JOIV : Int. J. Inform. Visualization, 8(3) - September 2024 1040-1048

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Enhancing Novice Developer Efficacy through UX Journey: Integrating User Experience and User Requirement to Develop Developer Skills

Wahyu Andhyka Kusuma [a,b,*], Azrul Hazri Jantan [a], Novia Admodisastro [a], Noris binti Mohd Norowi [a]

[a] *Human-Computer Interaction (HCI) Research Group, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia*

[b] *Informatics Department, Universitas Muhammadiyah Malang, Indonesia*

Corresponding author: [*]gs63875@student.upm.edu.my

***Abstract*—User experience and user requirements are two different approaches to software development. User requirements focus on meeting customer expectations and demands for software solutions, while user experience covers all aspects of software interaction with users. To increase the value of the software, the software must have usable and easy-to-use features with an attractive design or work environment that fits the user's behavior. Integrating software requirements and user experience can increase developer productivity by focusing on features that meet user requirements and expectations. This integration can also increase software development efficiency by addressing issues arising during development. This article addresses developers' challenges when addressing user needs and provides practical solutions widely accepted in industry and academia. Combining user experience and user needs into the UX Journey approach can increase developer productivity and confidence in software development. The design of the UX Journey is carried out by evaluating several existing design solution methods such as Design Thinking, IDEO, HPI, and Double Diamond to determine the existing conditions and needs for the problems faced. Then, by mapping the user, context, and domain, the model is obtained. appropriate. The proposed model comprises Discover, Explore, Test, and Listen activities. A trial was carried out on the respondents to test the method, and a feasibility test and an implementation schedule were obtained based on the statistical analysis of the initial user. It took 980-1500 minutes to complete the design solution. Focusing on features that align with user needs and improve problem-solving efficiency throughout development gives developers greater confidence in producing high-quality software.**

***Keywords*—User experience; user requirement; developer productivity; developer self-efficacy; solo software development; UX Journey.**

Manuscript received 20 May 2023; revised 20 Oct. 2023; accepted 3 Nov. 2023. Date of publication 30 Sep. 2024.

I. Introduction

Accurately selecting the appropriate software process is critical to delivering software products or services [1], [2]. Pragmatic thinking about methodology in software development sequences led to the popularity of software processes. A software process is a set of activities that aid in developing products or services [2]–[5]. Solo software development is a lightweight software development method (SSDM) that is designed to address the challenges associated with choosing the right software process, such as the Personal Software Process (PSP) [6]–[8], Personal Extreme Programming (PXP) [9], [10], Freelance as a Team (FaaT) [11], [12], Solo Scrum [13], [14], and Solo Software Development [15]–[17]. However, a significant challenge is the conventional method's ability to adapt to quick development iterations [2], [18], [19], software that meets more than just the users' needs [10], and smaller development teams [4], [9], [10], [15].

To ensure the success of the software, it is crucial to identify software requirements and features. Previous studies have highlighted four key features contributing to software success: complexity [20], [21], suitability [20], [22], changeability [2], [23], and transparency [2], [24]. Developers must have soft skills to comprehend these success factors and enhance the likelihood of their software products or services succeeding. One of the essential soft skills is the ability to grasp socio-technical skills [25]–[28], In a human-centered development perspective, developers are expected to be able to understand all aspects related to users [29].

Planning meticulously and regularly exploring requirements from the beginning of the development process is crucial to avoid issues with software requirements gathering. This involves identifying the needs of end-users and the business, defining user requirements, and ensuring all requirements are incorporated into the development process. Developing one's ability to interact socially requires an individual to trust his capacity to solve a problem with a specific goal. This ability is known as self-efficiency [30], [31]. When exploring software requirements, self-efficacy is crucial in assessing an individual's ability to perform a requirements assessment. Individuals with high self-efficacy in identifying software requirements will be more confident in gathering and collating end-user and business needs, ensuring that all requirements are incorporated into the development process. By enhancing their self-efficacy in identifying software requirements, they can improve their ability to effectively identify and develop software that meets end-user needs and underlying business objectives. Furthermore, in small or solo software development, an individual's capacity is crucial for maximizing resource utilization and achieving efficiency and effectiveness in the development process.

The high demand for software that has practical value requires developers to utilize technical abilities to interact socially to understand system needs from the perspective of users; this is the main objective of this research. An academic and industry practitioner-developed framework called UX Journey has been proposed to address this. This framework consolidates ongoing solutions from previous studies and integrates user experience and requirements to understand the human value in user requirements. This, in turn, increases their competence for solo and small-team development.

II. Materials and Method

The previous research reflects user requirements as essential in developing a system and the research carried out. In the context explained previously, understanding user needs from the user's perspective can be achieved by integrating user experience into the elicitation process. This will be useful for increasing usability at the organizational level and for users and society in general. It is a crucial aspect of the development process. This study explores how individual developers use socio-technical skills to understand user needs from a consumer's perspective.

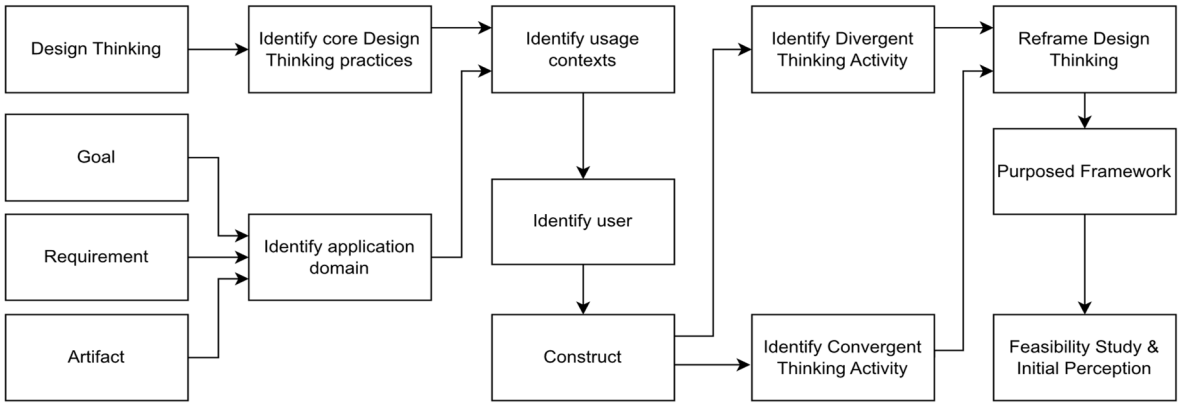


Fig. 1 Research Process

A. Research Process

User requirements are crucial and essential for successful user collaboration in software development. However, understanding what users need from a consumer perspective is a socio-technical skill important for increasing business value and competitiveness in the market. Collaborating with users also reduces the risk of producing low-quality software products. The focus of this research, illustrated in Figure 1, is to address the challenge of implementing self-efficacy in socio-technical skills, particularly for solo and small software development. The goal is to provide a widely applicable framework for training, academia, and industry to enhance individual productivity in software development. A mind map is used as the foundation of the research process to ensure a solid and precise approach, with every step described in the following section.

B. Core Design Thinking Practices

Over the past two decades, design thinking has become increasingly popular in various fields and is seen as a powerful approach to tackling complex, interdisciplinary problems [32]. Design thinking is a broad method for addressing socially ambiguous design problems [32]. Early definitions of design thinking describe it as the study of the cognitive processes involved in design, which are inherent in human cognition. Dunne and Martin suggest that design thinking is a way of thinking and applying mental processes to design products, services, or systems, resulting in elegant and user-friendly outcomes [33].

Design thinking is not limited to software development and encompasses various disciplines, as it involves conceptualizing processes, creating artifacts, planning, and intention. According to Brown's research, design thinking is a user-centered approach to innovation in design that takes a sensitive approach to the user [33]. Design thinking is recognized as a designer's method for matching user needs with what is technologically feasible and what can be changed by a business strategy to add value to customers and capture market opportunities. Brown's research provides a broader perspective on design thinking, which is used to view the design from multiple perspectives, such as an approach for creating new viable solutions and improving solutions to meet customer needs with added value. Brown's research also

suggests that design thinking is an integral way of thinking about design in the planning and design process [33]. Figure 2 illustrates Brown's approach to Design Thinking.

To understand how developers can incorporate design thinking into their work, it is essential to break down the concept of design and examine how its different aspects are interconnected. Design is a unified process that involves creating specifications for a design object. To approach design critically, several factors must be considered. The first is the environment in which the design object will be used, which will determine its unique characteristics and needs. The second is the goal or purpose of the design object, which should address user problems. The third aspect involves the desired properties or requirements of the design object based on user expectations. The fourth aspect involves collecting component types or primitives used in the design. Finally, the fifth aspect involves any constraints that may limit the possible solutions for the design.

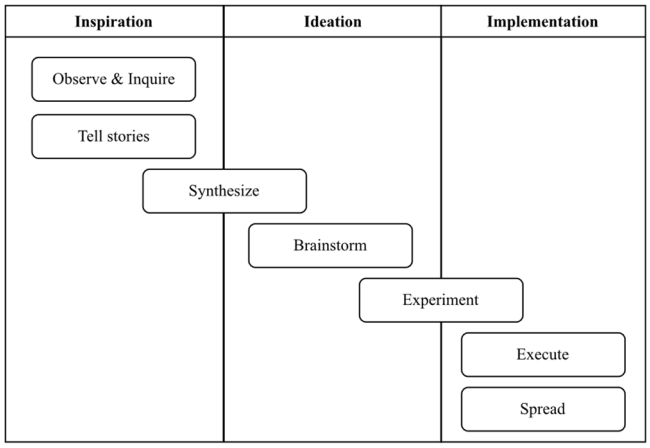


Fig. 2 Design Thinking is expressed by Brown [34].

C. Application Domain

Design objects can manifest in various forms, such as artifacts, products, systems, services, and even software products encompassing lines of code, database queries, and algorithms. The design concept is depicted in Figure 3, where the design form serves as an artifact within a specific context. The figure illustrates a domain where user requirements interact and influence each other within the design concept.

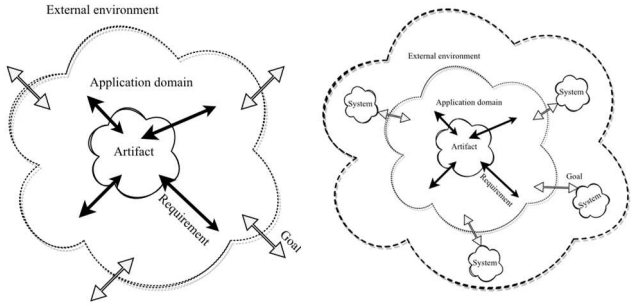


Fig. 3 Design concept related to artifact, domain, and external environment. (b) Design and Correlation with Other Systems [35]

To create a successful design, developers must adopt a holistic approach, considering the interdependence between systems. This includes business processes, work environments, and software, as depicted in Figure 3. The design must also meet the requirements of the application domain, which are influenced by the external environment. Understanding user needs and expectations is crucial, and developers should strive to understand each design aspect's unique characteristics. Design thinking addresses this challenge by emphasizing empathy, integrative thinking, experimentalism, optimism, and collaboration. Empathy requires developers to consider the user's context holistically from the perspective of multiple users. Integrative thinking involves presenting creative solutions from all aspects of the application domain and external environment. Experimentalism involves exploring novel solutions to determine the best potential solution. Optimism consists in maintaining the chosen solution as the best one. Collaboration consists of working with interdisciplinary stakeholders to find innovative solutions.

To create a design for a project, it's essential to conduct a thorough examination that generates a model addressing user issues. Design thinking employs a model or framework consisting of various techniques to tackle design problems effectively. Eris introduced the Divergent-Convergent Inquiry-based Design Thinking Model (DCIDT) as an approach to analyzing design problems comprehensively [36]. The model, depicted in Figure 4, outlines design thinking as two cognitive approaches associated with fundamental modalities: divergent and convergent thinking.

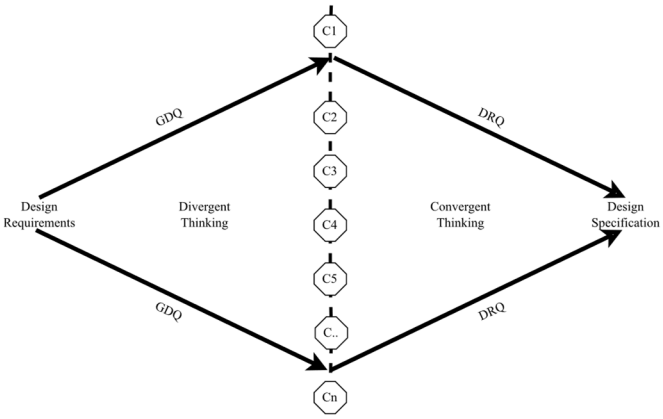


Fig. 4 Divergent-Convergent Inquiry-based Design Thinking Model [36]

Transforming user requirements into design specifications in the DCIDT model involves divergent and convergent thinking. This is achieved through a series of questions that start with Generative Design Questions (GDQ), which generate a series of design concepts from the design requirements. The GDQ also helps create, synthesize, and extend potential design concepts. The obtained design concepts are then analyzed, evaluated, and validated through a series of Deep Reasoning Questions (DRQ). The DRQ helps convert the design concepts into design potential and specifications, making it feasible according to the user's design needs and expectations.

III. Results and Discussion

This section outlines the design and architecture of the UX Journey, which consists of five key components elaborated in the following sub-sections: Usage Contexts, User and Construct Functional Description, UX Models Provided by

the UX Journey, Architecture, and Technical Feasibility. This section confirms that the UX Journey is built on a solid foundation and fundamentals, which enhances trust in its reliability and potential for implementation in various domains.

A. UX Journey: Usage Context

The intersection of user experience and requirements in software development results in software that is easy to use, meets user needs and expectations, and has an appealing design. Integrating these two approaches enhances the usability of software. Moreover, it increases developer productivity by prioritizing developing features that fulfill user needs, eliminating unnecessary ones, and proactively addressing potential issues. This integration enhances software development efficiency, saving developers time and effort. In UX Journey, integrating user experience and user requirements also enhances developer productivity and self-efficacy in software development. Focusing on user needs and improving problem-solving efficiency boosts developer confidence in creating high-quality software.

B. UX Journey: User

UX Journey is a methodology that integrates user experience and user requirements to explore and address user needs and solutions. It is designed for students, academics, researchers, and industry professionals aiming to enhance their skills in analyzing user needs for software requirements. This iterative approach includes various UX activities to identify user problems and craft corresponding solutions. UX Journey is structured to be manageable for individual developers or small teams operating within practical time frames.

For students, UX Journey provides a structured learning path to understand how user experience influences the exploration of user needs and the quality of software requirements in terms of usability, maintainability, and other development attributes. Academics can use UX Journey as an educational tool to teach students and bridge the gap between academic learning and industry practices. Researchers can leverage the methodology for practical or theoretical investigations into user requirements, mainly focusing on the quality attribute of user experience. Meanwhile, industry professionals can employ UX Journey for product research and development, even with limited resources, ensuring that delivered quality aligns closely with user expectations.

C. UX Journey: Construct

UX Journey is an adaptation of several design thinking approaches that have been reliably proven in previous studies. To accomplish the design requirements and meet users' expectations, developers must consider the problem holistically. Design thinking is iterative and non-linear and can be categorized into four main activities: empathy, problem framing, ideas and visualization, and testing and iteration. In the empathy phase, developers focus on understanding the problems users face through primary and secondary exploration. The problem-framing activity involves reviewing the solutions generated in the empathy phase and classifying them based on proximity and potential solutions. Idea and visualization activities include collaborating with users to bring inspiration and create low or high-fidelity displays. Testing and iteration activities include testing potential solutions for usability, evaluating and improving the solutions iteratively, and delivering the final product to the development team.

D. UX Journey proposed a model

Understanding one's abilities is the most essential part of this research. The effectiveness of the UX Journey depends on how confident a user is in understanding and capturing the market potential for an innovative product. One widely recognized model, developed by Brown, consists of three stages: inspiration, ideation, and implementation, each supported by relevant sub-activities. Central to design thinking emphasizes understanding the user's emotions, fostering empathy as a critical strategy for solving design challenges. Successful design solutions are rooted in deep insights, careful observation, and empathy toward user experiences and needs.

Creating effective design solutions involves thoroughly exploring the context, generating ideas, evaluating solutions, and implementing them in line with the project's specific goals. The framing process is essential for identifying or refining solutions by adjusting existing frames, shifting perspectives, or drawing connections to other contexts. The UX Journey (ilustrated in Figure 5) model includes four key activities: discover, explore, test, and listen. These activities guide developers in creating and refining design solutions based on user feedback. Testing, in particular, is critical to ensuring that solutions effectively meet user needs and expectations.

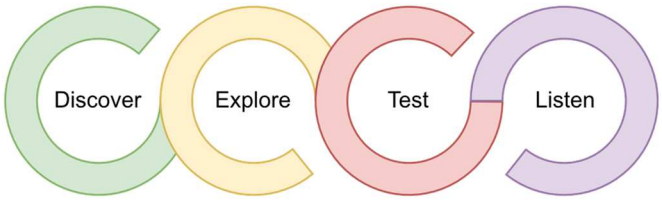


Fig. 5 Purposed UX model with UX Journey

Figure 6 illustrates the functional description of UX Journey, developed with foundational solid principles to address the challenges individuals and small development teams face in enhancing their socio-technical skills in both academic and industrial settings. UX Journey is a comprehensive tool that combines psychomotor and cognitive elements to strengthen the self-efficacy of individual developers. It employs a design thinking model and comprises four key structures, namely discovering, exploring, testing, and listening, to guide developers in integrating user experience and requirements to meet user expectations.

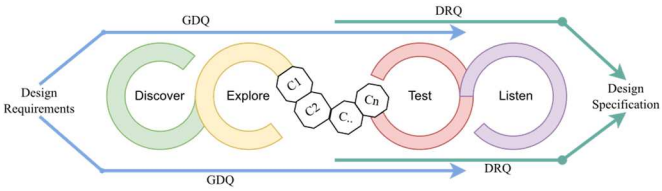


Fig. 6 UX Journey functional description

The activities of UX Journey commence with identifying the design requirements through a series of exploration

exercises that involve researching existing products and obtaining user or organizational feedback. The developer can then initiate testing to identify solutions that meet their specific needs before organizing listening activities to evaluate the potential solutions and determine their suitability for the design specifications. Figure 6 displays the functional model of the UX Journey, which is then transformed into a detailed technical architecture represented in Figure 7. There are four main components of the UX Journey: discovering, exploring, testing, and listening. These primary activities comprise sub-activities that implement the user experience method, offering a coherent set of quality elicitation methods. Figure 7 illustrates the detailed sub-activities of the UX Journey technical architecture, including:

1) Discover: This activity encompasses three sub-activities that intersect with the Explore activity. SWOT Analysis is used to identify project feasibility, Competitor Analysis gathers information about competitors in the market, and Hypothesis establishes predefined scope and goals for the project.

2) Explore: This main activity involves several sub-activities, such as identifying behavioral variables, preparing and selecting questions, using index cards, conducting map interviews, documenting findings, identifying significant behavior patterns, expanding descriptions and variables, synthesizing characteristics and relevant goals, checking for redundancy and completeness, creating wireframes, sitemaps, user scenarios, personas, customer journeys, and prototypes.

3) Test: This activity ensures that the design solution meets the needs and expectations of users.

4) Listen: Although placed outside the design solution process, this activity is essential for providing an overview of the market's response after the product's release. Obtaining user feedback is necessary to develop the product into the next version.

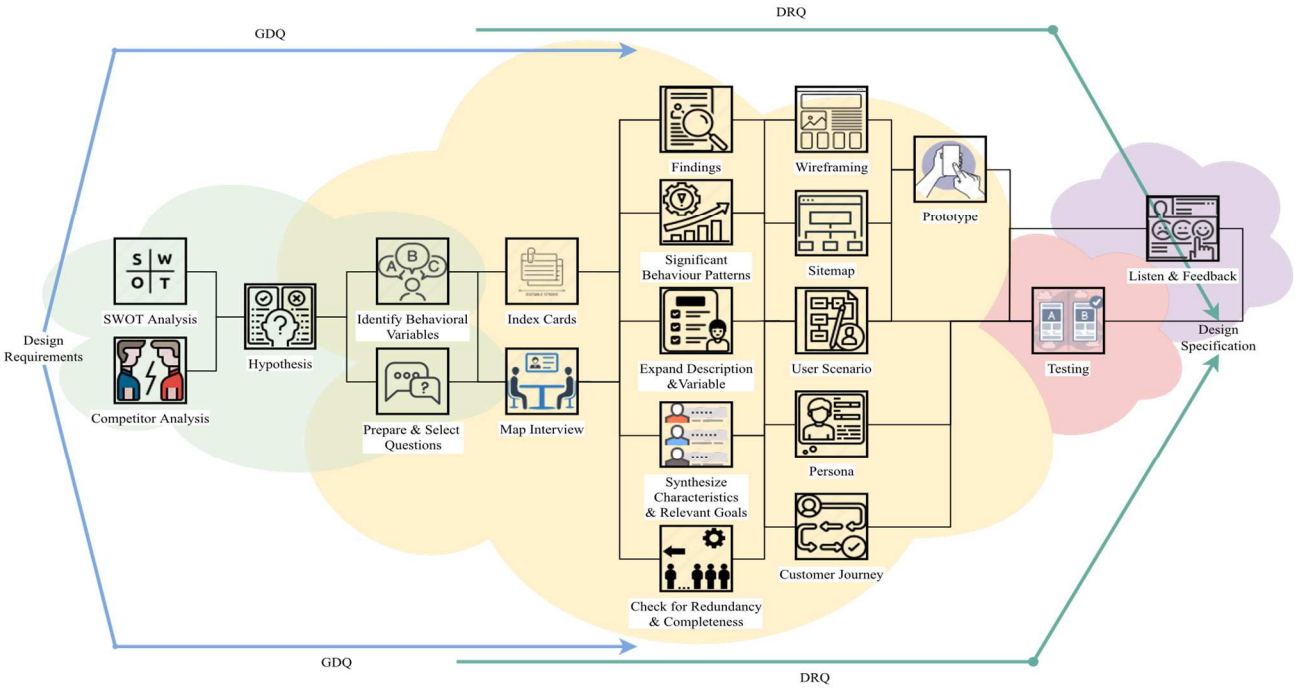


Fig. 7 UX Journey Architecture

E. Testing and Initial Perceptions

The characteristics of the UX Journey architecture have been reviewed and indicate that it is reliable, comprehensive, and effective in enhancing developer self-efficacy. The architecture allows developers to explore user needs and improve their socio-technical abilities to understand user expectations. The next phase of the research involves conducting a feasibility study, which examines the strengths and weaknesses of an existing or new process or method objectively and rationally, as well as the potential benefits and challenges of its implementation. This study will help determine whether the process or model is feasible.

A feasibility study must identify and evaluate alternatives to propose a suitable method or model. This process helps in finding solutions that are practical, feasible, and meet the relevant legal requirements. Evaluating the alternatives requires considering various factors such as risk level, costs, and benefits based on different feasibility areas. While there is yet to be a consensus on the specific domains that feasibility studies must cover, there is some agreement on the five general areas known as TELOS (technical, economic, legal, operational, and schedule). In the case of the UX Journey, a feasibility study analysis will be conducted using the TELOS approach to assess its viability.

F. Technical Feasibility

As shown in Table 1, the technical feasibility assessment evaluates the potential of utilizing existing or integrating new technology. The outcome addresses several queries, such as whether the technology produces satisfactory outcomes, whether it requires additional time and resources, and whether it enhances overall performance.

TABLE I
THE TECHNICAL ASPECT OF THE UX JOURNEY

Technical Aspect	Selected existing model			Purposed
	IDEO	HPI	Double Diamond	UX Journey
User Focus	Observing and understanding the challenge and user context	Understanding existing information, collecting insight about user needs	Searching for new opportunities, information, trends, and insight	Observing and understanding empathy, new opportunities, existing information, and insight.
Communication (Phase)	Ideation: sharing and making sense of collected data, feedback	Prototype: presenting the idea to potential user	Develop: using creative tools like brainstorming	Discover, explore, test, and listen to collaborative
Product improvement (Phase)	Implementation: refining business models	Test: iterative cycles, collective feedback every time	Deliver: final concept and launching	Listen: Launch the market analysis product and review user feedback.

The proposed UX Journey model focuses on improving the user experience and product creation and prioritizes enhancing individual skills. This differs from other models solely concentrating on the user or product creation. In the UX Journey, there is a strong emphasis on improving developers' confidence and socio-technical skills through collaboration with users while addressing their needs and expectations.

G. Economic Feasibility

The economic feasibility evaluation aims to assess whether the proposed new technology has a financial impact compared to the existing technology without any bias or prior assumptions. Two approaches can be used for this evaluation: measurable effects, which involve quantifiable indicators such as cost reduction, output improvement, or service improvement, and effects that are impossible to measure, such as risks or problems that may arise during the implementation process.

The proposed UX Journey model has an advantage regarding personnel requirements as it was designed to be executed by individuals or solo. This was done to enhance individual abilities to interact with users. Additionally, the UX Journey model can be completed within a reasonable timeframe of 16.3-25 hours from the beginning of the process to testing the design solution (based on schedule feasibility).

H. Legislative Feasibility

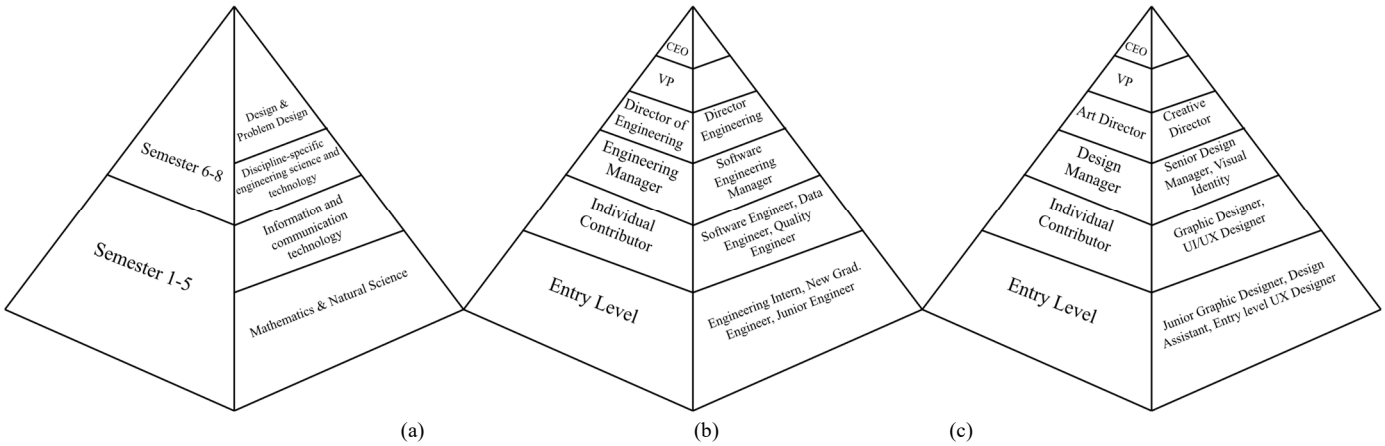
This aspect defines the legal requirements for introducing or implementing new technology. Evaluating the legal feasibility establishes the basis for conducting a quality assessment of the technology. Table 2 displays the legal basis used in this research.

TABLE II
SEVERAL STANDARDS IN LEGISLATIVE FEASIBILITY

Code	Standard
ISO/IEC 27000-7	Information security management systems
ISO/IEC TR 27008	Guidelines for auditors on information security controls
ISO/IEC TR 27015	Information security management guidelines
ISO/IEC TR 27016	Information security management
ISO 9241-11:2018	Ergonomics of human-system interaction
ISO 25010	Usability

I. Operational Feasibility

Operational feasibility aims to assess whether new technology can be effectively implemented within an organization compared to the current state. The feasibility study aims to determine whether the proposed UX Journey model can be implemented within an organization. The UX Journey model is primarily designed for individuals, with a hierarchical structure consisting of users and developers, As shown in Figure 8. However, the model could be used in academia, small teams, and individual training, requiring different organizational structures. Therefore, the designer's organizational structure is used to assess the feasibility of this model.



(a) (b) (c)

Fig. 8 Operational feasibility (a) academic structure (b) software engineer structure (c) designer structure [37]

J. Schedule Feasibility and Initial Perceptions

The feasibility schedule provides information on the requirements for implementing the technology, allowing for an evaluation of its overall performance and an understanding of how it can be adapted to its current state.

TABLE III
RESPONDENT PROFILE

Respondent	Industry	Location	Experience (Years)
Respondent 1	Property	Singapura	10
Respondent 2	Telecommunication	Indonesia	10
Respondent 3	Cloud computing	Singapura	10
Respondent 4	Cryptocurrency	Indonesia	16
Respondent 5	Software house	Singapura	12
Respondent 6	Software house	Indonesia	13
Respondent 7	Startup Education	Indonesia	13
Respondent 8-111	Startup and Freelance	Indonesia	1-2

Each respondent was asked to develop a solution design from the topics of Big Data, Disability, and Education; all data is publicly available [38] to develop a design solution for each respondent using the UX Journey Worksheet [39]. The results of the initial perception are then displayed in Table 4 as a statistical reference [38]. The initial stage is to inquire about the respondents' willingness to participate in the research. The second stage involves obtaining detailed demographic data and verifying the competence of the respondents. Verification is done through several criteria, including whether the respondent's name is listed on the company website, providing valid evidence that the respondent is working and competent, and verifying educational data by examining the respondent's scientific work history. In this research, a survey was used to gauge the initial perceptions of the UX Journey model and industry perspectives when using it. The third stage involved presenting respondents with two architectures: the design thinking model by Brown and the UX Journey model. Respondents were given seven days to evaluate both models and indicate which one they believed was more effective in combining user research activities with user experience and requirement activities. The fourth stage involved assessing whether the model could be used to increase individual competency or by a single developer for product research.

TABLE IV
UX JOURNEY SCHEDULE FEASIBILITY

Process	Time (minutes)		Activity
	Min	Max	
Empathy & define	30	60	SWOT Analysis
	10	30	Prepare Questions
	10	30	Selected Questions
	30	60	Competitor Analysis
	20	30	Hypotheses
	20	30	Identify Behavioral Variables
	60	60	Persona
Ideate	20	30	Findings
	60	120	Index card/ Sticky notes
	30	30	Map Interview
	30	30	Significant Behavior Patterns
	30	30	Synthesize Characteristics and Relevant Goals
	30	40	Check for Redundancy and Completeness
	30	30	Expand Description and Variable
	60	60	Customer Journey
	120	240	Wireframing (Low Fidelity)
Prototype	60	120	User Scenario
	30	30	Sitemap
	240	320	Mockup (High Fidelity)
Test	60	120	Testing
Total	**980**	**1500**	

K. Discussion

User requirements are a crucial part of software development, and they include the needs and expectations of software users. To ensure successful product delivery, developers should focus on software quality and address potential problems during development. It is essential to gather user requirements at the start of the project as they determine the project's success. The user experience should consist of four main activities: empathy, problem definition, idea and visualization, and testing and iteration. Integrating user experience and user requirements into the software development process can enhance production productivity and effectiveness. This study uses the design thinking approach to combine these two processes. Design thinking is an approach to solving socially ambiguous design problems, which helps developers think about design and create elegant and usable products. This integration improves efficiency in

software development by identifying and addressing potential problems during the development process. Several aspects need to be considered to understand how developers can think about design, including the environment in which the design object will exist, the goal ascribed to the object, user requirements and expectations, component types, and constraints.

IV. Conclusion

In this paper, the authors propose a framework called UX Journey that integrates user experience (UX) and user requirements (UR) to complement modern software development life cycle (SDLC) practices. The framework aims to enhance individual skills in socio-technical areas and enable developers to understand better user needs from the consumer's perspective. Developers can use UX Journey to improve productivity, self-efficacy, and confidence in developing quality software. The framework can be adopted by students, academics, researchers, and industry professionals to enhance their skills in analyzing user needs for software requirements and use cases. The UX Journey model adapts several design thinking approaches and includes several typical UX methods to understand user experience and emotions. The authors conducted a feasibility study using the TELOS approach and found that UX Journey has excellent potential as a self-efficacy method to capture user needs. The study also found the framework feasible at several levels, including academia, training, and industries. However, the authors recommend balancing divergent and convergent thinking activities for a specific purpose.

Acknowledgment

We acknowledge the support and funding provided by the Ministry of Higher Education (Kementerian Pengajian Tinggi) and the Research Management Centre (RMC), Universiti Putra Malaysia, through its Fundamental Research Grant Scheme (FRGS) – Project Code 08-01-20-2319FR - 5540451. We also thank the anonymous reviewers for their valuable feedback and comments on this article.

References

[1] I. Sommerville, *Software engineering*, 10. ed., Global ed. in Always learning. Boston Munich: Pearson, 2016.

[2] R. S. Pressman and B. R. Maxim, *Software engineering: a practitioner's approach*, Ninth edition. New York, NY: McGraw-Hill Education, 2020.

[3] R. Anwar, M. Rehman, K. S. Wang, M. A. Hashmani, and A. Shamim, "Investigation of Knowledge Sharing Behavior in Global Software Development Organizations Using Social Cognitive Theory," IEEE Access, vol. 7, pp. 71286–71298, 2019, doi:10.1109/access.2019.2912657.

[4] F. Almeida, J. Simões, and S. Lopes, "Exploring the Benefits of Combining DevOps and Agile," Future Internet, vol. 14, no. 2, p. 63, Feb. 2022, doi: 10.3390/fi14020063.

[5] W. A. Kusuma, A. H. B. Jantan, R. B. Abdullah, N. I. Admodisastro, and N. B. M. Norowi, "Integrating Good UX Development Practices in Solo Agile," presented at the 2022 8th International HCI and UX Conference in Indonesia (CHIuXiD), IEEE, 2022, pp. 47–52.

[6] L. B. Angarita, "Application of the Personal Software Process in Software Engineering Courses," *RISTI - Rev. Iber. Sist. E Tecnol. Inf.*, vol. 2021, no. E43, pp. 501–516, 2021.

[7] Z. Wang, "The Compare of Solo Programming and Pair Programming Strategies in a Scrum Team: A Multi-agent Simulation," Intelligent Algorithms in Software Engineering, pp. 122–147, 2020, doi:10.1007/978-3-030-51965-0_11.

[8] G. A. Arduino and V. A. Bollati, "Proceso Personal de Software aplicado a la formación universitaria: una revisión sistemática de literatura [Not available in English]," 2020 IEEE Congreso Bienal de Argentina (ARGENCON), Dec. 2020, doi:10.1109/argencon49523.2020.9505471.

[9] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming – An Agile Process for Autonomous Developers," p. 8, 2009.

[10] G. E. Iyawa, "Personal Extreme Programming: Exploring Developers' Adoption," in *Americas Conference on Information Systems*, 2020, p. 11.

[11] Enhancing Multiculturality, Oct. 2015, doi:10.1145/2808580.2808685.

[12] V. Gupta, J. M. Fernandez-Crehuet, and T. Hanne, "Freelancers in the Software Development Process: A Systematic Mapping Study," Processes, vol. 8, no. 10, p. 1215, Sep. 2020, doi: 10.3390/pr8101215.

[13] T. Pagotto, J. A. Fabri, A. Lerario, and J. A. Gonçalves, "Scrum solo: Software process for individual development," 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Jun. 2016, doi: 10.1109/cisti.2016.7521555.

[14] Z. Wang, "Modelling and Simulation of Scrum Team Strategies: A Multi-agent Approach," Software Engineering Perspectives in Intelligent Systems, pp. 32–63, 2020, doi: 10.1007/978-3-030-63322-6_4.

[15] S. Moyo and E. Mnkandla, "A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices," IEEE Access, vol. 8, pp. 33735–33747, 2020, doi:10.1109/access.2020.2971000.

[16] S. Moyo and E. Mnkandla, "A Metasynthesis of Solo Software Development Methodologies," 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Nov. 2019, doi: 10.1109/imitec45504.2019.9015867.

[17] W. A. Kusuma, A. H. Jantan, N. I. Admodisastro, and N. M. Norowi, "Reframed Design Thinking and Feasibility Analysis of UX Journey: Integrating User Experience and User Requirement for Solo Software Development," Jan. 2023, doi: 10.20944/preprints202301.0190.v1.W. A. Kusuma, A. H. Jantan, N. I. Admodisastro, and N. M. Norowi, "Reframed Design Thinking and Feasibility Analysis of UX Journey: Integrating User Experience and User Requirement for Solo Software Development," 2023, doi: 10.20944/preprints202301.0190.v1.

[18] R. Chanin, L. Pompermaier, A. Sales, and R. Prikladnicki, "Collaborative Practices for Software Requirements Gathering in Software Startups," 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), May 2019, doi: 10.1109/chase.2019.00014.

[19] S. Gabriel, N. Niewoehner, L. Asmar, A. Kühn, and R. Dumitrescu, "Integration of agile practices in the product development process of intelligent technical systems," Procedia CIRP, vol. 100, pp. 427–432, 2021, doi: 10.1016/j.procir.2021.05.099.

[20] N. Nelson, C. Brindescu, S. McKee, A. Sarma, and D. Dig, "The life-cycle of merge conflicts: processes, barriers, and strategies," Empirical Software Engineering, vol. 24, no. 5, pp. 2863–2906, Feb. 2019, doi:10.1007/s10664-018-9674-x.

[21] J. Tian, J. Yin, and L. Xiao, "Software Requirements Engineer's Ability Assessment Method Based on Empirical Software Engineering," Wireless Communications and Mobile Computing, vol. 2022, pp. 1–10, Mar. 2022, doi: 10.1155/2022/3617140.

[22] F. G. Faust et al., "Mixed prototypes for the evaluation of usability and user experience: simulating an interactive electronic device," Virtual Reality, vol. 23, no. 2, pp. 197–211, Jul. 2018, doi: 10.1007/s10055-018-0356-1.

[23] M. Zielske, T. Held, and A. Kourouklis, "A Framework on the Use of Agile Methods in Logistics Startups," Logistics, vol. 6, no. 1, p. 19, Feb. 2022, doi: 10.3390/logistics6010019.

[24] S. Scheuer, P. Ferner, Y. Prinzellner, and G. Aumayr, "Collection of End User Requirements and Use Cases during a Pandemic—Towards a Framework for Applied Research Projects," Information, vol. 13, no. 5, p. 255, May 2022, doi: 10.3390/info13050255.

[25] N. Ahmadi Eftekhari, S. Mani, J. Bakhshi, and S. Mani, "Project Manager Competencies for Dealing with Socio-Technical Complexity: A Grounded Theory Construction," Systems, vol. 10, no. 5, p. 161, Sep. 2022, doi: 10.3390/systems10050161.

[26] J. W. Miska *et al.*, "How do undergraduate engineering students conceptualize product design? An analysis of two third-year design courses," *J. Eng. Educ.*, vol. 111, no. 3, pp. 616–641, Jul. 2022, doi:10.1002/jee.20468.

[27] A. Hawlitschek, S. Berndt, and S. Schulz, “Empirical research on pair programming in higher education: a literature review,” Computer Science Education, vol. 33, no. 3, pp. 400–428, Mar. 2022, doi: 10.1080/08993408.2022.2039504.A. Hawlitschek, S. Berndt, and S. Schulz, “Empirical research on pair programming in higher education: a literature review,” *Comput. Sci. Educ.*, 2022, doi:10.1080/08993408.2022.2039504.

[28] S. D. Vishnubhotla, E. Mendes, and L. Lundberg, “Understanding the perceived relevance of capability measures: A survey of Agile Software Development practitioners,” Journal of Systems and Software, vol. 180, p. 111013, Oct. 2021, doi:10.1016/j.jss.2021.111013.

[29] W. A. Kusuma et al., “Bibliometrics Analysis and Research Profiling to Solve User Experience Overload Information,” Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, Nov. 2022, doi: 10.22219/kinetik.v7i4.1575.

[30] T.-C. Hsu, H. Abelson, E. Patton, S.-C. Chen, and H.-N. Chang, “Self-efficacy and behavior patterns of learners using a real-time collaboration system developed for group programming,” International Journal of Computer-Supported Collaborative Learning, vol. 16, no. 4, pp. 559–582, Dec. 2021, doi: 10.1007/s11412-021-09357-3.

[31] A. Issaee, R. Motschnig, and O. Comber, “Pair- versus solo-programming of mini-games as a setting for learning to program: An Action Research approach,” 2021 IEEE Frontiers in Education Conference (FIE), Oct. 2021, doi: 10.1109/fie49875.2021.9637178.

[32] E. D. Canedo, A. T. S. Calazans, G. R. S. Silva, P. H. T. Costa, R. P. de Mesquita, and E. T. S. Masson, “Creativity and Design Thinking as Facilitators in Requirements Elicitation,” International Journal of Software Engineering and Knowledge Engineering, vol. 32, no. 10, pp. 1527–1558, Oct. 2022, doi: 10.1142/s0218194022500607.

[33] M. Sanchez-Gordon, R. Mendoza-Gonzalez, and R. Colomo-Palacios, “Design Thinking in Practice,” IT Professional, vol. 23, no. 4, pp. 95–100, Jul. 2021, doi: 10.1109/mitp.2020.2993113.

[34] A. W. Brown, “Personal software engineering project management process,” Proceedings of the 21st international conference on Software engineering, May 1999, doi: 10.1145/302405.302928.

[35] S. Adikari, C. McDonald, and J. Campbell, “Reframed Contexts: Design Thinking for Agile User Experience Design,” Design, User Experience, and Usability. Design Philosophy, Methods, and Tools, pp. 3–12, 2013, doi: 10.1007/978-3-642-39229-0_1.

[36] F. Pinkow, “Creative cognition: A multidisciplinary and integrative framework of creative thinking,” Creativity and Innovation Management, vol. 32, no. 3, pp. 472–492, Dec. 2022, doi:10.1111/caim.12541.

[37] CC2020 Task Force, “Computing Curricula 2020,” Nov. 2020, doi:10.1145/3467967.

[38] Wahyu Andhyka Kusuma, Azrul Hazri bin Jantan, Novia Admodisastro, and Noris Norowi, “UX Journey Initial Perception.” 2023. doi: https://doi.org/10.17632/br83vbh2rp.1.

[39] W. A. Kusuma, “UX Journey - The process to optimize your resource for the quality User Requirement.” 2023. Accessed: Mar. 30, 2023. [Online]. Available: https://github.com/uxjourney/requirement