Article

Reframed Design Thinking and Feasibility Analysis of UX Journey: Integrating User Experience and User Requirement for Solo Software Development

Wahyu Andhyka Kusuma * , Azrul Hazri Jantan * , Novia Indriaty Admodisastro , Noris Mohd Norowi

Posted Date: 11 January 2023

doi: 10.20944/preprints202301.0190.v1

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

Article

Reframed Design Thinking and Feasibility Analysis of UX Journey: Integrating User Experience and User Requirement for Solo Software Development

Wahyu Andhyka Kusuma [1,2,*], Azrul Hazri bin Jantan [1,*], Novia Indriaty Admodisastro [1] and Noris binti Mohd Norowi [1]

[1] Universiti Putra Malaysia; gs63875@student.upm.edu.my; azrul@upm.edu.my; novia@upm.edu.my; noris@upm.edu.my
[2] Universitas Muhammadiyah Malang; wahyukusuma@umm.ac.id
* Correspondence: wahyukusuma@umm.ac.id; azrul@upm.edu.my

Featured Application: UX Journey has the potential opportunity to be used as a universal, approach for integrating user experience and user requirements. In the long term, UX Journey has potentially become a rich source of self-improvement and self-efficacy.

Abstract: User experience and user requirements are two independent approaches. User requirements address the customer's requirements and expectations for the solution, whereas user experience encompasses all aspects of how the user interacts with and uses the software. The software product should be easy to use and has usable features. Moreover, the additional value for the software is if the product has an attractive design or working environment that is in line with user behaviors, it can occur if integrate software requirements and user experience. Integration escalates developer productivity by focusing on features that meet the user's needs and expectations. That integration improves efficiency in software development by identifying and addressing problems that may arise during the development process, saving developer time and effort in developing software. The usage context of integration of user experience and user requirements in UX Journey contributes increase developer productivity and self-efficacy in developing software by focusing development on features that match the user needs, as well as increasing efficiency in overcoming problems that arise during the development process. UX Journey makes developers feel more confident in their ability to develop quality software.

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

1. Introduction

Software process or some literature referred to as the Software Development Life Cycle (SDLC) is a necessary method of the development of software, the success of the delivery of the software products or services depends on the accuracy of selecting the right software process [1,2]. The popularity of the software processes begins with the development of pragmatic thinking about methodology in software development sequences. The software process is a collection of activities that support the development of products or services [2–4]. The popular development method is the waterfall, the classical or generic method with the requirements, analysis, design, coding, testing, and deployment activity [1,2]. However, this method is less adaptable to modern software development problems. The challenging issue with the classical method is the adaptability to quick development iterations [2,5–7], software that understands more than just what users need [5], and smaller development teams [4,5,7].

Software development is a particular process, it means that each feature of the user and business processes within the user's organization is different from the user needs of others. From previous studies, it has been explained that there is no software process that currently exists in scientific articles, books, and standards that can guarantee that the software can be delivered successfully to users [2]. Efforts to identify the features of the software requirements are important activities to ensure the success of the software. At least it was reported from previous research that there are four features that can determine it, consisting of complexity [8,9], suitability, changeability [5]–[7], and transparency. The useful developer's soft skill is to grasp the success characteristics that help the developer to increase the success of software products or services. Four primary feature that determines the success of the software product is a socio-technical skill [10–12], an important soft skill that determines to understand what users need from a more human point of view. This expertise complies with the principles of user-collaborative development appropriate in modern development methods.

User requirements in many instances become larger and change more rapidly than they should. This situation occurs because of several possibilities, including (1) changes in business processes, (2) incomplete exploration of requirements, (3) a misunderstanding of the perceptions of users, (4) strategic changes inside the client organization, (5) stakeholder engagement, (6) the development team's communication quality (7), and (8) Improved comprehension of the technical solution [2,13]. Particularly, decisions about the deployment of preventative measures demand a thorough understanding of the risk of change. In this case, the change in requirements will affect the whole system, even the rigidity of the modules in the software [14]. Rapid changes in user requirements indeed affect the quality of the software. In addition, faults and failures to identify user problems in the user requirement software influence the success of the system, particularly if these problems are found during the testing phase, this will burden the available resources more than when the system was initially analyzed [15].

The problem of eliciting user requirements occurs at various stages in the software development process. In most cases, eliciting problem arises when the developer does not thoroughly explore user requirements or does not clearly define user requirements. Hence, software product or service does not match the needs of end users or the needs of the principal business. The elicitating problem of user requirements occurs for various reasons. Several factors can cause this problem including [2,13,16] :

1. The developer did not thoroughly explore the needs.
2. The developer did not develop clear and detailed requirements.
3. The developer team did not properly identify end-user needs or underlying business needs.
4. The developer does not ensure that all requirements are covered in the development process.
5. The developer did not update requirements throughout the development process, resulting in the development of software that did not match changing requirements.

In order to avoid the problem of requirements elicitation for software, it is important to organize punctilious and regular requirements exploration from the beginning of the development process. This includes identifying end-user and business requirements, obviously defining user requirements, and ensuring that all user requirements are included in the development process. Self-efficacy is a socio-technical skill concerning a person's belief in his ability to achieve certain goals [17,18]. In the context of exploring software requirements, self-efficacy plays an important role in determining how well an individual can perform a needs assessment. Someone who has high self-efficacy in extracting software requirements will be more confident in identifying and compiling end user and business needs, and in ensuring that all requirements are included in the development process. By increasing self-efficacy in identifying software requirements, a person can improve his ability to effectively identify requirements and produce software that meets the needs of end users and the underlying business needs. Moreover, in solo or small software development, individual capacity is important for efficient and effective resource software development.

To consolidate the ongoing solutions from the previous research and bring integration of user experience and user requirement to understand the human value in user requirement, academic, and industry practitioners, a UX-UR framework has been purposed. It is named UX Journey. It is intended to serve as a unified platform that serves the individual capacity building. The purposed framework has the potential to help developers with the issues related to eliciting user requirements, quality user requirements, understanding user empathy, and practical utilization for solo and small team development.

The paper is organized in the following order. Following the introduction, Section 2 presents a research design. Section 3 details the UX Journey architecture and design. It has also obtained early feedback on UX Journey from the software practitioners in Section 4.

2. Research Design

There are several motivations for integrating User Experience and User Requirements in software development, including:

1. Improve software quality. By integrating User Experience and user requirements, the developed software will suit the end user's needs and preferences, thus providing a pleasant and rewarding experience for its users.
2. Lowering the risk of project failure. By integrating User Experience and user needs, errors or oversights can be avoided in the software development process, thereby reducing the risk of project failure.
3. Speed up development time. Integrating User Experience and user requirements allows the software development process to be carried out in a structured and efficient manner, thereby reducing the time required to complete projects.
4. Increase business value. By integrating User Experience and user requirements, the developed software can be more useful for end users and more attractive to customers, thereby increasing the business value of the companies that develop it.
5. Efficient use of resources. By integrating user experience and user requirements at one time. Developer resources can be allocated to the implementation and testing phase. Moreover, in solo or small teams, efficient use of resources increases the possibility of delivering projects on time.

By integrating User Experience and user requirements in software development, various benefits can be obtained that are useful for organizations, end users, and society in general. Therefore, integrating User Experience and user requirements is important and useful in the software development process. The current research was conducted to explore answers to the following questions:

1. How to identify and structure user requirements in software development?
2. How to evaluate and measure user experience in software development?
3. How to integrate user experience and user requirements in each stage of software development?
4. Can the integration of user experience and user requirements improve software quality, reduce the risk of project failure, and speed up development time?
5. How to develop an effective and efficient requirements extraction method or technique to integrate user experience and user needs in software development?

This research was designed to meet the challenge of implementing self-efficacy in socio-technical skills. This is very important for both solo and small software development. The expectation of this research can contribute as a framework that can be used as a standard widely in training, academia, and industry in increasing the productivity of individuals in software development.

3. UX Journey: Architecture and Design

This section describes the architecture and design of the UX Journey. There are five main points that construct the UX Journey which is explained in the following sub-sections following Usage Contexts, user, and Use Cases, Functional Description, UX Models Provided by UX Journey, Architecture, and Technical Feasibility sections. This section proved that UX Journey is formed from solid parts on the fundamentals, and it increases confidence that UX Journey is reliable to be implemented in various fields.

3.1. Usage Contexts, user, and Use Cases

3.1.1. Usage Contexts

User experience and user requirements are two different approaches in software development with an intersection in some areas. User experience covers all aspects of how the experiences and uses software, while user requirements cover the user's needs and expectations of the software. Integration of user experience and user requirements in software development aims to develop software with the characteristics followings easy use, features that suit user needs, and an attractive appearance or work environment in accordance with the user habits. The integration of the two different approaches increases the quality of using the software.

In developer productivity, integration of user experience and user requirements escalates developer productivity in developing software by focusing software development on features that meet the user's needs and expectations. In addition, developers reduce production time and effort by developing features that do not match the user's needs and are not required. That integration improves efficiency in software development by identifying and addressing problems that may arise during the development process. This improvement saves developer time and effort in developing software.

The usage context of integration of user experience and user requirements in UX Journey contributes increase developer productivity and self-efficacy in developing software by focusing development on features that match the user needs, as well as increasing efficiency in overcoming problems that arise during the development process. This will make developers feel more confident and confident in their ability to develop quality software.

3.1.1. User

UX Journey is designed to be adopted by students, academics, researchers, and the industry to increase individual skills in analyzing user needs for software requirements. UX Journey is a framework that integrates activities in user experience and user requirements to explore user needs with user experience characteristics. From the perspective of student users, UX Journey can be used as a learning path to understand the implementation of user experience in exploring user needs. Moreover, students grasp the quality of requirements explored from the perspective of attributes in user experience and attributes in software development such as usability, maintainability, etc. From an academic perspective, UX Journey can be used as a learning block in determining the material to be conveyed to students when compiling the curriculum. In addition, UX Journey bridges the gap between academia and industry. From a researcher's perspective, UX Journey is an effective method for exploring user requirements with the user experience quality attribute. Moreover, UX Journey can be used for practical as well as theoretical research. From an industrial perspective, the UX journey can be used to conduct product research or develop products with limited resources. In addition, this method ensures that the quality delivers to the user match user expectations since it involves the user in every activity. The advantages of UX Journey are designed for solo software developers to increase developer productivity and self-efficacy. However, UX Journey is suitable for teams on a small scale for efficiency in the use of resources.

3.1.1. Construct

UX Journey is an adaptation of several design thinking approaches that reliably proven in previous studies. In the last two decades, design thinking has increased to gain wider popularity in substantial fields and is considered an exciting robust paradigm for taking over a problem multidisciplinary [19]. In addition, design thinking has been recognized as a broad approach to solving socially ambiguous design problems [20]. An early definition of design thinking by Cross et al. [21,22] describes design thinking as the study of the cognitive processes embodied in the act of design, as well as something inherent in human cognition [22]. According to Dunne and Martin [23], design thinking is the way designers think and apply their mental processes to design objects, services, or systems, which differ from the end result of elegant and usable products.

The design has a broad cognition, more than specified in the field of software development, practically all aspects of various disciplines have design concepts. Design is defined as a conceptualization of process (action), creation (artifact, product, system, or service), planning, intention, etc. Referring to the more general standard, IEEE 610.12-90 [24], design is defined as a process for exploring the architecture, components, interfaces, and other characteristics of a system or component and the results of that process.

Research conducted by Brown [25] explains that design thinking is a user-centered design innovation approach. In addition, design thinking takes a sensitive approach to the user. Design thinking is also widely recognized as a designer's method of matching user needs with what approach is technologically feasible and what can be changed by business strategy so that it is feasible to add value to customers and capture market opportunities. The research conducted by Brown provides a broader perspective on the design thinking method, where in this research design thinking is used to view design perspectives from several different points of view. The first concept is design thinking to be an approach used to create new viable solution situations without leaving value from customers or improving solutions to meet customer needs with added value. The second concept uses design thinking to become an approach to design. This is a consensus among developers because thinking about design is an integral part of planning or design. This strong background inspires ideas on how designers develop design processes with a creative mind toward design solutions and find new opportunities. Figure 1. is an approach to Design Thinking expressed by Brown [25].

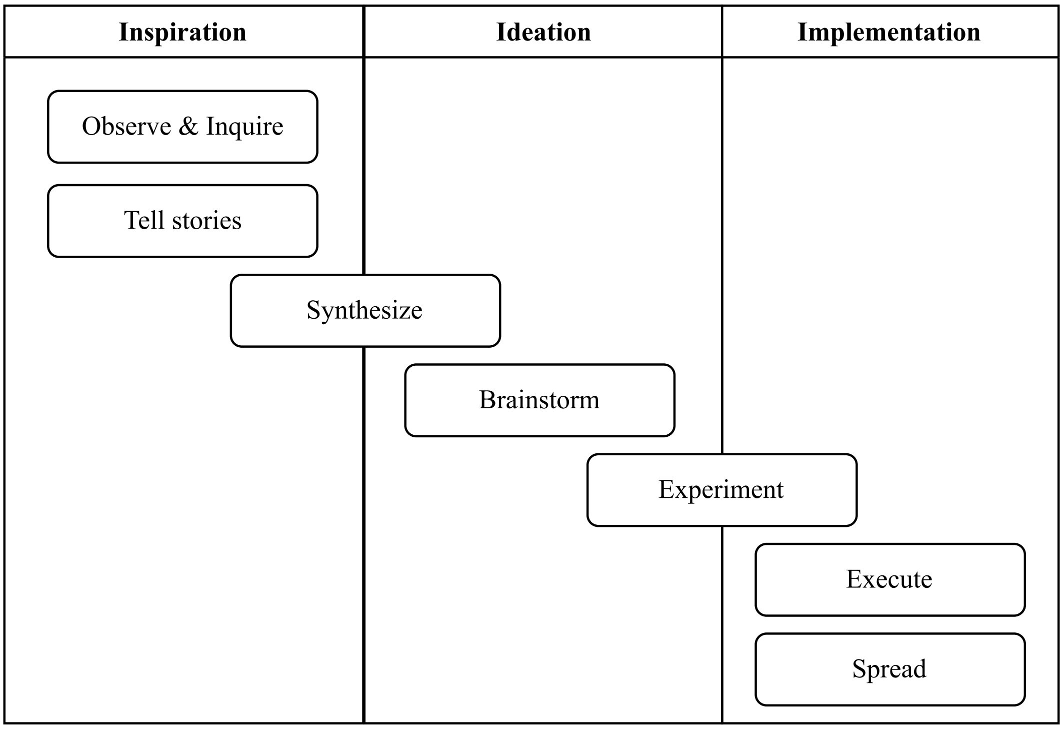


Figure 1. Design Thinking is expressed by Brown [25].

To be able to understand the context of how developers can think about design, the concept of design itself needs to be brought to the fundamental aspects that can explain how each aspect can be interrelated and interact. Design activity is visible as a unified process aimed at producing design object specifications. Think critically about a design by considering the attention to several aspects. The first is the environment in which the objects of that design will exist. In this case, it will determine where the design will be used, in this aspect it can be proof that a design is something that has unique characteristics and environments. The second is the goal that is ascribed to the object in the design. The purpose of thinking about or organizing the design starts from the problems that exist in the user. The third is the desired structural and behavioral properties of the design object (requirements). The third aspect of this design relates to user requirements and the expectations that users embed in requirements. The fourth aspect is the collection of component types (primitives). The fifth aspect is the constraints that might limit acceptable solutions.

In addition, design objects are the result of designs such as artifacts, products, systems, or services as well as software products whether it is a collection of lines of code, database queries, and algorithms embedded in a product. Broadly speaking the design concept is shown in Figure 2. Where the form of the design itself is an artifact in a certain context, in Figure 2 it is shown as a Domain. Where the user requirements influence each other from the design concept.

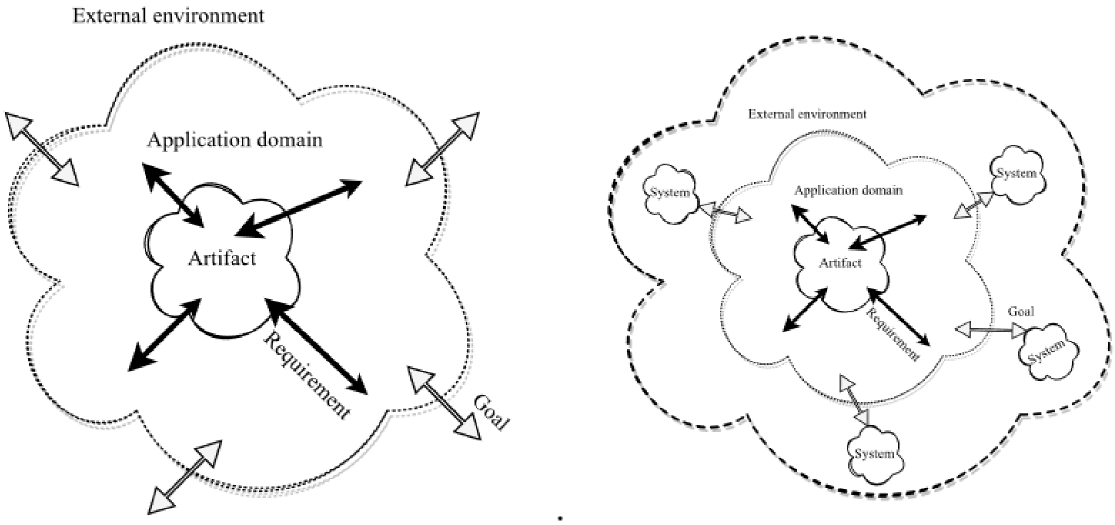


Figure 2. (a) Design concept related to artifact, domain, and external environment. **(b)** Design and Correlation with other systems [26].

In a design, developers shall think holistically because a system depends on and interact with other systems. In this research, we use the term system, it can represent general terms for example business processes, work environments, and software. As shown in Figure 2 that a design is in the application domain. Furthermore, a design should satisfy the design requirements of an application domain originating from the external environment as shown in Figure 2. Consider a holistic view of design is necessary to perform the user goals. The interrelationships between designs and the external environment influence user needs and expectations.

Design aspects as represented earlier have unique characteristics with regard to users, stakeholders, and different environments. Developers should focus on acquiring a deeper understanding while consistently accommodating the entire external environment holistically. In general, design thinking has accommodated this problem by focusing on several areas. The first area is empathy, in this area the developer considers at the context of the user needs and expectations holistically from the perspective of various and multiple users. The second area is integrative thinking. Hence, the developer should present creative thinking from all aspects both within the scope of the application domain and the external environment. The third area is experimentalism, to be able to determine which solution to choose as a potential solution, developers are competent to explore situations in creative ways for novel solutions. The fourth area is optimism, in the high level of thinking creatively, a developer is required to be optimistic about the solution that has been chosen and be capable to maintain it as a solution to the problem. The last area is collaboration, where developers can collaborate with interdisciplinary stakeholders for innovative solutions.

Representing a design in a project requires in-depth analysis to obtain a model that is relevant to user problems. Model or framework in design thinking is a couple of various activities to solve design problems. The approach used to represent in-depth analysis problems is the Divergent-Convergent Inquiry-based Design Thinking Model (DCIDT) which was introduced by Eris [27]. This model (shown in Figure 3) describes design thinking as two cognitive approaches related to fundamental modalities, divergent and convergent thinking.

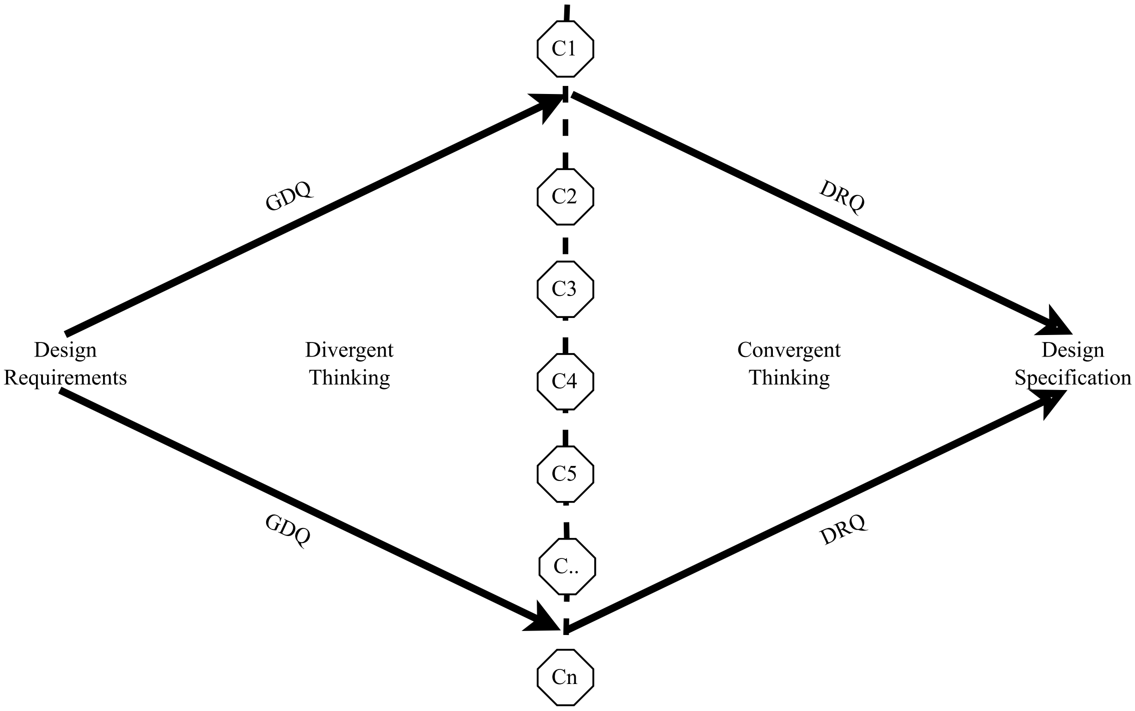


Figure 3. Divergent-Convergent Inquiry-based Design Thinking Model [26,27].

The DCIDT model sequentially transforms user requirements into design specifications according to user needs and expectations. Transformation of user requirements to design solutions through a collection of questions, which is in divergent thinking and convergent thinking. A detailed explanation of the DCITD start with changing the design requirements through Generative Design Questions (GDQ) into a series of design concepts. The shifting process is carried out by answering several questions converted into a design concept. Additionally, the GDQ is used to create, synthesize, and extend several potential design concepts. The basic knowledge of the GDQ as exposed by Eris [27] contains various questions that are used to explore design concepts from design requirements. Furthermore, the design concept that has been obtained (C1…Cn) is converted into design potential and design specifications through a series of questions in Deep Reasoning Questions (DRQ). DRQ in the DCIDT model is to analyze, evaluate, and validate design concepts (Cs) to make design decisions or potential designs that are feasible according to the user's design needs and expectations, classification of the DRQ and GDQ questions as shown in Table 1.

Table 1. Question Classsification [28].

Question Class	Aristotle	Dillon	Lehnert	Graesser	Eris
Low-level Questions	Affirmation	Affirmation	Verification	Verification	Verification
		Identification			
	Nature	Definition		Definition	Definition
				Example	Example
	Description	Description	Feature Specification	Feature Specification	Feature Specification
			Concept completion	Concept completion	Concept completion
			Quantification	Quantification	Quantification
		Rationale			
		Concomitance	Disjunctive	Disjunctive	Disjunctive

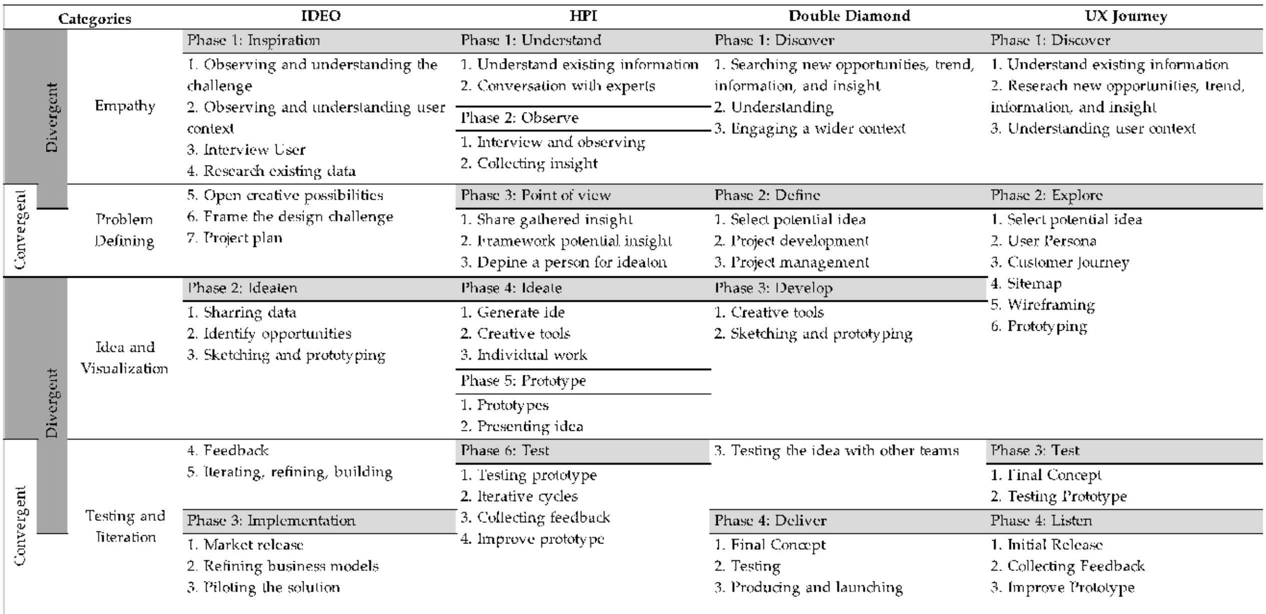
		Equivalence Difference		Comparison	Comparison
			Judgmental	Judgmental	Judgmental
Deep Reasoning Questions (DRQ)	Reason	Function	Goal Orientation	Goal Orientation	Rationale/Function
		Relation Correlation		Interpretation	Interpretation
		Conditionalit y Causality	Causal antecedent	Causal antecedent	Causal antecedent
			Causal consequent	Causal consequent	Causal consequent
			Expectational	Expectational	Expectational
			Procedural	Procedural	Procedural
			Enablement	Enablement	Enablement
Generative Design Questions (GDQ)					Method generation
					Scenario creation
					Ideation
		Rhetorical		Assertion	
			Request	Request	Request
		Deliberation			
		Unspecified			
		Unclear			

Developers consider a problem broadly and holistically to be able to accomplish the design requirements and expectations of users. Design thinking is described as a non-linear and repetitive activity. In this research, the process of design thinking is grouped into four main activities, consisting of empathy, problem framing, ideas and visualization, and experiments [29–35]. A more detailed explanation of each of these groups is as follows:

a. Empathy (divergent phase): Empathy is an ordinary activity that is often used in user experience activities, particularly in design thinking. In addition, empathy is an activity that is used to understand, observe, and find user problems and more closely with design users. In conducting empathy, developers carry out two main activities, consisting of primary exploration by eliciting design requirements problems directly to users. In addition, empathy is used to find out the user's emotions and expectations of a product. The next activity is to carry out secondary exploration, by collecting data on standard operational procedures in organizations, statistical data, scientific and popular literature, and solutions that are already available in the market.
b. Problem defining (divergent and convergent): activities that intersect with the two phases of design thinking. Activities in this category have been narrowed down to several design solutions. This activity can be grouped into two main sub-activities. The first is a review activity of the solutions generated in the empathy section. The developer will review some primary and secondary data to be classified based on proximity and possible problem solutions. Review activity is categorized as a divergent phase because it has not yet led to a solution provided to the user. The second is an activity to frame the results of the classification at the review stage so that they can be grouped together with potential solutions.
c. Idea and visualization (divergent phase): visualization is a common activity in representing a design, including the user experience. Idea and visualization are included in the divergent phase category because in this activity there is no solution that has emerged as a design solution. This activity is classified into two activities, the idea is to be able to collaborate with users to bring inspiration from the design problems they face. The second activity is visualization. In general, in this activity, the developer will create a low-fidelity or high-fidelity display. Furthermore, the visualization generated by the developer becomes a prototype for the technical specification of a potential solution.
d. Testing and iteration (divergent and convergent phase): this is an activity to explore user problems and test the potential solutions that have been obtained. This activity can be divided into three parts. The first activity is to test the usability of a potential solution from a predefined

design. This activity will determine whether the potential solution meets the design requirements and expectations of the users. The second activity is evaluation and improvement. Is an iterative activity to get a potential solution that is ready to be developed into a product. All design requirements and expectations from users have been fulfilled at this stage of the activity. The third activity is the delivery of potential solutions. The results of the potential solutions that have been developed are sent to the development team to implement in a product that is ready to compete in the market and provide solutions to organizations.

Table 2. Divergent and convergent phases compared with the previous study.



Categories		IDEO	HPI	Double Diamond	UX Journey
Divergent	Empathy	Phase 1: Inspiration 1. Observing and understanding the challenge 2. Observing and understanding user context 3. Interview User 4. Research existing data	Phase 1: Understand 1. Understand existing information 2. Conversation with experts Phase 2: Observe 1. Interview and observing 2. Collecting insight	Phase 1: Discover 1. Searching new opportunities, trend, information, and insight 2. Understanding 3. Engaging a wider context	Phase 1: Discover 1. Understand existing information 2. Reserach new opportunities, trend, information, and insight 3. Understanding user context
Convergent	Problem Defining	5. Open creative possibilities 6. Frame the design challenge 7. Project plan	Phase 3: Point of view 1. Share gathered insight 2. Framework potential insight 3. Depine a person for ideaton	Phase 2: Define 1. Select potential idea 2. Project development 3. Project management	Phase 2: Explore 1. Select potential idea 2. User Persona 3. Customer Journey 4. Sitemap 5. Wireframing 6. Prototyping
Divergent	Idea and Visualization	Phase 2: Ideaten 1. Sharring data 2. Identify opportunities 3. Sketching and prototyping	Phase 4: Ideate 1. Generate ide 2. Creative tools 3. Individual work Phase 5: Prototype 1. Prototypes 2. Presenting idea	Phase 3: Develop 1. Creative tools 2. Sketching and prototyping	
Convergent	Testing and Iteration	4. Feedback 5. Iterating, refining, building Phase 3: Implementation 1. Market release 2. Refining business models 3. Piloting the solution	Phase 6: Test 1. Testing prototype 2. Iterative cycles 3. Collecting feedback 4. Improve prototype	3. Testing the idea with other teams Phase 4: Deliver 1. Final Concept 2. Testing 3. Producing and launching	Phase 3: Test 1. Final Concept 2. Testing Prototype Phase 4: Listen 1. Initial Release 2. Collecting Feedback 3. Improve Prototype

3.2. UX Models Provided by UX Journey

All models principally accommodate divergent and convergent phases in each of their model activities. In addition, the sequence of the phases is relatively the same starting with a divergent phase and ending with a convergent phase. Besides that, in general, there will be overlapping of the two phases: divergent, convergent, divergent, and convergent so that in some phases there will be overlap. Table 3 is several models and processes that exist in these activities.

Table 3. Several design thinking models.

Year	Source	Process
2001	IDEO, Deloitte	Understand and observe-Synthesise-Visualise-Prototype, Evaluate, Refine-Implement
2005	Design Council UK, Charity for Strategic Design	Double Diamond (Understand-Define-Explore-Create)
2005	British Design Council	Discover-Define-Develop-Deliver
2006	Dunne and Martin	Induction-Abduction-Deductions-Test
2008	Brown	Inspiration-ideation-implementation
2009	Frog2	Imagine-Make-Scale
2010	Hasso Plattner Institute of Design at Stanford	Empathize-Define-Ideate-Prototype-Test
2010	Google Design Sprints (I)	Understand-define-diverge-decide-prototype-validate
2010	Austin Center for Design AC4D, Educational Program	Ethnography-Synthesis-Prototyping
2010	DEEP Design Thinking, Design Educator Mary Cantwell	Discover-Empathize- Experiment-Produce

2011	Desain school Paris	Inspiration (Understand, Observe, POV)-Ideation (Ideate, Prototype, Test)-Implementation (Storytelling, Pilot, Business Model)
2011	Hasso Plattner Institute of Design	Understand-Observe-Point of View-Ideate-Prototype-Test
2011	Kolko	Define-Discover-Synthesise-Construct-Refine-Reflect
2011	Lean Startup	Build-Measure-Learn
2012	IDEO, International Design and Consulting Firm	Discovery-Interpretation-Ideation-Experimentation-Evolution
2012	HCD - Human Centred Design	Hear-Create-Deliver
2013	Google Design Sprints (II)	Understand-diverge-decide-prototype-validate
2013	SAP, Software Programming Company	Plan-Research-Design-Adapt-Measure
2014	Designing for Growth	What is?-What if?-What wows?-What works?
2015	Pontis	Understand-Empathise-Define-Ideate-Prototype-Test
2016	Journalism	Pitch-Assign-Report-Publish-Feedback
2016	Meinel, Von Thienen	Empathise-Define persona-Ideate-Test prototypes-Bring home

To shape the construct of the UX Journey, consider several previous models that are tailored to the goals of the UX Journey, which focus on increasing self-efficacy to competence in understanding the intent of the user, or proficiency in actively capturing market potential with creative product solutions. In the model presented by Brown, constructs design thinking through three activities, consisting of inspiration, ideation, and implementation where each activity is supported by sub-activities as an orderly and directed stage. Inspiration activities motivate developers to explore context with an empathetic and user-centered approach to identify design problems and explore potential design idea opportunities. The results of the identification and observation implement into the ideation process by generating and testing ideas into potential solutions. The last stage is to carry out implementation to realize a feasible solution for the potential idea.

Most of the design thinking models are influenced by the mental aspects of the users. These aspects include cognition, affection, and conation. Moreover, paying attention to the emotional aspect of the user has become an important consensus in the design [36]. The process of paying attention to the emotions of the user leads to empathy as a design solution to the problem at hand. This inspires developers to create design solutions according to user needs and expectations [37,38]. Moreover, design thinking is considered as a complex design activity from a cognitive, motivational, and emotional process of developers and users [26,39].

The various models have common characteristics, and consist of holistic contextual exploration, idea creation, and evaluation of solutions and the realization of solutions generated in the context of potential design ideas. In this case, the nature of the design thinking process based on the purpose of the model awakens the natural potential to reframe it into new solutions or improve solutions from the model with ideas that are better suited to goals centered on individual innovation for contextual problems.

The framing of a model emphasizes the process of identification and reflection through appreciation, action, and re-appreciation [26,40]. Framing in design solutions is used to find a model that fits the purpose of the model to explore contextually. In addition, framing is an approach to capture, analyze, and create knowledge from new perspectives. A study conducted by Adikari et al. [26,32] revealed that the framing process is an activity that provides new potential for more specific problems from different realities. There are three ways to frame the process. The first realign the existing frame into a new perspective. This means that the existing model is rearranged from the components in the model. The second shifts the perspective of the model into a new domain usually used to represent an idea for different or interdisciplinary knowledge. The third is to draw some connections from the activity in the model and conclude that it is a new or the same activity in a different context.

Design solutions are products of user problems or creative ideas from the results of research on market needs. Sometimes, users of the design are not established at the beginning of exploration or observation, but the solution is an initial product that is used to test the market acceptance of a product. Successful design solutions focus on three important elements that reinforce each other at the outset, consist of insight, observation, and empathy [25]. Insight and observation, the existence of a design solution that leads to products and services, and empathy are design challenges to bring up creative design thinking that reflects the emotional aspects and experiences of all users in context. In UX Journey, these three elements are framed in the discover element, which is the first activity in the proposed model.

Elements of interpretation, ideas, experiments, and prototypes are activities that are used by developers to find and provide potential design solutions that are used to solve problems. Moreover, in this activity, is possible for developers to explore creative design ideas from various holistic perspectives. Design solutions are realized in the form of product design prototypes that can be used for market exploration or provide an overview of the solutions provided. In UX Journey, these elements are framed in explore element, which is the second activity that the main function to explore existing design problems.

Testing is a general activity that is always used to convince developers that the design solutions created to meet the needs and expectations of users, or in the form of products and services testing can be used to determine market acceptance of the proposed product or service. In the UX Journey, it is a mandatory activity that becomes a consensus to know that design solutions are useful and quality guaranteed.

The elements of refining, reflecting, and evolution of potential design solutions are useful for developing solutions if discrepancies occur or changes are found that may occur during solutions exploration. In addition, this element is to prepare the results of the tests carried out to find out whether the solution is feasible to continue into a product. In UX Journey, this element is framed as a listening element. It is the last activity before a new or continued cycle of the UX Journey model.

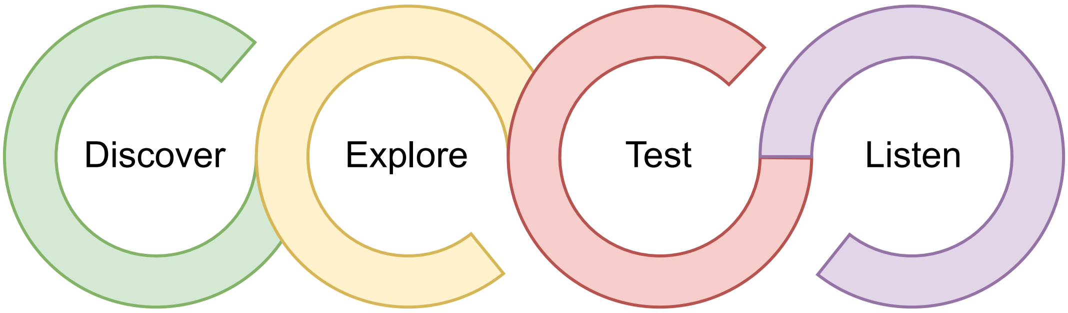


Figure 4. Purposed UX model with UX Journey.

3.3. Functional Description

The functional description of the UX Journey is shown in Figure X. It was developed from robust fundamentals to describe the need to improve individual socio-technical skills in industry and academic challenges.

UX Journey is designed specifically for the personal in solo or small development teams to integrate activities in user experience and user requirements to satisfy user needs and expectations with the experience characteristics. Moreover, UX Journey is a combination of psychomotor and cognitive aspects for the developer to increase self-efficacy. The functionality of the UX Journey is formed from four main structures based on the design thinking model. The constructs of the UX Journey are discovering, exploring, testing, and listening.

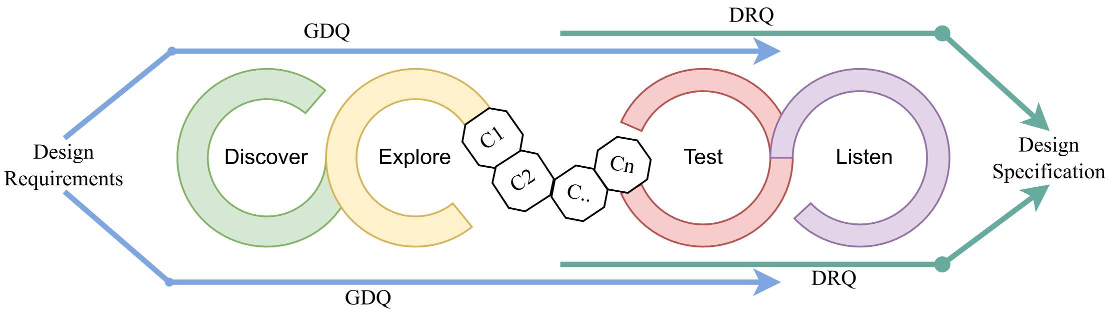


Figure 5. UX Journey functional description.

The activities of the UX Journey begin with finding existing design requirements. In this context, design requirements are obtained from users or organizations as well as research on a product that will compete in the market. The process of finding design requirements is carried out through a series of activities. The developer should explore various possible solutions and start testing to obtain solutions that match their needs and expectations. In the last stage, the developer will organize listening, activities to obtain an overview of the potential solutions, and whether the solutions are appropriate and can be continued to become specifications of the design.

3.4. Architecture

The functional model in Figure 5 is converted into a detailed technical architecture as shown in Figure 6. There are four main components in UX Journey, consist of discover, explore, test, and listen. Each main activity shows a sub-activity, which implements the user experience method that provides a coherent set of quality elicitation methods. Detailed sub-activity on the UX Journey technical architecture is shown in Figure 6 as follows:

a. Discover, there are three activities that establish in discover that also have an intersection with Explore activity. SWOT Analysis is a sub-activity that is used as a feasibility study to identify the project eligibility, Competitor Analysis is used to gather information from the competitor that exists in the market, and Hypothesis predefined scope and goals for the project.
b. Explore, is the main activity with lots of sub-activity. Identify behavioral variables, Prepare and Select Questions, Index Cards, Map Interviews, Findings, Significant Behavior Patterns, Expand Description and Variable, Synthesize Characteristics and Relevant Goal, Check for Redundancy and Completeness, Wireframing, Sitemap, User Scenario, Persona, Customer Journey, Prototype
c. Test, Testing is an activity in the UX Journey that is useful for ensuring that the design solution meets the needs and expectations of users.
d. Listen, although this activity is basically placed outside the design solution process, listen has an important assignment to provide an overview of the response from the market when the product is released. Moreover, to find out how the product can be developed to the next versions it is necessary to get user feedback.

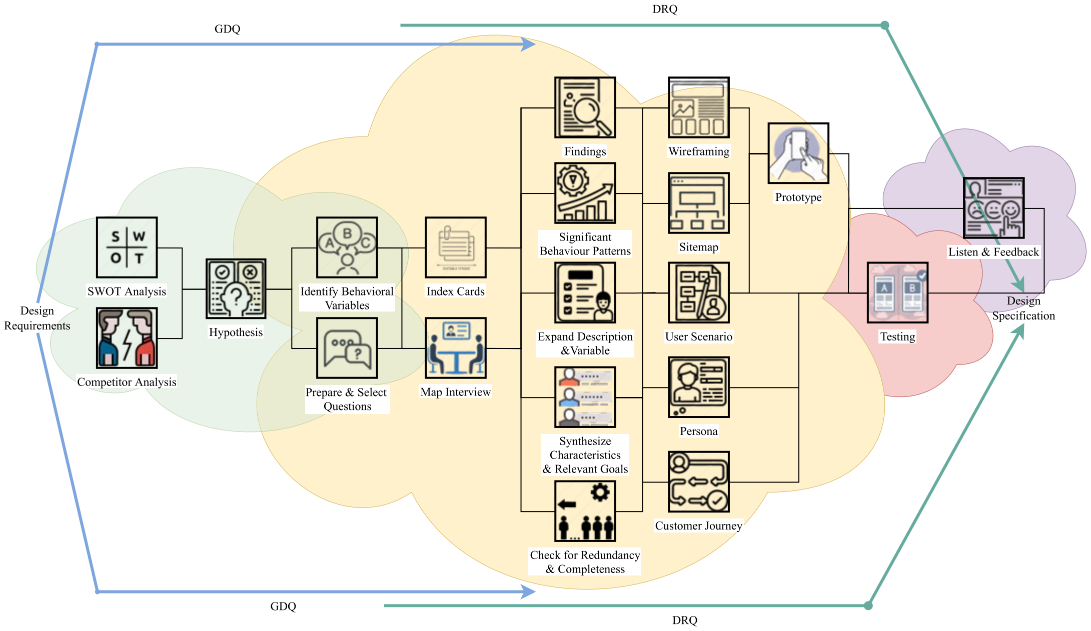


Figure 5. UX Journey architecture.

3.5. Technical Feasibility

A review of the UX Journey architecture characteristics indicates that the architecture is reliable, comprehensive, and useful for increasing developer self-efficacy. Moreover, developers can discover their abilities in exploring user needs, and at the same time, the socio-technical abilities of developers to know the needs and expectations of users can be improved. The next stage of the research is conducting a feasibility study. The purpose of a feasibility study is to observe, explore, and define the strengths and weaknesses of an existing or new process or method, including the possibilities that can be obtained from the process or method and the possible challenges in its implementation [41] objectively and rationally. In addition, the feasibility study is also used as a process of deciding from an evaluation whether a process or model is feasible to implement.

The process of conducting a feasibility study through several stages, according to Avison et al. [41] four stages must be passed to carry out a feasibility study, consisting of:

1. The process of preparing a feasibility study activity. In this first stage, it will be determined how extensive the object of study will be measured.
2. Define the problem. The second stage is to compare the requirements of the problems faced with the current situation.
3. Choose the feasibility study option. Stages to determine the available alternatives and choose the solution used.
4. Make a report of the feasibility study. Steps to represent the feasibility study in a document.

The ability of a feasibility study to identify and evaluate alternatives to propose a method or model increases the need for a feasibility study. Moreover, the feasibility study process can provide identification and alternative solutions to optimally implement in terms of what is practicable, feasible, and one that will address the relevant legal requirements [ssegawa]. The process of determining an optimal solution requires several element evaluations including the level of risk, costs, and benefits of alternatives based on several feasibility areas. From previous research, there has not been a general agreement on several domains where feasibility must be carried out. However, there is some convergence in the five general areas known as TELOS (technical, economic, legal, operational, and schedule). The following is a feasibility study analysis of the UX Journey using the TELOS approach.

3.5.1. Technical feasibility

Technical feasibility tests the possibility of implementing available technology or implementing new technology if needed. The results of the technical feasibility are used to answer several questions including whether the technology has a satisfactory effect, whether there is an increase in time, and whether there is an increase in performance that has been completed.

Table 4. 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 contect	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, listen: user collaborative
Product improvement (Phase)	Implementation: refining business models	Test: iterative cycles, collective feedback every time	Deliver: final concept and launching	Listen: launching market analysis product, review user feedback.

The performance improvement of existing models has generally been fulfilled in the proposed UX Journey model, moreover that the proposed model is a model that is actually used to improve individual abilities. Different from other models, where the focus of other models is on the user or the creation of a product. For the UX Journey, apart from focusing on the needs and expectations of users, it is also used to increase the confidence of developers through socio-technical skills by collaborating a lot with users.

3.5.2. Economic feasibility

Aims to evaluate whether there is an economic effect resulting from the proposed new technology compared to existing technology today. The reliability of technology will be measured based on the process carried out without knowing or prejudicing the results of the evaluation carried out. When conducting an economic feasibility evaluation, there are expected results. In general, there are two approaches used [drijaca]. The first approach is through measurable effects. The indicators of this approach are based on numbers or trends of something that can be measured such as the number of employees, cost reduction, output improvement, service improvement, etc. The second approach is effects that are impossible to measure, in this case in the form of risks, or problems that cannot be measured but can occur when activities are carried out.

The proposed UX Journey model has an advantage in the number of personnel. The UX Journey model was originally designed to be dedicated to individuals or solo, this was motivated by the need to increase individual abilities to be able to interact with users. Moreover, the UX Journey is designed to be executed within 16.3-25 hours from the start of the process to testing for the design solution.

3.5.3. Legislative feasibility

This aspect defines the legal requirements to introduce or implement new technology. Legal feasibility must provide answers to form the basis for carrying out a quality evaluation of the technology. The legal basis used in this research is shown in Table 5.

Table 5. Several standards in legislative feasibility.

Code	Standard
ISO/IEC 27000	Information security management systems — Overview and vocabulary
ISO/IEC 27001	Information security management systems — Requirements
ISO/IEC 27002	Code of practice for information security controls
ISO/IEC 27003	Information security management system implementation guidance
ISO/IEC 27004	Information security management — Measurement
ISO/IEC 27005	Information security risk management
ISO/IEC 27006	Requirements for bodies providing audit and certification of information security management systems
ISO/IEC 27007	Guidelines for information security management systems auditing
ISO/IEC TR 27008	Guidelines for auditors on information security controls
ISO/IEC 27010	Information security management for inter-sector and inter-organizational communications
ISO/IEC 27011	Information security management guidelines for telecommunications organisations based on ISO/IEC 27002
ISO/IEC 27013	Guidance on the integrated implementation of ISO/IEC 27001 and ISO/IEC 20000-1
ISO/IEC 27014	Governance of information security
ISO/IEC TR 27015	Information security management guidelines for financial services
ISO/IEC TR 27016	Information security management — Organisational economics
ISO 9241-11:2018	Ergonomics of human-system interaction
ISO 25010	Usability

3.5.4. Operational feasibility

Operational feasibility aims to evaluate the implementation of new technology compared to the initial state within an organization. The result of this feasibility study is to identify whether the technology can be implemented within the organization. In this case, UX Journey is a model that is designed to be implemented for individuals. Therefore, in this feasibility study, the hierarchical structure associated with this model is the user and the developer. However, there is a possibility that this model will be used in academics (Figure 6 (a)), small teams, and individuals in training (Figure 6 (b) and (c)), therefore the organizational structure of the designer is used for the feasibility of this model.

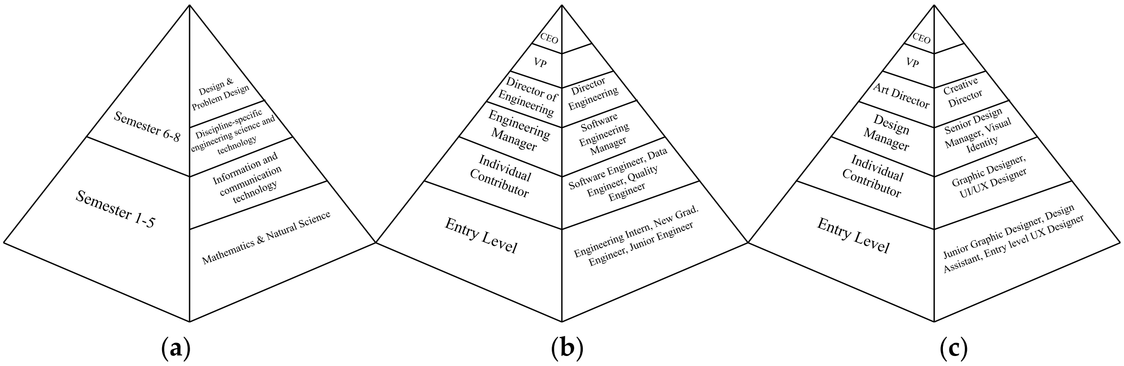

Semester 6-8
Semester 1-5
Design & Problem Design
Discipline-specific engineering science and technology
Information and communication technology
Mathematics & Natural Science
(a)
CEO
VP
Director of Engineering
Engineering Manager
Individual Contributor
Entry Level
Director Engineering
Software Engineering Manager
Software Engineer, Data Engineer, Quality Engineer
Engineering Intern, New Grad. Engineer, Junior Engineer
(b)
CEO
VP
Art Director
Design Manager
Individual Contributor
Entry Level
Creative Director
Senior Design Manager, Visual Identity
Graphic Designer, UI/UX Designer
Junior Graphic Designer, Design Assistant, Entry level UX Designer
(c)

Figure 6. Operational feasibility (a) academic structure (b) software engineer structure (c) designer structure [42].

3.5.5. Schedule feasibility

Evaluating the overall performance of a technology provides an idea of how the technology can be adapted to its current state. The feasibility schedule can indicate the requirements for implementing the technology.

Table 6. UX Journey schedule feasibility.

Process	Time (minutes)		Activity	Document
	Min	Max		Name
Empathy & Define	30	60	SWOT Analysis	SWOT Analysis
	10	30	Prepare Questions	
	10	30	Selected Questions	List Question
	30	60	Competitor Analysis	Competitor Analysis
	20	30	Hypotheses	Hypotheses
	20	30	Identify Behavioral Variables	Identify Behavioral Variables
	60	60	Persona	Persona
Ideate	20	30	Findings	
	60	120	Index card/ Sticky notes	Index card/ Sticky notes
	30	30	Map Interview	Map Interview
	30	30	Significant Behaviour Patterns	Significant Behaviour Patterns
	30	30	Synthesize Characteristics and Relevant Goals	Synthesize Characteristics and Relevant Goals
	30	40	Check for Redundancy and Completeness	Check for Redundancy and Completeness
	30	30	Expand Description and Variable	Expand Description and Variable
	60	60	Customer Journey	Customer Journey
	120	240	Wireframing (Low Fidelity)	Wireframing (Low Fidelity)
Prototype	60	120	User Scenario	User Scenario
	30	30	Sitemap	Sitemap
	240	320	Mockup (High Fidelity)	File Mockup
Test	60	120	Testing	Testing
Total (minutes)	980	1500		
Total (hours)	16,3	25		

4. Academic and Industry Perception Study

In this study, to get initial perceptions from industry and academics about UX Journey, a survey was used to complete the feasibility study of UX Journey before it was implemented in academia and training. This section describes the survey and its results in detail.

4.1. Detail of the Study

This study was conducted on a limited sample (n = 7), with professional experience of 10 to 15 years in the software field, respondent profile as shown in Table 7. The selected respondents are individuals who are responsible for developing software work products and have the authority to develop human resources where the respondents work. In addition, respondents have the necessary educational background in the field of computer science or information systems and use and are aware of the importance of the software development life cycle, resource management, software quality improvement, and the effects of software failures. The selected respondents represent organizational policies to determine the career development of developers, starting from recruitment, and training, to retirement preparation. Each respondent was selected using a judgmental sampling technique and each respondent agreed to participate in this study by guaranteeing the confidentiality and anonymity of their name and the organization where they worked.

Table 7. Respondent profile.

Respondent	Industry	Location	Employe	Experience (Years)	Education
Respondent 1	Property	Singapura	5-10	10	Bachelor of Computer Science
Respondent 2	Telecomunication	Indonesia	>50	10	Bachelor of Computer Science
Respondent 3	Cloudcomputing	Singapura	10-50	10	Bachelor of Computer Science
Respondent 4	Cryptocurrency	Indonesia	5-10	16	Master of Computer Science
Respondent 5	Software house	Singapura	5-10	12	Bachelor of Computer Science
Respondent 6	Software house	Indonesia	<5	13	Bachelor of Computer Science
Respondent 7	Startup education	Indonesia	<5	13	Master of Computer Science

There are five stages in getting the results of the survey conducted. The first stage is to communicate with potential respondents about their willingness to participate in the research being conducted. The second stage is the collection of detailed demographic data and verification of the competence of the respondents. Some of the criteria for verification carried out include, the name of the respondent being included on the company website or valid evidence that the respondent is indeed working and has the required competence, verification of the respondent's educational data by tracing the history of the respondent's scientific work. As already mentioned, this research uses a survey to find out initial perceptions of the UX Journey implementation and perspectives from the industry when this model is used. To explain this phenomenon, several input and output variables are tabulated, as shown in Table 8.

Table 8. Input and Output Variables.

Input Variables	Output Variables
Name, age, gender, education, SDLC knowledge, knowledge of cost quality, knowledge of software requirement, knowledge of user experience, open-ended question about challenges to build design solutions, self-efficacy, and UX Journey.	SDLC knowledge, response to each survey question, and list of challenges to build design solutions, self-efficacy, and UX Journey.

The third stage is taking a detailed survey by starting with giving an explanation of the data collection process that was carried out. Retrieval of detailed survey data was carried out in two ways: interviewing and filling out forms because some respondents did not wish to be interviewed in person. The details of the questions asked of respondents are as follows:

1. Understanding of respondents to the software development cycle
2. Understanding of respondents to reduce production costs by combining several activities
3. Respondents' experience of needs that do not match the needs or expectations of users
4. Understanding the risk of failure to identify needs
5. Implementation of increasing the competence of developers in the company
6. Respondent experience combining user research activities in user experience with user requirement activities.

The fourth stage is giving respondents two architectures, the first is the design thinking model by Brown and the second is the UX Journey model. Respondents were given seven days to respond to the two models as to which model is more effective for combining user research activities in user experience with user requirement activities, as well as an effective model to be used to increase individual competency or used by a single developer for product research.

This study was designed using the protocol at several stages that have been described, however, some challenges must be faced to ensure the validity. Therefore, several attempts were made to maintain the validity through several approaches. Threats to construct validity, research questions require a different approach to validity, generally, a logical approach is used to find out that there is consistency between questions and answers. In the research conducted, a superficial approach was used, by asking directly respondents whether the questions asked correlated with the research or not, moreover, the questions were given straightforwardly and each answer from the second stage would be compared with those carried out using a Likert scale so that construct validity was formed.

The fifth stage is to verify the respondents' answers using a Likert scale for validation using a logical approach. Results are tabulated and summarized for each response to the questions asked. To get two categories of analysis combined in each one analysis. In addition to the challenges to the two threats above, the potential for bias in generalizing results, and sampling assessments may occur, therefore variations are needed from several respondent backgrounds both in demographics and competency abilities. Future studies also need to be carried out; therefore, a comprehensive report is needed to identify and repeat the research carried out.

4.2. Analysis of the Results of the Survey

4.2.1. Understanding of respondents to the software development cycle

The first question in the survey conducted by respondents was asked about how effective a software development cycle is for maintaining the continuity of activities in software development. From the survey results, it was found that there was no denying that the software development cycle is important to very important to maintain the continuity of activities. 86% of respondents as shown in Table 9 believe that using the software development cycle is effective. Verification using interview results stated that there was no objection to the survey results where all respondents stated that modern methods such as Agile, Scrum, and XP were always used for development activities in their organization.

Table 9. Survey result: respondent understanding of the software development cycle.

Question 1: Understanding of respondents to the software development cycle				
Do you believe that a software development cycle is effective to maintenance software development activity?				
Strongly Believe	Believe	Neutral	Disbelieve	Strongly Disbelieve
6	1	0	0	0
86%	14%	0%	0%	0%
Group 1:	Believe 86%+14%=100%, N1: 6+1=7			
Group 2:	Disbelieve 0%+0%=0%, N2: 0+0=0			

4.2.2. Understanding of respondents to reduce production costs by combining several activities

The results of the survey as shown in Table 10 stated that 85.71% of respondents or six respondents believed that combining several activities in software development could reduce production costs, while one respondent did not believe this. Verification by interviewing respondents resulted that all respondents had combined several activities in software development, however one respondent who did not believe at the time of the survey stated that combining several activities was not proven to reduce production costs, but this could speed up time this was done by adding resources.

Table 10. Survey result: respondent understanding to reduce production cost.

Question 2: Understanding of respondents to reduce production costs by combining several activities				
Do you believe that combining activities in software development can reduce production costs?				
Strongly Believe	Believe	Neutral	Disbelieve	Strongly Disbelieve
1	5	0	1	0
14.29%	71.43%	0%	14.29%	0%
Group 1:	Believe 14.29%+71.43%=85.71%, N1: 1+5=6			
Group 2:	Disbelieve 14.29%+0%=14.29%, N2: 1+0=1			

4.2.3. Respondents' experience of needs that do not match the needs or expectations of users

The results of the survey as shown in Table 11 stated that 85.71% of respondents or six respondents had an experience that software requirements did not match user needs or did not meet user expectations. The results of the verification state that from Group 1 the problem of not meeting user needs is due to changes in requirements that are too fast and not balanced with development time, besides that to align between user needs and user expectations is a challenge because basically, the expectations of users are in software that already available. What's more, users always believe that the new system will definitely add to their volume of work and be difficult to use. One respondent in Group 2, has experience in startup companies, where products are produced from market research results and products are relativized quickly to get feedback from users.

Table 11. Survey result: developer experience of software requirement.

Question 3: Respondents' experience of needs that do not match the needs or expectations of users				
In your experience how many times do software requirements that do not match the needs or expectations of users?				
Always	Often	Sometimes	Rarely	Never
5	1	0	1	0
71.43%	14.29%	0%	14.29%	0%
Group 1:	Always 14.29%+71.43%=85.71%, N1: 5+1=6			
Group 2:	Never 14.29%+0%=14.29%, N2: 1+0=1			

4.2.4. Understanding the risk of failure to identify needs

The results of the survey as shown in Table 12 state that all respondents believe that failure to identify user needs will risk success in product delivery. At the time of verification, identification of any factors that cause failure in identifying the needs of users. Among the factors mentioned by the respondents, among others, the developer was incomplete when exploring needs, problems with widespread needs, changes in organizational structure or the person responsible for the system, and changes in business processes.

Table 12. Survey result: Developers understand the risk of failure to identify needs.

Question 4: Understanding of respondents the risk of failure to identify needs				
Do you believe that failure to identify user needs risks the success of product delivery?				
Strongly Believe	Believe	Neutral	Disbelieve	Strongly Disbelieve
7	0	0	0	0
100%	0%	0%	0%	0%
Group 1:	Believe 100%+0%=100%, N1: 7+0=7			
Group 2:	Disbelieve 0%+0%=0%, N2: 0+0=0			

4.2.5. Implementation of increasing the competence of developers in the company

Workers increase competence in the organization where the respondent's work is divided into two groups. As shown in Table 13, Group 1 is an organization that supports increasing worker competence, where four organizations often improve the individual abilities of their workers, while Group 2 is two organizations that rarely improve the competence of their workers. The verification results from the interviews found that three organizations with the opinion that sometimes to rarely improve the competence of their workers place individual improvements to each worker, there is no plan from the organization. Whereas startup companies always improve according to current technological developments.

Table 13. Survey result: developer competence improvement.

Question 5: Management developer competence improvement				
In your organization, how many efforts have you made to improve employee competency?				
Always	Often	Sometimes	Rarely	Never
0	4	1	2	0
0%	57.14%	14%	28.57%	0%
Group 1:	Always 0%+57.14%=57.14%, N1: 0+4=4			
Group 2:	Never 28.57%+0%=0%, N2: 2+0=2			

4.2.6. Respondent experience combining user research activities in user experience with user requirement activities

Based on the respondent's experiences combining user experience activities and exploring user needs, the results were different compared to the results in Table 10. The results in Table 14 show that respondents combined several activities in software development, but not in the area of user experience and exploring software needs. Group 2 shows that the five respondents rarely combined the two activities. The results of verification using interviews stated that six respondents combined the activity of exploring software requirements with the user interface design phase, where the developer separated the user interface and user experience. In addition, respondents stated that combining the processes was confusing for developers to capture user requirements.

Table 14. Survey result: combining user experience and user requirement elicitation.

Question 6: developer experience combining user experience and user requirement elicitation				
Do you have experience combining user research activities in user experience with user requirement elicitation activities?				
Always	Often	Sometimes	Rarely	Never
0	1	1	5	0
0%	14.29%	14%	71.43%	0%
Group 1:	Always 0%+14.29%=14.29%, N1: 0+1=1			
Group 2:	Never 71.43%+0%=71.43%, N2: 5+0=5			

5. Discussion

Discussion to answer several questions as presented at the beginning of this study.

RQ 1: How to identify and structure user requirements in software development? User requirements cover the user's needs and expectations of the software. The characteristics of the user requirements in the software development following are easy to use and pass user requirements. Identifying and addressing problems that may arise during the development process is the main point to elicit user requirements. Moreover, developers should focus on software quality to deliver successful products on time.

RQ 2: How to evaluate and measure user experience in software development? The user experience should consist of four main activities. First is empathy, which is an activity that is used to understand, observe, and find user problems and more closely with design users. The second is problem-defining, activities in this category have been narrowed down to several design solutions. The developer will review some primary and secondary data to be classified based on proximity and possible problem solutions. The third is idea and visualization, visualization is a common activity in representing a design, including the user experience. The last is testing and iteration, this is an activity to explore user problems and test the potential solutions that have been obtained. Iterative activity to get a potential solution that is ready to be developed into a product. All design requirements and expectations from users have been fulfilled at this stage of the activity.

RQ 3: How to integrate user experience and user requirements in each stage of software development? In this study, we used a design thinking approach. Design thinking has been recognized as a broad approach to solving socially ambiguous design problems. Design thinking is the way designers think and apply their mental processes to design objects, services, or systems, which differ from the end result of elegant and usable products. The design has a broad cognition, more than specified in the field of software development, practically all aspects of various disciplines have design concepts. Design is defined as a conceptualization of process (action), creation (artifact, product, system, or service), planning, intention, etc.

RQ 4: Can the integration of user experience and user requirements improve software quality, reduce the risk of project failure, and speed up development time? The research conducted by Brown provides a broader perspective on the design thinking method, where in this research design thinking is used to view design perspectives from several different points of view. The first concept is design thinking to be an approach used to create new viable solution situations without leaving value from customers or improving solutions to meet customer needs with added value. The second concept uses design thinking to become an approach to design. This is a consensus among developers because thinking about design is an integral part of planning or design.

RQ 5: How to develop an effective and efficient requirements extraction method or technique to integrate user experience and user needs in software development? In developer productivity, integration of user experience and user requirements escalates developer productivity in developing software by focusing software development on features that meet the user's needs and expectations. In addition, developers reduce production time and effort by developing features that do not match the user's needs and are not required. That integration improves efficiency in software development by identifying and addressing problems that may arise during the development process. This improvement saves developer time and effort in developing software. To be able to understand the context of how developers can think about design, the concept of design itself needs to be brought to the fundamental aspects that can explain how each aspect can be interrelated and interact. Design activity is visible as a unified process aimed at producing design object specifications. Think critically about a design by considering the attention to several aspects. The first is the environment in which the objects of that design will exist. In this case, it will determine where the design will be used, in this aspect it can be proof that a design is something that has unique characteristics and environments. The second is the goal that is ascribed to the object in the design. The purpose of thinking about or organizing the design starts from the problems that exist in the user. The third is the desired structural and behavioral properties of the design object (requirements). The third aspect of this design relates to user requirements and the expectations that users embed in requirements. The fourth aspect is the collection of component types (primitives). The fifth aspect is the constraints that might limit acceptable solutions.

Author Contributions: Conceptualization, W.A.K., A.H.; Methodology, W.A.K., N.I.A, N.M.N; Validation, W.A.K., N.I.A, N.M.N; Formal Analysis, W.A.K., N.I.A, N.M.N; Investigation, N.I.A, N.M.N; Resources, W.A.K.; Data Curation, N.I.A, N.M.N; Writing-Original Draft, W.A.K., A.H.; Writing-Review Editing, W.A.K., A.H.; Supervision, A.H.; Project Administration, W.A.K.. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Ministry of Higher Education (Kementerian Pengajian Tinggi) and Research Management Centre (RMC), Universiti Putra Malaysia for supporting/funding this article under its Fundamental Research Grant Scheme (FRGS) – Project Code 08-01-20-2319FR – 5540451 and Universitas Muhammadiyah Malang for the APC support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study to voluntarily participate in the study and share their opinions.

Data Availability Statement: The study has made an available summary of the response data in Chapter 4.

Acknowledgments: We would like to take this opportunity to acknowledge and thank the Ministry of Higher Education (Kementerian Pengajian Tinggi) and Research Management Centre (RMC), Universiti Putra Malaysia for supporting/funding this article under its Fundamental Research Grant Scheme (FRGS) – Project Code 08-01-20-2319FR – 5540451. Universitas Muhammadiyah Malang for the funding support for the APC. We also like to thank the anonymous reviewers for their valuable feedback and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sommerville, I. *Software engineering*; Always learning; 10. ed., global ed.; Pearson: Boston Munich, 2016; ISBN 978-1-292-09613-1.
2. Pressman, R.S.; Maxim, B.R. *Software engineering: a practitioner's approach*; Ninth edition.; McGraw-Hill Education: New York, NY, 2020; ISBN 978-1-260-54800-6.
3. Anwar, R.; Rehman, M.; Wang, K.S.; Hashmani, M.A.; Shamim, A. Investigation of Knowledge Sharing Behavior in Global Software Development Organizations Using Social Cognitive Theory. *IEEE Access* **2019**, *7*, 71286–71298, doi:10.1109/ACCESS.2019.2912657.
4. Almeida, F.; Simões, J.; Lopes, S. Exploring the Benefits of Combining DevOps and Agile. *Future Internet* **2022**, *14*, doi:10.3390/fi14020063.
5. Caballero, L.; Moreno, A.M.; Seffah, A. How Agile Developers Integrate User-Centered Design into Their Processes: A Literature Review. *Int. J. Softw. Eng. Knowl. Eng.* **2016**, *26*, 1175–1201, doi:10.1142/S0218194016500418.
6. Chanin, R.; Pompermaier, L.; Sales, A.; Prikladnicki, R. Collaborative Practices for Software Requirements Gathering in Software Startups. In Proceedings of the 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE); IEEE: Montreal, QC, Canada, 2019; pp. 31–32.
7. Sukaviriya, N.; Sinha, V.; Ramachandra, T.; Mani, S.; Stolze, M. *User-centered design and business process modeling: Cross road in rapid prototyping tools*; 11th IFIP TC 13 International Conference on Human-Computer Interaction, INTERACT 2007; Springer Verlag: Rio de Janeiro, 2007; Vol. 4662 LNCS; ISBN 03029743 (ISSN); 9783540747949 (ISBN).
8. Nelson, N.; Brindescu, C.; McKee, S.; Sarma, A.; Dig, D. The life-cycle of merge conflicts: processes, barriers, and strategies. *Empir. Softw. Eng.* **2019**, *24*, 2863–2906, doi:10.1007/s10664-018-9674-x.
9. Schön, E.-M.; Thomaschewski, J.; Escalona, M.J. Agile Requirements Engineering: A Systematic Literature Review. *Comput. Stand. Interfaces* **2017**, *49*, 79–91, doi:10.1016/j.csi.2016.08.011.
10. Ahmadi Eftekhari, N.; Mani, S.; Bakhshi, J.; Mani, S. Project Manager Competencies for Dealing with Socio-Technical Complexity: A Grounded Theory Construction. *Systems* **2022**, *10*, 161, doi:10.3390/systems10050161.
11. Bourimi, M.; Barth, T.; Haake, J.M.; Ueberschär, B.; Kesdogan, D. *AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2010; Vol. 6409 LNCS, p. 189;.
12. Getto, G. Managing Experiences:: Utilizing User Experience Design (UX) as an Agile Methodology for Teaching Project Management. *Int. J. Sociotechnology Knowl. Dev.* **2015**, *7*, 1–14, doi:10.4018/IJSKD.2015100101.
13. Gräßler, I.; Pottebaum, J.; Oleff, C.; Preuß, D. HANDLING OF EXPLICIT UNCERTAINTY IN REQUIREMENTS CHANGE MANAGEMENT. *Proc. Des. Soc.* **2021**, *1*, 1687–1696, doi:10.1017/pds.2021.430.
14. Almeida, F.; Espinheira, E. Adoption of Large-Scale Scrum Practices through the Use of Management 3.0. *Informatics* **2022**, *9*, doi:10.3390/informatics9010020.
15. Abdeen, W.; Chen, X.; Unterkalmsteiner, M. An approach for performance requirements verification and test environments generation. *Requir. Eng.* **2022**, doi:10.1007/s00766-022-00379-3.
16. Yousaf, M.S.; Ali, S.; Nawaz, Q.; Afsar, S.; Mumtaz, I.; Rashid, N. Fagan Inspection: A Defects Finding Mechanism in Software Requirements Specification (SRS) Document. *VFAST Trans. Softw. Eng.* **2022**, 10.
17. Hsu, T.-C.; Abelson, H.; Patton, E.; Chen, S.-C.; Chang, H.-N. Self-efficacy and behavior patterns of learners using a real-time collaboration system developed for group programming. *Int. J. Comput.-Support. Collab. Learn.* **2021**, *16*, 559–582, doi:10.1007/s11412-021-09357-3.
18. A. Issaee; R. Motschnig; O. Comber Pair- versus solo-programming of mini-games as a setting for learning to program: An Action Research approach. In Proceedings of the 2021 IEEE Frontiers in Education Conference (FIE); 2021; pp. 1–9.
19. Dorst, D.; Stewart, S.; Staudinger, I.; Paton, B.; Dong, D. Introduction. *DTRS8 Proc. 8th Des. Think. Res. Symp.* **2010**.
20. Lindberg, T.; Gumienny, R.; Jobst, B.; Meinel, C. Is there a need for a design thinking process? *Des. Think. Res. Symp.* **2010**, 243–254.
21. Cross, N.; Dorst, K.; Roozenburg, N. Research in Design Thinking. *Res. Des. Think.* **1992**.
22. Cross, N. *Des. Think. Underst. Des. Think Work* **2011**.
23. Dunne, D.; Martin, R. Design thinking and how it will change management education: An interview and discussion. *Acad. Manag. Learn. Educ.* **2006**, *5*, 512–523, doi:10.5465/AMLE.2006.23473212.
24. IEEE standard glossary of software engineering terminology. *IEEE Stand. Gloss. Softw. Eng. Terminol.* **1990**.

25. Brown, A.W. Personal software engineering project management process. *Proc Int Conf Softw. Eng* **1999**, 669–670, doi:10.1145/302405.302928.
26. Adikari, S.; McDonald, C.; Campbell, J. Reframed Contexts: Design Thinking for Agile User Experience Design. In *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools*; Marcus, A., Ed.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013; Vol. 8012, pp. 3–12 ISBN 978-3-642-39228-3.
27. Eris, O. Insisting on truth at the expense of conceptualization: Can engineering portfolios help? *Int. J. Eng. Educ.* **2006**, *22*, 551–559.
28. Eris, Ö. ASKING GENERATIVE DESIGN QUESTIONS: A FUNDAMENTAL COGNITIVE MECHANISM IN DESIGN THINKING.
29. Carlgren, L. Identifying latent needs: towards a competence perspective on attractive quality creation. *Total Qual. Manag. Bus. Excell.* **2013**, *24*, 1347–1363, doi:10.1080/14783363.2013.776762.
30. Carlgren, L.; Elmquist, M.; Rauth, I. Design Thinking: Exploring Values and Effects from an Innovation Capability Perspective. *Des. J.* **2014**, *17*, 403–423, doi:10.2752/175630614X13982745783000.
31. Carlgren, L.; Rauth, I.; Elmquist, M. Framing Design Thinking: The Concept in Idea and Enactment: Creativity and Innovation Management. *Creat. Innov. Manag.* **2016**, *25*, 38–57, doi:10.1111/caim.12153.
32. Carlgren, L.; Elmquist, M.; Rauth, I. The Challenges of Using Design Thinking in Industry - Experiences from Five Large Firms: THE CHALLENGES OF USING DT IN INDUSTRY CREATIVITY AND INNOVATION MANAGEMENT. *Creat. Innov. Manag.* **2016**, *25*, 344–362, doi:10.1111/caim.12176.
33. Beckman, S.L.; Barry, M. Innovation as a Learning Process: Embedding Design Thinking. *Calif. Manage. Rev.* **2007**, *50*, 25–56, doi:10.2307/41166415.
34. Liedtka, J. Perspective: Linking Design Thinking with Innovation Outcomes through Cognitive Bias Reduction: Design Thinking. *J. Prod. Innov. Manag.* **2015**, *32*, 925–938, doi:10.1111/jpim.12163.
35. Grönman, S.; Lindfors, E. The Process Models of Design Thinking. **2021**.
36. McDonagh, D.; Denton, H.; Chapman, J. Design and emotion. *J. Eng. Des.* **2009**, *20*, 433–435, doi:10.1080/09544820902915300.
37. Alsanoosy, T.; Spichkova, M.; Harland, J. A framework for identifying cultural influences on requirements engineering activities.; 2020.
38. Kouprie, M.; Visser, F.S. A framework for empathy in design: Stepping into and out of the user's life. *J. Eng. Des.* **2009**, *20*, 437–448, doi:10.1080/09544820902875033.
39. Badke Schaub, P.G.; Roozenburg, N.F.M.; Cardoso, C. Design thinking: A paradigm on its way from dilution to meaninglessness? *Proc. 8th Des. Think. Res. Symp.* **2010**.
40. Schön, D.A. The reflective practitioner: How professionals think in action. *Basic Books* **1983**.
41. Drljaca, D.P.; Latinovic, B. Using TELOS for the planning of the information system audit. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *294*, 012022, doi:10.1088/1757-899X/294/1/012022.
42. CC2020 Task Force *Computing Curricula 2020: Paradigms for Global Computing Education*; ACM: New York, NY, USA, 2020; ISBN 978-1-4503-9059-0.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.