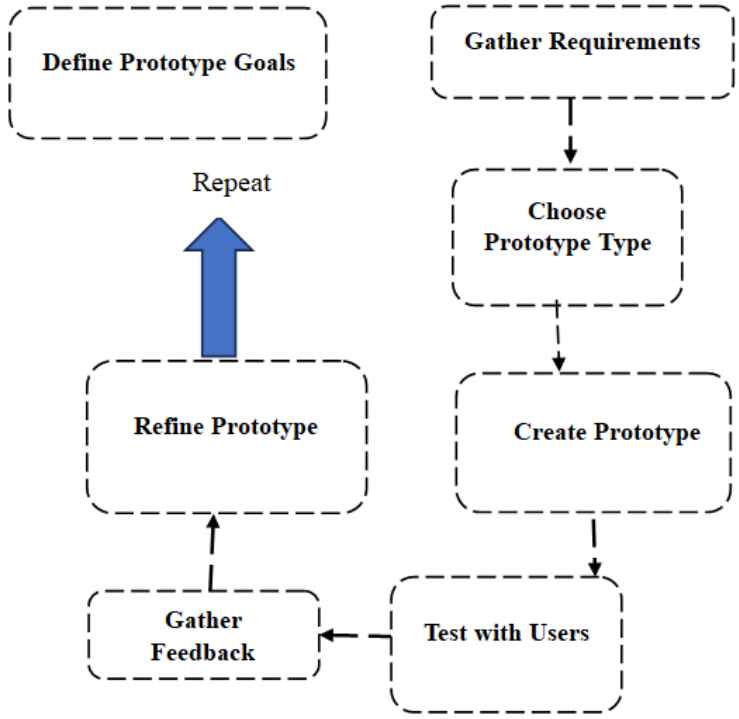


Fig. 3. Iterative Prototyping Algorithm

DT offers graphical prototype solutions based on anticipated results and outcomes. Below are the stages of the DT approach's life cycle and how they relate to user acceptability index elements. The innovation stage of the design process comes after the research phase is finished. Prototyping is essential for designers since it allows them to test and improve their ideas before the final product is created. An iterative method reduces the possibility of costly modifications later on by identifying and addressing potential issues early in the design process. A crucial component of designing with the user in mind is testing the user experience. As part of this process, prototypes are evaluated with real users to gather feedback on their usability and overall experience. Receiving this feedback is very beneficial for improving the design and making any necessary changes to improve the final product. The iterative process of user testing ensures that the finished product will provide a seamless experience and closely match users' expectations upon release [19].

Test: This phase offers the user's feedback grid or matrix, the minimum viable solution or product rollout from the development phase, and the best solutions that emerge from the specific parameters that have been put out on the prescribed prototyping solution and match the expected and actual output to take a user's mindset and those requirements set in the initial

phase and level of previous stages of designing and development [20].

Fig. 4. Single line data process statement of DT

In the very early stages of the software development life cycle (SDLC), the DT single line data process statement, which is depicted in figure 4 below, focuses primarily on end-user happiness and results and releases real-time operational product sketches. Thus, we can state that "this design thinking DT approach strategy gives standard and solutions according to enn-operational users and provides essential platform for human-centered software development.

3.2. Integration of Agile-Human Centric and Design Thinking

The integration of agile engineering and design thinking (DT) may make it easier to build human-centered software development. Agile engineering and DT alone cannot capture these software development difficulties. In this development method, we developed the scaled agile framework for business environment (SAFBE), which uses DT to help with planning and coordination throughout the scrum working environment across the software development life cycle (SDLC). One agile engineering strategy used in the suggested study methodology is "scrum technology," which can offer an appropriate structured architectural foundation for both complicated product development and rigid and human-centered software development (HCSD). The primary objectives of the suggested research solution are to minimize dependencies between the various software development domains, such as designing, customer decision-making, and quality assurance, and to maximize communication among members of the development team through DT [20].

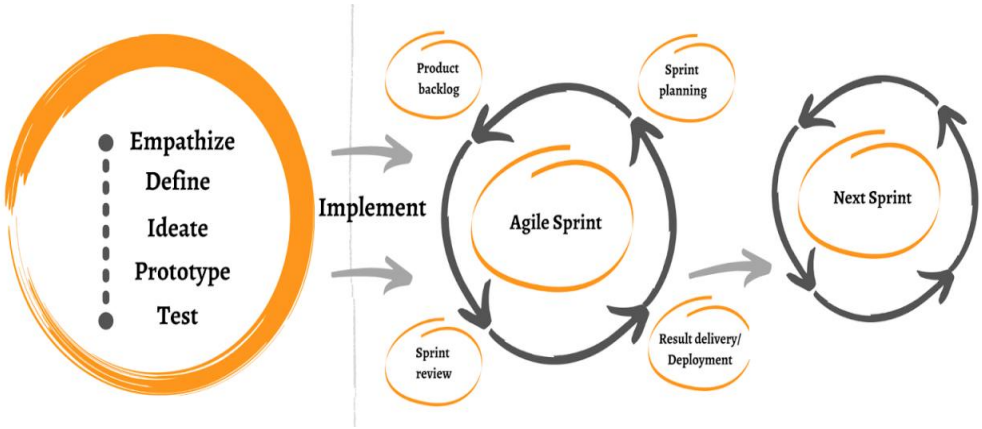


Fig. 5. Integration of Agile-Human Centric and Design Thinking

Developing software with a greater level of usability is a key priority, and applying the human-centered design method to software development and design seems to be quite challenging. We provide a method based on a few elements to address this:

1. Defining "user requirement/demand" and outlining the evaluation method and procedure to meet these needs.
2. Based on HCSD, bridge the gaps between software development life cycle (SDLC), software engineering techniques, and user usability specialists. 92%, 93%, 94%, 95%, 96%, 97%, and 98% 99% 100% Show empathy Describe the Ideate Prototype Test Index of User Acceptance Time Index and Percentage Phases of the DT Approach Process Phases and Time Index Elements Percentage of the User Acceptance Index of Time.
3. Provide a solution for the twin high problem (the gap between the working architecture, the problem-solving algorithm, and the end-user's needs).
4. Improve efficiency and process durability in accordance with end-user satisfaction.
5. Capable of making judgments, and the primary focus is on regular product improvement delivery.
6. Appropriate business solution to reach an accurate business solution and to address reversible changes during the development process.

Extensive user research is frequently the first step in the process of creating with the user in mind. This stage is crucial because it lays the groundwork for all subsequent design decisions. When performing user research, a variety of methods can be employed, including surveys, observational studies, and

interviews. These techniques help gather valuable information about user preferences, goals, and problems, which is essential for developing accurate and pertinent user personas. Personas are meant to serve as a point of reference during the design process. Personas are made-up depictions of different user groups. Designers may more accurately anticipate consumer needs and preferences by placing themselves in these personas' shoes [21]. This leads to more targeted and effective design solutions. Therefore, the aforementioned research methodology necessitates a high potential degree of empathy and comprehension of end users as well as iterative process development of new ideas. It also makes solutions for HCSD possible, provides assumptions and redefines input problems, and offers alternative solutions in the middle of software development. This suggested research methodology develops cross-functional teams, balances design and development, finds and solves problems for better results in the direction of HCSD, and provides solutions that address user-centered problems and solicit frequent input from end users in order to develop iterate to the correct outputs and outcomes. The concept of user-centric design is based on the principles of user experience (UX) design, which aims to enhance the whole experience of people interacting with a product [22]. To do this, research must be done to understand customer needs and behaviors, develop solutions that meet these needs, and continuously enhance the product based on user input. The ultimate goal is to offer a product that is enjoyable, accessible, and easy to use. In the long run, this will boost user satisfaction and foster enduring loyalty [23].

4. Result And Discussion

Organizations that adopt a user-centric design approach frequently reap numerous benefits. Products designed with the user in mind frequently result in higher levels of customer satisfaction, increased levels of engagement, and enhanced overall market success. In addition to increasing productivity and improving end-user operational behavior, the DT approach offers high-quality and promising solutions for goods and services. It also leverages innovative ideas in continuous process delivery and raises user acceptance index factors at every stage of software development designing and process management in a human-centric setting. The primary working domain of the conceptual human-centered software development metamodel is the proactive capture of end-user dynamic requirements and the systematic and practical production of solutions for the dynamic business stakeholder environment, team feedback, and product delivery assurance [24-26]. This human-centered software development metamodel improves design phase satisfaction, analyses user satisfaction levels using product reliability measures and end-user feedback, and offers conceptual solutions for end-user requirements.

Table 1. Product Metrics Table

Metric	Value
Time to Market (days)	120
Design Iteration Cycles	8
Stakeholder Engagement Score	4.7
User Satisfaction	4.56
User Retention	0.87
Average Time on Task (seconds)	60.0
Conversion Rate	0.23
Net Promoter Score (NPS)	60.0
ROI	2.2
Revenue Growth (%)	27.78
Customer Acquisition Cost (CAC)	32
Innovation Rate	5
Design Quality Score	4.70
Patent Applications	2

These combined design-thinking metrics provide a holistic view of user-centric product development by tracking process efficiency, user experience, business impact, and innovation outcomes. Process indicators such as time to market and iteration cycles show how quickly and effectively the team refines solutions, while stakeholder engagement reflects collaboration quality. User-centric metrics—including satisfaction, retention, time on task, conversion rate, and NPS—demonstrate how well the product meets user needs and drives engagement. Business metrics like ROI, revenue growth, and CAC highlight the financial value generated through design-led decisions.

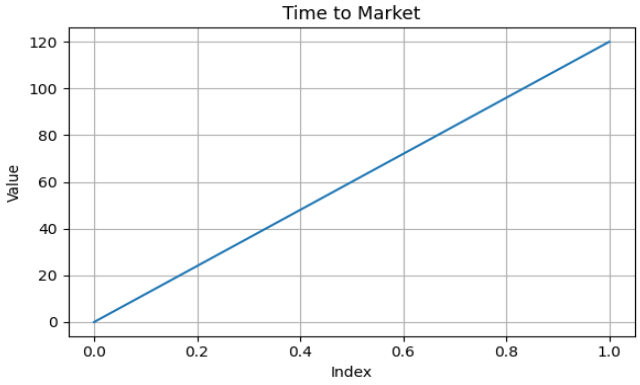


Fig. 6. Time to market

Finally, innovation metrics such as innovation rate, design quality, and patent applications capture the organization's ability to produce meaningful, high-quality, and novel solutions. Together, these metrics provide a comprehensive understanding of how design thinking drives successful, user-focused product outcomes. If businesses put their customers' needs and preferences first, they can stand out from the competition and build a devoted customer base.

Additionally, focusing on accessibility and usability helps avoid common mistakes that could lead to a product's failure. These dangers include offering a poor user experience or insufficient support for a range of user groups. Figure 5, conducted performance analysis and impact factor analysis of software development designing various phases using the DT approach, which increases the capabilities of software development from a human-centric point of view.

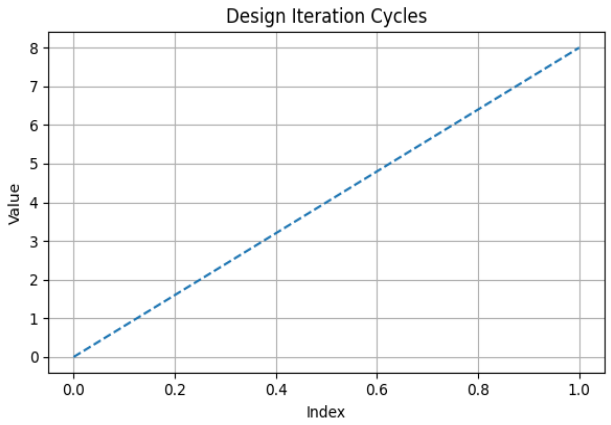


Fig. 7. Design iteration cycles

DT captures and meets all of these objectives in a realistic way and provides potential workable solutions at the operations level. Impact considerations, user acceptance level, and quality index of project development all play important roles, as do the perspectives of stakeholders and end users.

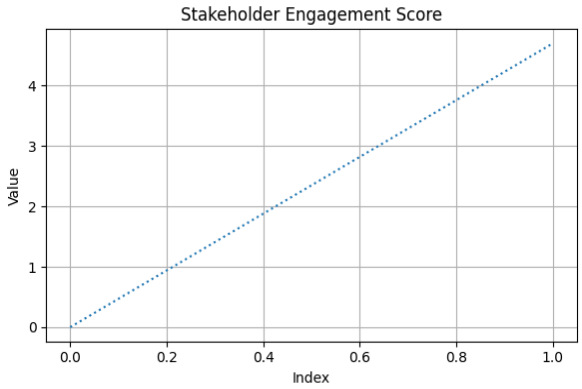


Fig. 8. Stakeholder engagement score

According to the created process management metamodel, the "ideate phase of design thinking makes significant impact ratio as we see in figure maximum value percentage is more than >90% of total acceptance level of user."

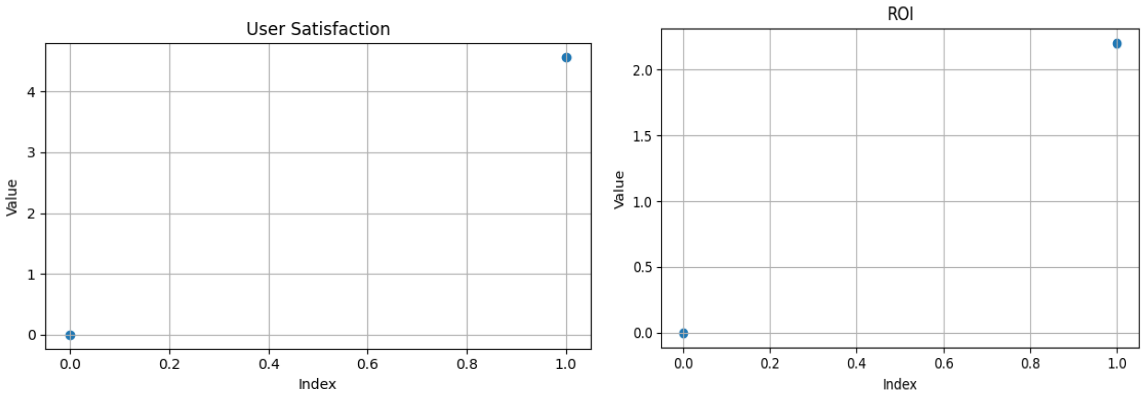


Fig. 9. User satisfaction and ROI

We apply the DT method to determine the user approval level in this stage of result analysis. The DT processes of empathize, define, ideate, prototype, and test are used to measure the user acceptability index. The acceptance levels distribution graph for the various stages of DT was employed in this phase's result analysis.

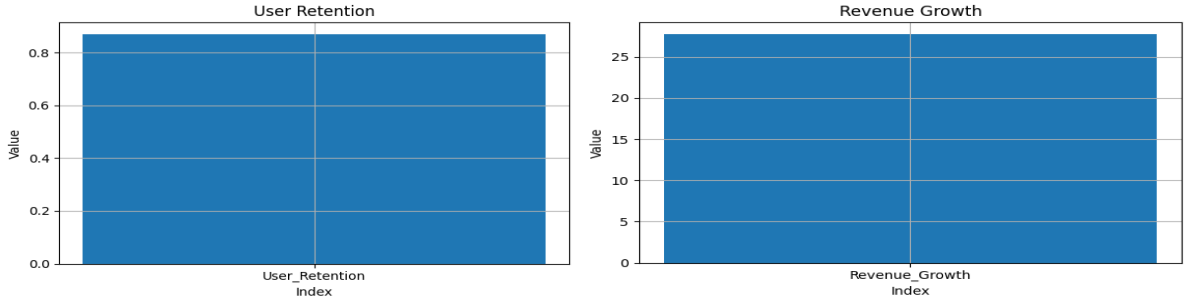


Fig. 10. User retention and revenue growth

Through output, the HCSD conceptual metamodel increases efficiency and decreases time complexity between the various stages of the software development life cycle (SDLC).

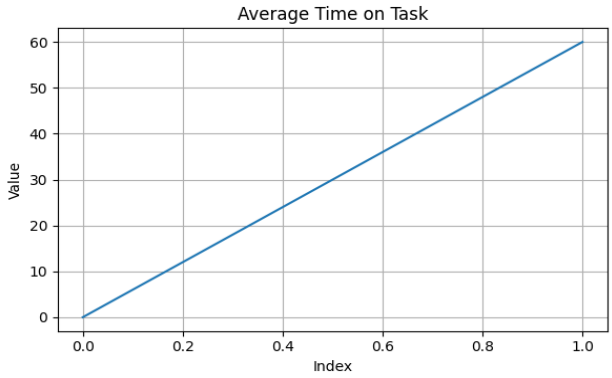


Fig. 11. Average Time on task

The comparative analytical results of the design and developed HCSD model are discussed in this phase with regard to the time required to develop software using this generated model. In this study, we use HCSD to analyse design time, implementation time, maintenance time, and code size. We also validate user acceptance in relation to time duration, and the results show that code impact has the greatest influence on human-centric process development.

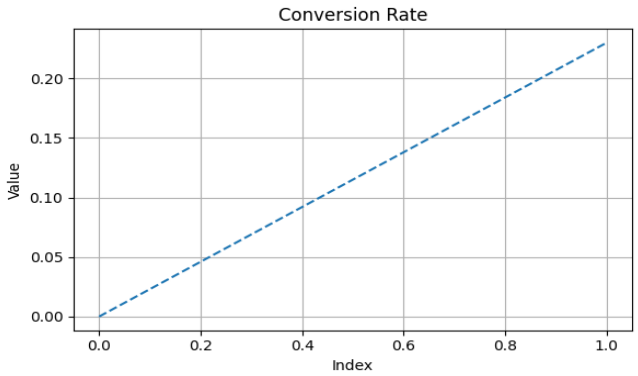


Fig. 12. Conversion rate

In terms of technological scenario and dependency, verification and validation elements are taken into account. Here, we also obtain the analytical result that the human-centric requirements satisfaction ratio is greater than 20%, and by using this model, the execution phase ratio is greater than 23%.

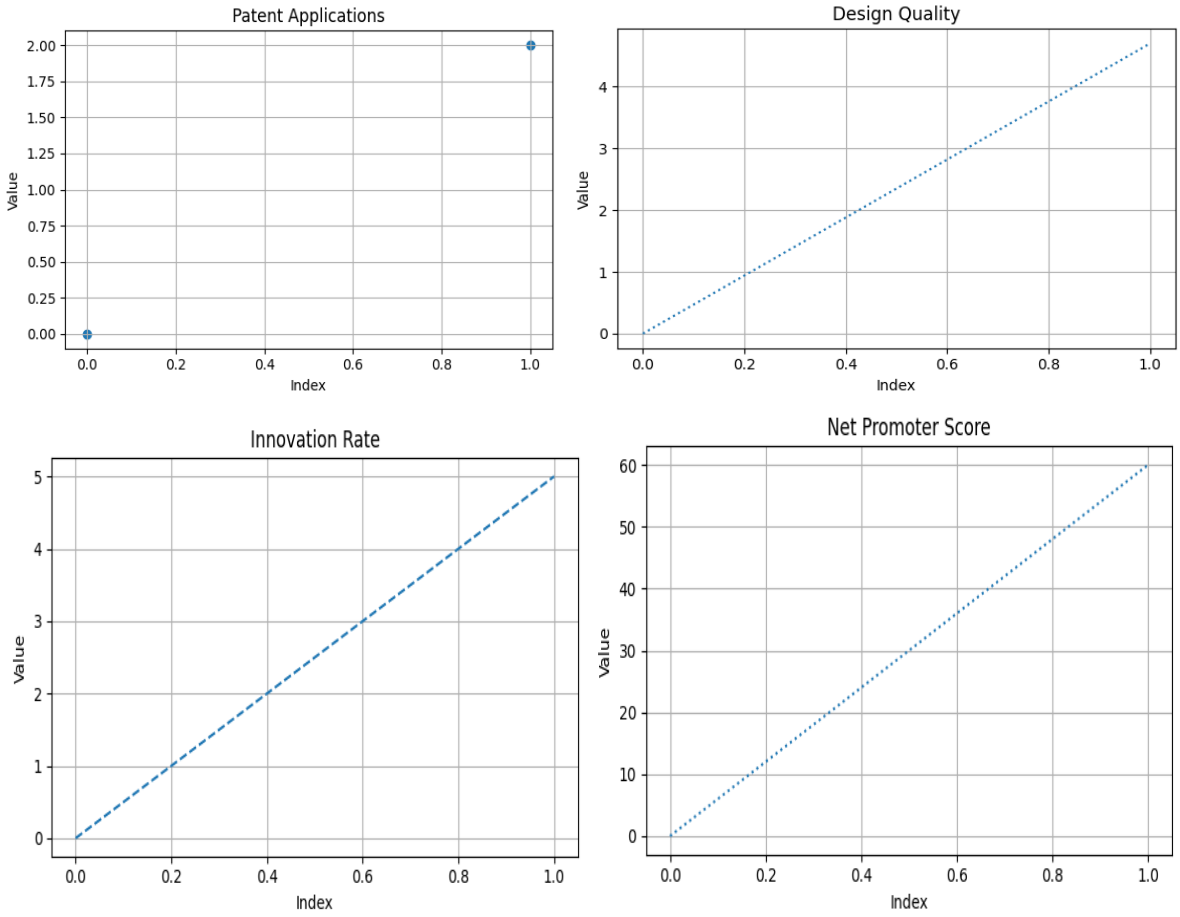


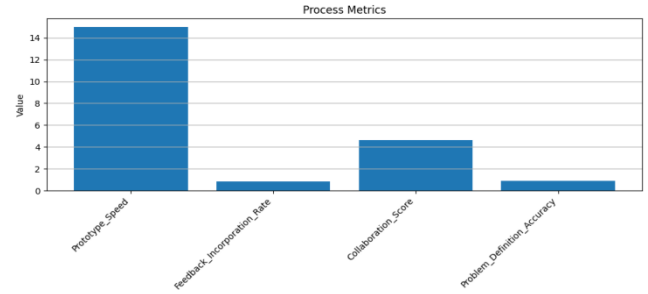
Fig. 13. Product Metrics

Here, the established software process management metamodel primarily focuses on end-user and business stakeholder feedback and requirement validation. Create a conceptual integrated human-centered software development (HCSD) model with a maximum quality index value of more than 37% of the overall reliability quality index for project validation and quality assurance.

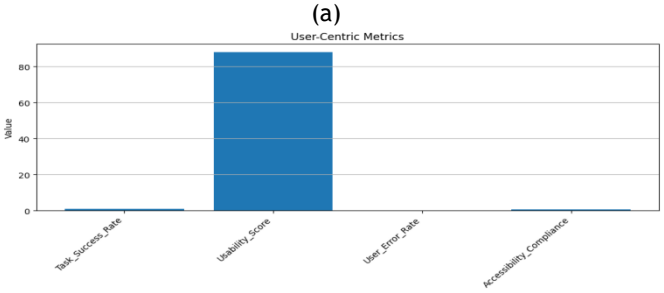
Table 2. Design Thinking for User-Centric Product Development

Category	Metric	Value
Empathize	User Interviews Conducted	40
	Empathy Score	8.7 / 10
	Persona Accuracy	92%
	Pain-Point Coverage	88%
Define	Problem Clarity Score	9.1 / 10
	Requirements Accuracy	93%
	Scope Stability Index	0.89
	Reframe Quality Score	8.4 / 10
Ideate	Number of Ideas Generated	120
	Idea Diversity Index	0.82
	Brainstorm Participation Rate	96%
	Concept Novelty Score	7.9 / 10
Prototype	Prototype Completion Time	3.2 days
	Iteration Count	6
	Prototype Fidelity Score	0.86
	Design Feasibility Score	91%
Test	Usability Score	8.8 / 10
	Task Success Rate	94%
	User Error Rate	3%
	User Satisfaction Score	4.6 / 5

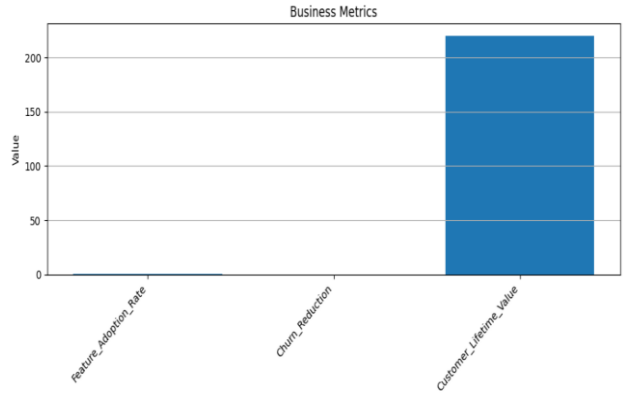
This metrics summary captures the full Design Thinking lifecycle, showing strong performance across all five stages—Empathize, Define, Ideate, Prototype, and Test. In the Empathize phase, extensive user research with 40 interviews and high empathy, persona accuracy, and pain-point coverage demonstrate a deep understanding of user needs.



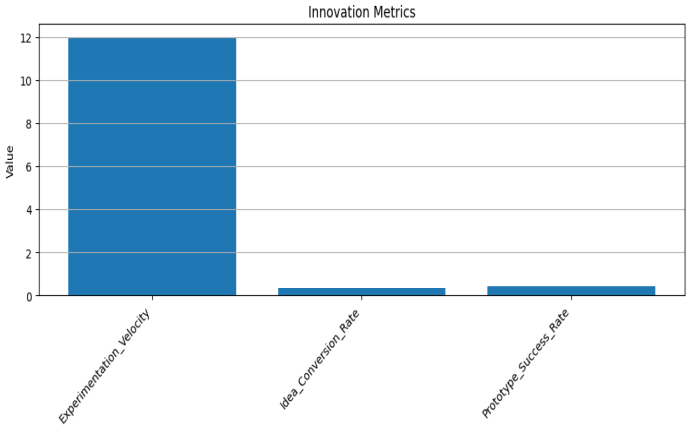
(a)



(b)



(c)



(d)

Fig. 14. Design Thinking for User-Centric Product Development

The Define stage reflects clear problem framing with strong requirement accuracy and a stable project scope. Ideation metrics highlight creativity and collaboration, with a large number of diverse ideas and strong participation. Prototype metrics show efficient and iterative development with high-fidelity, feasible designs produced quickly. Finally, Test-stage results indicate an excellent user experience, with high usability, task success, low error rates, and strong satisfaction. Together, these metrics show a well-executed, user-centered design process that balances insight, innovation, speed, and validation.

5. Conclusion

Using agile, human-centered, and DT methodologies, the research enhances the performance and efficacy of software development design and process management. Extreme Programming and Scrum technology are agile-human-centric software development approaches that focus on human-centered software development, process management, and other aspects of the DT approach used in earlier stages of software development to develop software products based on end-user satisfaction and viewpoints. In order to release the graphical representation of software processes and their development stages progress in the middle and running

stages of software productivity, we design and develop a conceptual integrated process management metamodel in this study using agile-human centric and DT approaches. Agile engineering methodologies improve software design efficiency and effectiveness by developing software in parallel across several stages and sprints with low resource usage and providing optimization solutions based on user requirements. The five steps of software design and development—empathize, define, ideate, prototype, and test—are used in the DT approach to gather user input and responds to predetermined designing phases and software development mindsets.

References

1. Krishan, Chitra, Amrish Kumar Choubey, Ayush Thakur, Astha Gupta, and Pranay Verma. "Enhancing Product Management Through User-Centric Design: A Human-Computer Interaction Perspective." In *2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pp. 428-433. IEEE, 2024.
2. Lemola, Martta. "Towards user-and customer-centricity: service design maturity and user experience capability in organizational transformation." (2024).
3. Rosário, Albérico Travassos. "Design Thinking in Product Design: Challenges and Opportunities." *Innovative Digital Practices and Globalization in Higher Education* (2023): 239-265.
4. Al Subhi, Nada Nasser, and David Bell. "The Changing Conception of Methodology: A Commentary on the Application of User Centric Design." *Journal of Mental Health Disorders* 5, no. 1 (2025): 98-101.
5. Härmälä, Sara. "Enhancing User-Centricity in Digital Service Development." (2021).
6. Onyeke, FIDELIS OTHUKE, WAGS NUMOIPIRI Digitemie, M. U. S. A. Adekunle, and IKIOMOWORIO NICHOLAS DIENAGHA Adewoyin. "Design thinking for SaaS product development in energy and technology: Aligning user-centric solutions with dynamic market demands." *Journal name missing* (2023).
7. Chapman, Andrew. "User-centric service design in concept development: case start-up company growth." (2017).
8. Martins, Flavio, Maria Fatima Almeida, Rodrigo Calili, and Agatha Oliveira. "Design thinking applied to smart home projects: a user-centric and sustainable perspective." *Sustainability* 12, no. 23 (2020): 10031.
9. Sinha, Akash Rakesh, and Anuradha Bansilal Choudhary. "Synergizing Teams for User-Centric Innovation: Cultivating Collaboration and Streamlining Development for Enhanced User Experience." *INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT Учредители: Indospace Publications* 1, no. 04 (2022): 1-6.
10. Lindfors, Leevi. "The role of user-centric design in global platform economy value creation." (2024).
11. Mukherjee, Biswajeet. *Product Intervention: User-centric Design to Fabrication*. CRC Press, 2023.
12. Subhi, Nada Al, Vaidehi Pandurengan, Adil Al Balushi, and David Bell. "Design Mindset Driven Educational Processes: A User-Centric Dynamic Model." In *Achieving Sustainable Business through AI, Technology Education and Computer Science: Volume 1: Computer Science, Business Sustainability, and Competitive Advantage*, pp. 299-310. Cham: Springer Nature Switzerland, 2024.
13. Lahiri, Aparna, Kathryn Cormican, and Suzana Sampaio. "Design thinking: From products to projects." *Procedia computer science* 181 (2021): 141-148.
14. Tuomi, Ilkka. "Beyond user-centric models of product creation." In *Everyday innovators: Researching the role of users in shaping ICT's*, pp. 21-38. Dordrecht: Springer Netherlands, 2005.
15. Abreu-Romu, Tatiana. "Using service design methods to improve discovery process in an e-learning business: accelerating innovation in user-centric product development-a case study." (2023).
16. Lopez, Diego, and Fatima Bhutto. "Human-centered design in product development: A paradigm shift for innovation." *Abbottabad University Journal of Business and Management Sciences* 1, no. 02 (2023): 94-104.
17. Luchs, Michael G., Scott Swan, and Abbie Griffin. *Design thinking: New product development essentials from the PDMA*. John Wiley & Sons, 2015.
18. Ammayath, Navya. "Design Thinking Approach to Develop a User-Centric Fitness Application." In *2024 3rd International Conference for Innovation in Technology (INOCON)*, pp. 1-8. IEEE, 2024.
19. Wöhrl, Andrea, Sophia Korte, Michael Bartl, and Volker Bilgram. "HOW TO LEVERAGE THE RIGHT USERS AT THE RIGHT TIME WITHIN USER-CENTRIC." *The PDMA Handbook of Innovation and New Product Development* (2023): 315.
20. Tolino, Umberto, and Ilaria Mariani. "A Pathway to Innovation. Design-Driven and User-Centric." In *Design Behind Interaction: Interfaces, Technology, Meanings*, pp. 39-54. Cham: Springer Nature Switzerland, 2024.
21. Suzic, Nikola, and Linda Zhang. "Customer-Centric UI/UX Design for Sustainable." In *Production Processes and Product Evolution in the Age of Disruption: Proceedings of the 9th Changeable, Agile, Reconfigurable and Virtual Production Conference (CARV2023) and the 11th World Mass Customization & Personalization Conference (MCPC2023), Bologna, Italy, June 2023*, p. 148. Springer Nature, 2023.
22. Perälä, Kalle. "The role of user-centric design in global platform economy value creation." (2024).
23. Sungkar, Risyad, and Dina Dellyana. "Transforming Enterprise Application Rationalizations: A Design Thinking Approach to Customer-Centric Innovation." (2025).
24. Margaria, Tiziana, and Bernhard Steffen. "Agile IT: thinking in user-centric models." In *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, pp. 490-502. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
25. Kumar, Rahul. "A Framework for Innovation Based on Design Thinking." *Building A Blueprint For Progress* (2023): 57.