E-ISSN 2224-2333
P-ISSN 2222-9930
Pakistan Journal of Engineering Technology and Science (PJETS)
Volume 13. Issue.01, PP. 15-30, JUNE 2025
https://doi.org/10.22555/pjets.v13i1.1179

Enhancing User-Centered Feature Development: Integrating Design Thinking into Feature-Driven Development

Anum Khalid [1*], Safa Tariq[1], Rohina Akram[1], Dr.Maryam Kausar[1]
[1]Foundation University Islamabad, Pakistan
[*]Corresponding author: anumkhalid3003@gmail.com

Abstract:

The popularity of agile is rising as technology continues to advance. The integration of design thinking into agile methodologies has gained significant attention in producing user-centric products. Feature-driven development incorporates market demands in a very short period and accomplishes outcomes and interpretations from each iteration. Whereas design thinking is a human-centered approach that keeps the user at the core of the decision. This research aims to propose a framework that establishes a bridge between feature-driven development and design thinking, as user needs are not prioritized during feature-driven development due to the main emphasis on feature completion, and to enhance user-centered feature delivery in agile environments. This research investigates the 40 respondents' opinions, consisting of low-experienced and highly experienced practitioners, drawing on a mixed-methods research design that includes surveys with agile teams. We explore the proposed framework's effects on user specifications, speculation, differentiation, data dissemination, emotional engagement, level of fidelity, user-centric experience, co-creation, realism, and user contentment. The analysis revealed that user satisfaction is influenced by experience levels, with years of experience showing a significant main effect ($p = 0.024$). Low-experienced individuals had lower satisfaction scores compared to high-experienced individuals. These findings suggest that experience duration plays a key role in shaping user satisfaction. A mean difference of -0.170 indicates lower user satisfaction among less experienced individuals, suggesting that satisfaction varies with experience duration. These findings suggest that variations in user satisfaction are influenced by the duration of experience.

Keywords: *Agile Methodology, Design Thinking, Feature-Driven Development, Innovation, Iterative Process, User-Centered Design.*

I. INTRODUCTION

Agile gained popularity due to its ability to facilitate cultural change in administrative decision-making and adaptation to the changing environment, user values, and needs within software development [1]. Agile is centrally concerned with providing value to customers and stakeholders by giving priority to their needs and acting swiftly in response to shifting priorities and requirements [1] [2]. Agile software project management is a specific application of Agile for the management of software development projects. It involves using Agile principles and practices to guide the planning, execution, and delivery of software projects [3]. Agile methodology in software development is characterized by its flexibility, collaboration with team members, iterative development, customer focus, and emphasis on continuous improvement. It allows teams to quickly respond to changing requirements, promotes communication, involves iterative development for faster feedback, focuses on delivering value to customers, and encourages learning from past successes and failures to identify areas for improvement [1] [2].

Feature-driven development (FDD) is gaining popularity due to its focus on the iterative delivery of features and emphasis on design that makes it suitable for large and complex software projects as well as with large and distributed teams [4][5]. FDD in Agile emphasizes status reporting that helps to monitor progress and results, ultimately delivering tangible and quantifiable results [6]. The DT approach is to improve the software's quality and usability by using processes that link customer demands to the good or service being developed [7] [8]. According to an article on Hotjar, DT and agile are user-centric methodologies that rely on user feedback to guide the development of a product. Likewise, another evolving concept of UX (User Experience) is to describe the whole interaction and experience a user has when using a product or service; it includes the user's perceptions, feelings, and reactions during the process. Usability testing, user research, and user persona creation are examples of user experience activities that can be incorporated into agile sprints to help teams stay focused on users and validate presumptions. Together, user experience and DT in agile software project management make sure that the product being developed not only meets the demands of the user but is also efficient and enjoyable to use [9]. DT is especially crucial in Agile software project management because it enables teams to collect useful user feedback early in the development process, which can help to identify and resolve potential issues before they become significant roadblocks [10]. Teams can work more cooperatively and productively by incorporating DT into Agile because they can rapidly test and iterate on potential solutions rather than spending weeks or months developing a single feature [11]. Agile software project management includes DT because it serves to ensure that the finished product is both useful and valuable to the customer [12] [13].

The main goal of the empathize stage of the design process is to understand the requirements, driving forces, and behavioral patterns of the target audience. To learn more about users' experiences and difficulties entails conducting research, observing users in their natural environments, and having sympathetic discussions. The define phase entails combining the insights obtained from the empathize phase to identify the primary issue or opportunity that must be addressed. To do this, the problem statement must be reframed in light of the knowledge acquired from user involvement. The objective of the ideation phase is to produce a broad spectrum of innovative concepts and practical solutions to deal with the identified issue or opportunity. During this brainstorming session, participants are urged to consider unique ideas and think outside the box without passing judgment. The next step is to develop low-fidelity prototypes or illustrations to represent the concepts that were found to be promising during the brainstorming process. Prototypes are the actual models or interactive simulations, and mock-ups. Prototyping serves the function of rapidly and affordably testing concepts, getting user feedback, and iterating designs before devoting substantial resources to development. Prototypes are shown to users during the test phase to get their input and assess how well they solve the identified opportunity or problem. To find out how people engage with the prototypes and pinpoint areas for improvement, this entails holding user testing sessions, observations, and interviews. Prototypes are iteratively improved in response to input until a workable solution is produced. However, there is a relatively inadequate done thorough study on DT that may be applied in the implementation of a particular methodology like FDD. In this area of study, more investigation is necessary [14]. Therefore, the purpose of our study is to establish a bridge that will fill the gap between FDD and the adoption of DT. Understanding user needs and preferences is emphasized by DT. The development process is more intently focused on producing features that directly meet the requirements and wants of users when it is integrated with FDD.

The attempt to integrate DT into FDD arises from DT's encouragement of distinctive concepts and unconventional approaches. It can encourage the creation of distinctive features that distinguish the product from rivals when paired with FDD. In the early phases of development, DT frequently incorporates user testing and prototyping. Teams can lower the risk of developing features that users find useful or valuable by including this method in FDD and validating feature concepts early on. Since DT strongly emphasizes stakeholder involvement from the beginning of the design process and cross-disciplinary collaboration, integrating DT with FDD improves collaboration. DT encourages flexibility and feedback-based iteration. By integrating this mindset into FDD, teams can respond more effectively to changes in user requirements or market situations. By integrating DT into FDD, failure risk is decreased and problem-solving abilities are enhanced. All things considered, DT integration with FDD can result in a more creative, cooperative, and user-focused development process, which will eventually produce products that satisfy customer needs and accomplish organizational goals. It is anticipated that the incorporation of DT into FDD will give users priority during the FDD process. Our focus is on the integration of DT into FDD, although incorporating DT into Scrum or other Agile techniques offers many advantages as well, including collaborative development, flexibility, and iterative development. Out of the many studies that have examined DT's integration with Scrum and other techniques, this specific integration has garnered little to no attention in the literature. So our investigation into this brings us to answer the following questions.

Research Question 1: How may DT practices be included in the agile FDD process?

Research Question 2: In the context of integrating DT into FDD, what is the validity and reliability of the survey instrument used to gauge user happiness, taking participants' years of experience into account?

Research Question 3: What impact does integrate DT into FDD have on user satisfaction when considering the level of experience of individuals involved?

This study is structured as follows: Initially, concepts related to agile, FDD, and DT were discussed, then a review of associated work in these areas. Following the outline of the study approach, we obtained data from publications such as books, reports, academic papers, and other published materials to shed light on issues. This process is also known as secondary research. Then, we present the theoretical framework of FDD with the integration of DT principles. Subsequently, a mathematical model was used to validate the results. Finally, we explain the study's limitations and point up potential directions for future research.

II. LITERATURE REVIEW

FDD adapts to changing requirements, timebox focusing, and a feature-centric approach to organizing and delivering software functionality. A feature is a plannable functionality derived from a planning perspective and related to the standard Unified Process. FDD's principal slogan is 'Feature Per Fortnight' (Carmichael and Haywood, 2002). The rule implies a deliverable feature to the customer after every two weeks. FDD is applied in the iterative software process development that does not try to complete the whole system in a single attempt [15]. In addition, FDD is utilized in large-scale projects due to its scalable nature. According to [16], the FDD process follows five core activities. Initially, an overall model is developed that includes defining the scope and boundaries of the system. Next, a feature list is built to break down the model into smaller features, each of which occupies a specific period. Then, the planning and scheduling of features take place. Furthermore, each feature is designed to determine the data structure to use and create algorithms and user interfaces. Finally, a feature is built and tested before it integrates with another feature or system.

DT, which is praised for facilitating a thorough grasp of user demands and fostering team cooperation, has been shown by different research to have a substantial impact on software development. With a more user-centered approach to invention, the DT idea claims to increase inventiveness [17]. Furthermore, we observed that design thinking might be used to improve processes, for instance, by doing retrospectives or a parallel design thinking phase.DT not only facilitates particular tasks but can additionally encourage Scrum and agile development principles. The gap in the paper was that the majority of developers were utilizing well-known single DT techniques. Therefore, it would appear that planning training initiatives and resources to increase developers' understanding of the DT approach would be advantageous and worthwhile. The fundamental tenet of DT is that goods, processes, and services are complicated and that consumer perceptions regarding them are highly changeable. With this goal in mind, DT claims that by empathizing with users, trying out ideas, prototyping them, and receiving quick feedback, designers are more likely to be successful in satisfying user needs and optimizing and humanizing solutions [18].

Although a large portion of the material currently available on design thinking concentrates on its applications in user experience (UX) and product design, there aren't many thorough frameworks made especially for incorporating design thinking concepts into content strategy. Quan Zhou [19] compared content strategy and DT. It details two techniques—user journey mapping and storyboarding—that effectively use the benefits of DT to strengthen content strategy. DT provides disciplined and practical strategies to translate abstract empathy into workable solutions. When used in content strategy, DT not only roughly resembles content strategy but also enhances it with rich insights and in-depth findings.

In the paper by Lauriane Pereira and Rafael Parizi [20], DT assists developers by helping them to comprehend the actual needs of users. To address user demands and accomplish organizational objectives, DT strives to bring together a collection of activities influenced by design for product development. These practices use empathy, imagination, and reason. Some of the challenges faced by the developers are Changes in requirements, a lack of comprehension of the scope of the system, and no linkage between the requirements, code, and documentation to overcome these challenges DT has been used to gather requirements and specify them more effectively, considering and constructing suitable answers to help them. One potential research gap is the lack of studies on its scalability and long-term effects when used in large-scale, intricate requirements engineering projects. Most studies concentrate on short-term results, frequently in isolated or smaller settings, and fail to consider how Design Thinking may be scaled or modified in bigger, multi-team, multi-stakeholder situations where the dynamics and difficulties are very different.

Another research article has discussed the implementation of IBM DT procedures and how it affected the creation and customer acceptance of IBM z14. The majority of current research focuses on initiatives for digital transformation or new product development, but little attention is paid to how these contemporary approaches can be scaled and modified for legacy system evolution, which entails more complicated limitations and difficulties like preserving backward compatibility, utilizing antiquated technologies, and integrating with current infrastructure. In this paper, the user experiences outlined there supported each of the four Pillars of the z14 program (pervasive encryption, cognitive, simplification, and infrastructure). The paper has succeeded in generating meaningful results that are centered on the needs of users by embracing the IDT and Agile mindset and behavioral model of constant 'observe, analyze, and develop' [21].

Agile and DT concepts can improve the effectiveness of the idea exploration and definition activities as well as the requirements evaluation and creation processes. DT's emphasis on prototyping a variety of possibilities and Agile's emphasis on incremental development and attaining a working product as soon as possible can eliminate risks and uncertainty earlier in the process of creation. M. Ann Garrison Darrin and William S. Devereux [22] proved by some potential advantages and disadvantages that these novel methodologies can be modified and included in a more agile systems engineering process instead of being in contradiction to conventional systems engineering approaches. One weakness in the paper is the absence of a cohesive, all-encompassing framework that combines these three methods (Agile, Design Thinking, and Systems Engineering) for overseeing the evolution of complex systems. Jennifer Hehn and Falk Uebernickel [23] concluded that with the help of DT, techniques that are frequently applied in requirements elicitation can be applied practically. Additionally, it encourages innovation by continually redefining the problem and solution domain to determine the ideal answer for the end user. According to them, DT functions as a helping hand for RE to seize difficult challenges, and RE provides a solid inclusion framework for DT within the software development life cycle. They provided an overview of how DT supports RE by addressing issues that practitioners in the RE field are already aware of.

Peter Newman and Maria Angela Ferrario [24] defend the importance of DT in social software engineering and outline the consequences for the field generally. Little research has been done on the usefulness of physical prototyping, which is usually connected to hardware development and product design, in the context of social software engineering. This study provided DivingBoard, a DT element that encourages a potentially unrestricted exploration of the problem area. In OnSupply, they discussed the iterative process through which DivingBoard develops well-contextualized domain-dependent artifacts and concepts following the Agile manifesto.

Mali Senapathi and Meghann L. Drury-Grogan [25] discussed how DT was used to explore and validate the issue and solution domains to integrate Kanban practices, including visualization, control flow, categories of service, and board design. The business unit used STATIK to deploy Kanban in an iterative process where responses from one stage were utilized to guide and affect the others cooperatively. Enhancements in time to completion, teamwork, and visibility were all positive experiences. Lack of team acceptance, limiting ongoing work, and a temporary Kanban coach were among the findings. The proposed model was benchmarked against existing approaches to demonstrate its superiority, as shown in the Table. 1.

Table 1. Benchmarking Results: DT@FDD vs. State-of-the-Art Models

Parameters	**DT@Scrum [7]**	**DT@XP [8]**	**DT@FDD (Proposed model)**
Objective	Focuses on Requirements engineering management. Refine backlog through clarity, prioritization, and align stakeholders.	Focuses on technical practices. Enhanced usability through user feedback to produce iterative prototypes.	Focus on improving overall structure to produce innovative products.
Technique or approach used	Integrate DT practices into enhancing backlog and sprin planning.	Integrate DT practices into XP's planning, design, coding, and testing to improve usability.	Integrating DT practices into FDD's structure approach provides a roadmap from planning to feature development.
Collaboration	Emphasis on close collaboration among Product Owners, Scrum Masters, and Developers with stakeholders for requirement gathering and validation. Support cross-functional team collaboration.	Emphasis on close collaboration among developers, designers, and customers, and in pair programming (collective code ownership).	Emphasis on close collaboration across teams throughout feature development to address user needs and insights from the DT practices, ensuring each feature adds value to the user experience.
User-centric Experience	After each sprint, user feedback enhances product relevance.	Frequent releases facilitate quick changes based on user feedback.	Deliver features that align with user expectations.

Quality Enhanced	Scrum may exceed time and result in high-quality products.	TDD, pair programming, and refactoring ensure quality is maintained.	User-centric feature delivery and refinement.
Risk Management	After each sprint, the incremental testing and update backlog.	Frequent releases help in risk mitigation.	Feature validation helps in managing risk.
Flexibility	Scrum allows for backlog re-prioritization after each sprint.	Adaptability during development due to frequent releases and refactoring.	Feature prioritization and user feedback after feature delivery.

III. METHODOLOGY

This paper was prepared using a systematic literature review (SLR), following Kitchenham and Charter's Guidelines [26]. The purpose of this research is to address the questions that follow:
Research Question 1: How may DT practices be included in the agile FDD process?
Research Question 2: In the context of integrating DT into FDD, what is the validity and reliability of the survey instrument used to gauge user happiness, taking participants' years of experience into account?
Research Question 3: What impact does integrate DT into FDD have on user satisfaction when considering the level of experience of individuals involved?

A. *Factors Affecting the Research Question:*

The main factor affecting the research question is the Integration of DT. This independent variable indicates how much the practices and principles of DT are integrated into the FDD process. User Satisfaction these dependent variable gauges how content users or stakeholders are with the finished item or result of the combined DT and FDD approach and Experience, the amount of experience or expertise of those involved in the development process is represented by the control variable, which may have an impact on how well they can incorporate DT and FDD.

B. *Data Source and Search Strategies:*

Initially, a detailed review of the literature was conducted to integrate DT into FDD. Every paper that was found online was written in English. Five widely recognized e-databases were employed for the review: Google Scholar, ACM Digital Library, Springer Link, Elsevier, ScienceDirect, and IEEE Xplore. The paper review procedure and the number of papers identified at each stage are depicted in Fig. 1.

The databases were chosen in the first step using the search criteria listed in Table 2. There are variants of the terms "Agile" and "design thinking" under Category 1 and Category 2, respectively. Agile and design thinking are two keywords that must be in an article for it to be considered for inclusion in this research. This is because all search items were combined using the Boolean "AND" operator. Consequently, every possible combination from both Categories 1 and 2 was looked up.

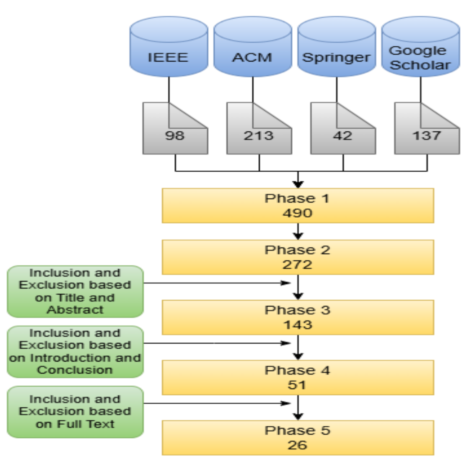

Figure 1. Primary Paper Selection Process

Table 2. Search Terms Used for This Review

Type	Category	Keywords
1	Agile	Agile software development, Agile Methodologies, Agile Practices, Feature-driven development
2	Use of design thinking	Design thinking practices

The papers that fulfilled the search requirements in terms of titles, abstracts, or keywords were selected. Conferences and peer-reviewed journals were also selected as trustworthy sources of the required information. This search strategy resulted in 1,374,059 'hits' in all. However, after accounting for duplicate papers, there were only 4,677 hits left. Subsequently, the second stage comprised removing articles based on how well the titles and keywords matched the search criteria. 272 articles were selected in total after reading all the titles. After reading every abstract, we only included works that had a clear relationship to DT. A total of 143 articles were selected for the third phase. About 51 articles were selected for phase 4. Lastly, 26 articles that focused on DT were selected.

C. *Instrument for Data Collection:*

Thirdly, an instrument for data collection was a questionnaire with the target population being Agile experts and software development teams that used FDD or had experience in product development. The primary goal of the study was to find out how people view the use of DT@FDD and how useful they think it is. To accomplish this, a survey questionnaire with several questions was used. The survey has been carried out through the Google Forms platform https://forms.gle/MbmNtqbaVqAa4PJ9A. Email invitations, social media, and internet forums were used to find participants. Questions concerning software quality experience and opinions about DT@FDD were all included in the survey form. To obtain information about software quality and tool integration, responses were gathered securely and subjected to both quantitative and qualitative analysis. The survey yielded expert perspectives on the integration of DT with FDD.

To quantify the survey, a centered approach Likert scale was used. A 5-point Likert scale was employed to gauge respondents' attitudes or perceptions regarding different issues. By giving respondents a variety of options, this scale enabled them to express their thoughts with differing degrees of agreement or disagreement. This scale is straightforward for respondents to understand and use efficiently, while it offers enough granularity to capture variations in responses. The scaling options were Strongly Disagree (-2), Disagree (-1), Neutral (0), Agree (1), and Strongly Agree (2). The type of sample was probability sampling. The outcome of the variable factors will be analyzed by one-way ANOVA, followed by a univariate analysis in IBM SPSS Statistics 21. Each technique fulfilled a distinct function: Levene's test made sure the assumption of homogeneity of variances was met, ANOVA evaluated whether there were statistically significant differences between group means, descriptive statistics gave an overview of the data distribution, and pairwise comparisons with Bonferroni correction helped pinpoint group differences while accounting for type I error rate.

Table 3. Details of Employees

Category	Experience in FDD	Experience in Agile	No. of employees
Low Experience	1-4	< 3	23
High Experience	5-14	4-10	17

Table 3 presents information about the employees who participated in the survey, grouped according to their level of experience with Agile and FDD. The workers are separated into two primary groups based on their experience levels: Low Experience and High Experience. There are 23 personnel in the Low expertise group that have less than 3 years of Agile expertise and between 1 and 4 years of FDD experience. Conversely, the High Experience group consists of 17 workers who have between 5 and 14 years of FDD and 4 and 10 years of Agile experience.

Table. 4., provides information on each company from which individuals with both low and high levels of experience were chosen. Four companies were invited to the survey. Based on our previous categorization, there was one profitable company, two break-even companies, and one startup. Every participating company has at least two years of FDD experience.

Table 4. Invited Company Details

Company	Category	Type of business	HQ	Location of business	Experience in FDD
C1	Startup	Product-based	Pakistan	Dubai, UK	2
C2	Break-even	IT services	Pakistan	Dubai	5
C3	Break-even	Multinational IT service provider	Pakistan	KSA, Egypt	10
C4	Profitable	Software Solutions	United States	Pakistan	14

IV. PROPOSED FRAMEWORK

RQ1: How may DT practices be included in the agile FDD process?
To answer RQ1, a framework is proposed that combines DT with FDD to complement software development teams' capacity for problem-solving because FDD does not take DT practices into account with the intent of producing their product more ingeniously and user-friendly. Integrating the DT concepts with the agile frameworks delivers more importance to the users and can increase output in providing deliverables [27]. In the proposed framework, the five core activities of DT are integrated into the FDD activities. The adapted approach of DT@FDD is shown in Fig. 2.

The initial phase of FDD is to design a structural model. The outcome of FDD is to create a scope or high-level concept abstraction without prior user research. However, a user-centered design approach is not implied. FDD does not emphasize putting user needs in the front and center. The issue is a lack of awareness of the user's journey and a sprint into production without any user research. Therefore, the Empathy phase of DT will acquire the user's pain points and thoughts with the assistance of conducting interviews. The Define phase will analyze and synthesize the data gathered from the interview. It will produce the core problem or the scope of the system. To keep the efforts human-centered, personas and user stories will be created to ensure the user research and to create a domain object model.

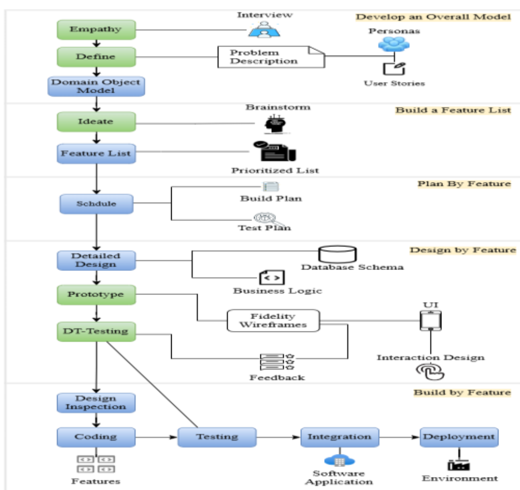

Figure 2. DT integration into FDD

FDD's second phase includes building a feature list. This phase generates features in the form of requirements or feature lists and prioritizes them. The team was focused on a single concept, but other solutions may have been constructed, allowing for the elicitation and verification of the requirements through various types of solutions. To take exception, the efforts of generating feature lists without ideation are meritless. Therefore, the essence of the Ideate phase will stimulate the minds of team members to think divergently and come up with contemporary ideas. The team members will perform the technique of problem-solving activity known as brainstorming. By brainstorming, advanced ideas may shape into a feature list that may subsequently be prioritized by the team leader.

The third phase of FDD includes planning by feature. It includes allocating resources such as time, cost, and quality in an optimized manner that results in the scheduling of tasks with the help of plans such as build and test plans.

FDD's fourth phase involves the implementation of a detailed design incorporating a database schema and business logic. Developers were only able to produce a basic prototype that lacked accessibility, equity, and usability due to the absence of a user experience designer or UX team. However, the integration of the DT phase, such as the Prototype, will be enacted before designing the user interfaces. The prototype phase will produce low or high-fidelity wireframes of the features, depending on the team's selection criteria for low or high-fidelity wireframes. These fidelity wireframes outline the blueprints of the features that are evaluated by the user through testing to gather feedback or insights before implementing the business logic, database schema, and user interfaces.

The last phase of FDD incorporates inspecting the design before coding initiates, testing, integration, and deployment to the production environment. Because test case scenarios for the testing phase do not contain actual user responses, intercept surveys are not used to obtain on-site feedback from a target audience. Therefore, the DT prototype phase will lead to the testing phase that supports FDD testing activity. Test case scenarios will focus more on human-centered design, and software products will be less prone to user error or usability error.

Suppose the revamping of an e-commerce application (version 1.0) required the integration of DT into the existing FDD methodology. The initial application, while efficiently developed and delivered within a short timeframe, focused exclusively on core features such as user registration and login, product catalog, shopping cart, payment integration, wish list, ratings and reviews, and user profiles. However, this approach failed to yield profitability or client satisfaction, as the application lacked differentiation from competitors. This paper outlines the process of combining DT and FDD to address these shortcomings, focusing on user-centric development and innovative feature integration. The initial version of the e-commerce application was developed using the FDD approach, emphasizing the rapid implementation of essential functionalities. Despite the expedited delivery of features, such as user registration, product catalog, and payment integration, the application failed to meet client expectations or achieve a competitive advantage. It lacked unique features, resulting in dissatisfaction among clients and a lack of market differentiation. To address these challenges, the team integrated DT into the FDD methodology, aiming to prioritize user-centric design and innovation.

- Empathize Phase

The redesign process commenced with the empathize phase, wherein the UX designer conducted user interviews and research to identify critical pain points. Key issues included:

  - Poor navigation and inadequate search functionality.
  - Limited accessibility for non-English-speaking users.
  - Inability to make purchases in multiple currencies.

Based on this research, the UX designer developed detailed personas representing target user groups, such as last-minute shoppers, fashion enthusiasts, and frequent returners. This phase clarified the problems, including difficulties in navigating the product catalog and the absence of multi-language and multi-currency support. The domain object model was then expanded to include key entities such as users, search functionality, multi-language, and multi-currency features.

- Ideation and Build-by-Feature Phase

  In the subsequent build-by-feature phase, a cross-functional team brainstormed solutions to address the identified problems. Proposed features included:

  - Artificial Intelligence (AI) integration to recommend products based on purchase history.
  - Augmented Reality (AR) to enable virtual product trials.
  - Voice search and chatbots to enhance accessibility and assist with frequently asked questions (FAQs).

The product manager prioritized the proposed features based on timelines, resource constraints, and project milestones, resulting in a comprehensive development and testing plan.

- Design-by-Feature Phase

  During the design-by-feature phase, a detailed database schema was created to support the enhanced application architecture. Low-fidelity and high-fidelity wireframes, along with interaction designs, were developed to visualize the app's user interface and experience. The DT methodology enabled iterative testing and refinement, allowing users to provide feedback on prototypes. Wireframes were updated based on this feedback until the desired outcomes were achieved.

- Implementation and Testing Phase

  In the implementation phase, code was developed for each feature, accompanied by rigorous testing procedures, including unit testing, integration testing, and user acceptance testing. Insights from the DT testing phase were incorporated to ensure the application met user expectations. Following successful testing, the backend, frontend, and database components were integrated, leading to the final deployment of the re-vamped application.

Integrating DT into the FDD process enhanced the e-commerce application's ability to address user pain points and differentiate itself in a competitive market. By prioritizing user-centric design and iterative development, the team successfully delivered an application tailored to the needs and expectations of its target audience. This case study demonstrates the efficacy of combining DT and FDD methodologies for creating innovative, user-focused digital solutions.

User satisfaction was the main component, and sub-factors that contributed to the total level of satisfaction included a thorough analysis of pre-existing data sources, including articles, and we were able to understand how these sub-factors combined to affect users' overall satisfaction levels. With the help of this secondary data analysis technique, we were able to examine a variety of user interactions and feedback to identify the subtle elements influencing user satisfaction. User specifications, speculation, differentiation, data dissemination, emotional engagement, level of fidelity, user-centric experience, co-creation, realism, and user contentment. With the sub-factors standing in for components, the research evaluated user satisfaction as a composite measure, as indicated by the hierarchical structure. The formulated 10 factors were drawn from a distinct DT component or idea. Furthermore, if there are too many factors, the analysis may become unduly complex, and if there are too few factors, significant DT components may be missing. Table. 5, contains the selection of these ten elements attempted to achieve a balance in the study between comprehensiveness and manageability.

Table 5. Factors Description

S.No	Factors	Description
SF1	**User Specifications**	In FDD, conducting user interviews is crucial to obtain insightful feedback from the users' point of view when developing the overall model. The team can better grasp users' wants and translate them into actionable problem statements owing to these insights.
SF2	**Speculation**	The team frequently imposes limitations and self-assumptions while describing an issue. User stories and persona-based scenarios should be used to guarantee openness and user alignment. By outlining user interactions and high-level business requirements that the system must meet, these tools help users and the team stay consistent.
SF3	**Differentiation**	Teams frequently focus on identifying requirements instead of considering creative ideas when generating a feature list. To solve this, it is essential to halt and conduct brainstorming sessions to uncover upcoming technologies and trends. Before going into specific needs, incorporating these concepts early on helps set the product apart from competitors and creates new opportunities.
SF4	**Data Dissemination**	Written documentation and feature list demonstrations are usually the main forms of communication within a team during the planning stage of feature creation. Through iterative feedback loops, the team must actively engage and incorporate cross-functional team members throughout the process to handle this difficulty.
SF5	**Emotional Engagement**	Teams frequently build prototypes during feature design only based on the feature list without user validation. The team should use user feedback to evaluate the prototype and confirm the feature list to close this gap. The team can make sure that the prototype elicits the intended emotional involvement and successfully satisfies user expectations and needs by involving users in the validation process.
SF6	**Level of Fidelity**	Teams frequently place great value on precise, high-fidelity designs that closely resemble the finished product while creating prototypes. On the other hand, this may result in resource limitations and challenges when implementing modifications. To overcome this, teams ought to get early user feedback using low-fidelity sketches as well. Teams can uncover possible problems, save time, and prevent long-term expenses by validating the design with users.
SF7	**User-Centric Experience**	Incorporating user feedback through prototyping, the team designs features and uses the feature list to discover usability issues. Producing tangible items that enable stakeholders to engage and visualize concepts is essential for improving the process. The whole user experience is eventually enhanced by this method.
SF8	**Co-Creation**	The user is frequently not involved as much in the feature design process. To solve this, it's critical to promote an environment of open communication

		between users, developers, and designers, as well as a collaborative approach to product development.
SF9	**Realism**	App designers frequently place less emphasis on visual aesthetics while creating user interfaces. To make an interaction design memorable and engaging, it is advised to concentrate on using animations to create an emotional connection with consumers.
SF10	**User Contentment**	The team usually prioritizes checking that the requirements listed in the report are met while testing a feature. However, this process may be made much easier by taking into account user feedback from the prototype phase, which helps create test-based scenarios that match user preferences and demands.

RQ2: In the context of integrating DT into FDD, what is the validity and reliability of the survey instrument used to gauge user satisfaction, taking participants' years of experience into account?

To gather quantifiable information about the efficiency of combining DT@FDD, a survey was carried out. The survey from the specific organization was selected for their willingness, as they had been employing agile practices for more than 3 to 4 years. The survey follows a primary data collection method. A series of predetermined, close-ended questions was used in surveys to collect structured data on the use of DT@FDD from participants. Statistical analysis model technique ANOVA will be used to examine the quantitative data gathered from the survey to assess the success of incorporating DT into the FDD process. All aspects of the study took ethical considerations into account. All participants provided informed consent, and confidentiality and anonymity were upheld throughout the research.

Table 6 provides numerous important insights into the respondents' perceptions of the various elements that the research looked at. Firstly, the responses for each element range from -1.00 to +2.00, demonstrating the respondent's ability to select Disagree (-1), Neutral (0), Agree (1), and Strongly Agree (2) and discarding the Strongly Disagree (-2). Second, mean scores reveal the average degree of agreement or perception concerning each factor. Factors such as user specifications, speculation, and user contentment yield mean scores above 1.00, signifying generally positive perceptions, while data dissemination and emotional engagement garner mean scores below 1.00, suggesting less favorable perceptions. Thirdly, the difference between each factor's highest and minimum values is used to identify response variability. Some factors, such as level of fidelity, show a smaller range, indicating less variability, while others, such as user specifications and user contentment, exhibit a larger range, suggesting greater variability. Furthermore, sum scores shed light on respondents' overall agreement or favorable opinion of each aspect, with greater sums assigned to criteria like user satisfaction and specifications. Finally, the sample size is reflected in the valid N (listwise) of 40, which represents the total number of valid cases examined for analysis.

Furthermore, SF2, SF7, and SF10 have significantly lower levels of response, SF1, SF3, SF4, SF6, SF8, and SF10 show a moderate degree of response. SF5 and SF9, on the other hand, exhibit comparatively higher levels of reaction. Most of the components' standard deviations show a moderate amount of variation in responses from the mean, pointing to some participant variability. The higher standard deviation of SF5 indicates that responses are more variable.

Table 6. Descriptive Statistics

Factors	Min	Max	Mean	Sum	Std. Deviation	Variance	Skewness		Kurtosis	
	Statistics	Statistics	Statistics	Statistics	Statistics	Statistics	Statistics	Std. Error	Statistics	Std. Error
SF1	-1.00	2.00	1.1250	45.00	.72280	.522	-.623	.374	.638	.733
SF2	-1.00	2.00	1.0750	43.00	.61550	.379	-.734	.374	2.673	.733
SF3	-1.00	2.00	1.0500	42.00	.74936	.562	-.852	.374	1.339	.733
SF4	-1.00	2.00	.9250	37.00	.72986	.533	-.299	.374	.023	.733
SF5	-1.00	2.00	.9250	37.00	.85896	.738	-.362	.374	-.527	.733
SF6	-1.00	2.00	.9750	39.00	.69752	.487	-.443	.374	.592	.733
SF7	.00	2.00	1.0500	42.00	.63851	.408	-.040	.374	-.395	.733
SF8	.00	2.00	.9500	38.00	.67748	.459	.060	.374	-.708	.733

SF9	-1.00	2.00	1.0500	42.00	.84580	.715	-.901	.374	.706	.733
SF10	.00	2.00	1.1500	46.00	.66216	.438	-.170	.374	-.637	.733
Valid N (listwise)	40									

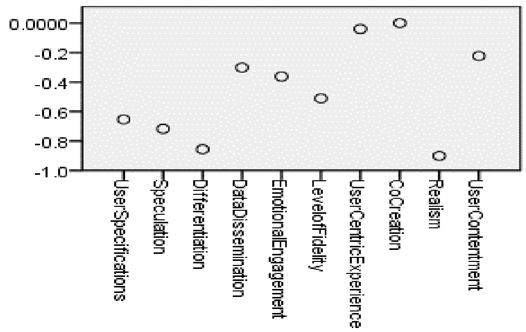

Figure 3. Skewness Statistics

According to the skewness values as shown in Fig. 3, most components have slightly left-skewed distributions, which suggests a propensity for lower scores. The distribution of SF7 is almost symmetrical, whereas SF9 shows a somewhat more significant left skew, as shown in Fig. 3. The kurtosis values show that the distributions peaked to varying degrees. While SF9 and SF3 have distributions that are just slightly peaked, SF2 has an extremely peaked distribution. The distributions for SF4, SF7, and SF10 are comparatively flat.

Table. 7., represents frequency information as well as value labels for the variables "Year" and "DT@FDD." There are two categories for the variable "Year," "Low-Experience" and "High-Experience". The data table, which had 40 respondents divided into 10 categories, forming 400 cells, was subjected to a one-way ANOVA using the following methodology. Initially, the information was arranged so that every participant supplied answers for every one of the ten categories included in the "DT@FDD" variable. Therefore, 400 cells were produced, with 40 respondents contributing to each category. These categories depict several facets or elements connected to the incorporation of DT@FDD.

Table 7. Between-Subject Factors

		Value Label	**N**
Year	1.00	Low-Experience	400
	2.00	High-Experience	400
DT@FDD	1.00	UserSpecifications	40
	2.00	Speculation	40
	3.00	Differentiation	40
	4.00	Data Dissemination	40
	5.00	EmotionalEngagement	40
	6.00	LevelOfFidelity	40
	7.00	UserCentricExperience	40
	8.00	Co-Creation	40
	9.00	Realism	40
	10.00	UserContentment	40

Table 8. Levene's Test of Equality

Dependent Variable: User Satisfaction			
F	**df1**	**df2**	**Sig.**
.928	19	380	.548
a. Design: Intercept + Year + DT@FDD + Year * DT@FDD			

A Levene's Test of Equality of Error Variances was performed on the dependent variable "UserSatisfaction". The test looks for statistically significant variations in error variances between groups as shown in the Table. 8. The test statistic (F) in this instance is 928 with 19 degrees of freedom in the numerator and 380 degrees of freedom in the denominator. The associated p-value is .548, demonstrating that the error variances between the groups are not significantly different. As a result, the premise of equal error variances is satisfied, and the RQ3 is not rejected. This shows that the comparison groups' levels of user satisfaction score variability are comparable.

Table 9. Levenes Test of Equality

Source	Type lll Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	9.710[a]	19	.511	.907	.575	.043
Intercept	412.090	1	412.090	731.065	.000	.658
Year	2.890	1	2.890	5.127	.024	.013
DT@FDD	3.310	9	.368	.652	.752	.015
Year * DT@FDD	3.510	9	.390	.692	.716	.016
Error	214.200	380	.564			
Total	636.000	400				
Corrected Total	223.910	399				

Table. 9. presents the findings of univariate analysis of variance (ANOVA) on the impact of user satisfaction, DT@FDD, and years of experience. The "Corrected Model" evaluates the model's overall statistical significance and concludes that its effects are not statistically significant (p = 0.575). Additionally, not significant are the primary impacts of Year and DT@FDD as well as their interaction. The phrase "Error" denotes variation within the groupings, whereas the term "Total" denotes variation overall. According to the R-squared value, only a small percentage (4.3%) of the variance in user satisfaction can be attributed to the independent variables. In terms of years of experience, the table does not specifically state which group of individuals with high or low experience produced the better results. The interaction effect between "Year" and "DT@FDD" is not significant (p =.716), according to Table 9. Further analysis was performed to ascertain this.

As discussed earlier, RQ3 was to determine the impact of integrating DT@FDD on user satisfaction when considering the level of experience of individuals involved. A pairwise comparison was conducted to ascertain which group, individuals with high or low experience, yielded better results in terms of user satisfaction.

Table 10. Pairwise Comparison: Bonferroni

(I) Year	(J) Year	Mean Difference (I-J)	Std. Error	Sig. [b]	95% Confidence Interval for Difference[b]	
					Lower Bound	Upper Bound
Low-Experience	**High-Experience**	-.170*	.075	.024	-.318	
High-Experience	**Low-Experience**	.170*	.075	.024	.022	

Table 10. states the RQ3, the pairwise comparisons between the less-experienced and High-Experience groups show a significant mean difference of -.170, showing that people with less experience often have lower user satisfaction scores than people with high experience. “Year” has a significant main effect (p =.024), suggesting that the variable (years of experience) has a statistically significant effect on the outcome variable. The fact that “Year” has a significant main effect implies that the number of years of experience may have an impact on differences in the outcome variable known as user satisfaction. This conclusion is corroborated by the reverse comparison, which reveals a substantial mean difference of 170 between the High-Experience group and the Low-Experience group. In the context of incorporating DT into FDD, these results emphasize the influence of experience level on user satisfaction.

In the context of an ANOVA analysis, the profile plot displays the trend of the dependent variable (User Satisfaction) at various levels of the independent variable (Experience). The profile plot in Fig. 4 shows that average user satisfaction rises as the experience level rises from low to high. The graph’s upward-trending line indicates a favorable correlation between user satisfaction and experience quality.

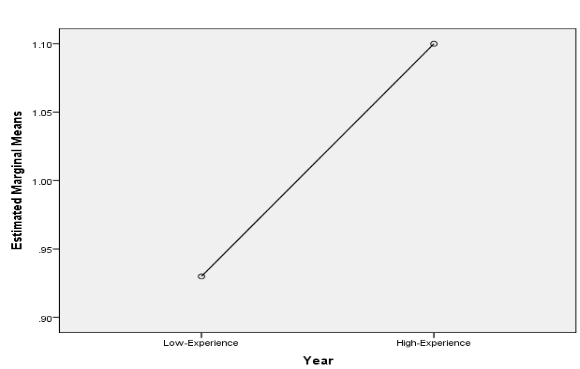

Figure 4. Estimated Marginal Means of User Satisfaction

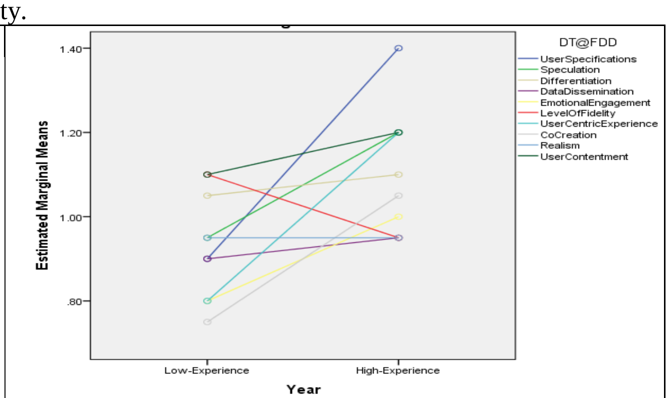

Figure 5. DT@FDD

Figure 5. Displays computed marginal means of the user satisfaction variations between groups and determines the relative influence of the independent sub-factor variables on the outcome variable. It is a frame of reference for comparing the mean scores among Low-Experience and High-Experience for research evaluation.

V. THREATS TO VALIDITY

There are four types of threats such as internal, external, construct validity, and ecological validity. The integration of DT into FDD was the subject of our study, and we used a Likert scale as a measurement tool to gauge participant attitudes and views of this approach. Internal threats include selection bias because of the small sample size of 40 randomly chosen respondents, which may restrict the generalizability of the results. The external validity of the results may be limited due to the study’s sample size and features that may not fully represent the larger population or settings of interest. Given that the study is exploratory, a smaller sample size was chosen accordingly. Respondent changes over time may affect their beliefs and actions, therefore, maturation offers a risk. The small sample size, the respondents’ unique traits, and the potential problems with ecological validity brought on by the study’s artificial environment are examples of external threats. If the operationalization of variables fails to adequately convey the essence of DT integration and user satisfaction, construct validity may be jeopardized. Due to social desirability bias, respondents may give answers that are more likely to be accepted by others rather than their real opinions. Additionally, by adopting numerous data collection techniques, performing the study in various locations, using verified measuring tools, and considering potential biases in design and analysis, future researchers can counteract these dangers. External validity can be improved by conducting the study again with larger and more diverse sample sizes. Future studies could use larger, more varied samples to guarantee higher representativeness and lower the possibility of selection bias in an effort to lessen this threat. Furthermore, the study only considered participants’ judgments based on a description of the framework, participants did not get first-hand experience with the framework.

VI. FUTURE WORK

This study acknowledges several limitations that present opportunities for future research. Such research should involve larger and more diverse participant groups. Expanding the sample size would enhance the generalizability of findings and reduce the likelihood of selection bias. DT@FDD should replicate the research across varied settings and populations to better represent the broader demographics and contexts in which the DT and FDD integration framework might be applied. Additionally, conducting the study in real-world environments, rather than artificial settings, would enhance credibility and offer deeper insights into the framework's practical effectiveness. The study can be applied to educational technology solutions, customer relationship management systems, smart home automation systems, banking and financial applications, and healthcare management systems. Finally, a significant limitation of this study is that participants assessed the framework based solely on its description rather than through direct interaction. Future research should involve first-hand experience with the framework to evaluate its impact more comprehensively. By addressing these areas, future studies could provide more conclusive evidence on the effectiveness and applicability of the proposed framework in diverse contexts.

VII. CONCLUSION

In this paper, the approach was aimed at integrating DT@FDD. DT is a collaborative method that emphasizes interaction between designers and users. Whereas feature-driven development is a customer-focused approach to software development characterized by brief iterations and periodic releases. DT helps engineers or professionals with innovative solutions on how users think, believe, and do. Utilizing a mixed-method research approach that incorporates surveys with agile teams, this study explores the viewpoints of the 40 respondents, who were practitioners with low and high levels of expertise. Effects such as user specifications, speculation, differentiation, data dissemination, emotional engagement, level of fidelity, user-centric experience, co-creation, realism, and user contentment are explored. Levene's test was performed to support the validity of ANOVA analysis, and it suggests that the error variances are approximately equal across the groups. The univariate analysis results indicate that neither the overall model nor its components, including the interaction between "Year" and "DT@FDD", showed statistical significance ($p = 0.716$). However, the main effect of "Year" was found to be statistically significant ($p = 0.024$), suggesting that the variable (years of experience) has a notable influence on the outcome variable, namely user satisfaction. This implies that differences in user satisfaction may be impacted by the number of years of experience. From an analysis of the R-squared value, independent variables originating with more experienced persons account for just 4.3% of the variance in user satisfaction. However, there is a substantial mean difference of -0.170 between the groups with high and low experience, indicating that the low-experienced individuals scored lower on user satisfaction than the high-experienced individuals.

VIII. APPENDIX

Several structured questions spanning a range of subjects, including the degree of integration of DT practices into FDD, are included in the survey. Every question in the study is thoughtfully designed to elicit answers that shed light on the research issues that are being asked.

- User Specifications: When building the overall model in FDD, conducting user interviews is essential for gaining valuable insights from their perspective. These insights enable the team to empathize with users, understand their needs, and effectively translate them into actionable problem statements.
- Speculation: When the team describes a problem, they often make self-assumptions and constraints. To ensure transparency and alignment with users, it is important to implement user stories and persona-based scenarios. These tools outline user interactions and high-level business requirements that the system should fulfill, promoting consistency between the team and users.
- Differentiation: When creating a feature list, teams commonly prioritize defining requirements without exploring innovative ideas. To address this, it is recommended to pause and conduct brainstorming sessions to identify emerging technologies and trends. Incorporating these ideas early on helps differentiate the product from competitors and opens up new possibilities before delving into detailed requirements.
- Data Dissemination: When a team is in the planning phase of feature development, communication typically centers around written documentation and demonstrations of the feature list. However, to address this challenge, it is crucial for the team to actively engage and involve cross-functional team members throughout the process through iterative feedback loops.
- Emotional Engagement: During feature design, teams commonly create prototypes solely based on the feature list without user validation. To bridge this gap, it is advisable for the team to verify the feature list and validate the prototype using user feedback. By involving users in the validation process, the team can ensure that the prototype evokes the desired emotional engagement and effectively meets user expectations and needs.

- Level-of-Fidelity: When building prototypes, teams often prioritize detailed high-fidelity designs that closely resemble the final product. However, this can pose resource constraints and difficulties in accommodating changes. To address this, teams should also include low-fidelity sketches to gather early user feedback. By validating the design with users, teams can identify potential issues, save time, and avoid unnecessary expenses in the long run.
- User-Centric Experience: When designing a feature, the team incorporates user feedback through prototyping to identify usability issues based on the feature list. To enhance the process, it is crucial to create tangible artifacts that allow stakeholders to visualize and interact with ideas. This approach ultimately improves the overall user experience.
- Co-Creation: In the process of designing a feature, there is often limited collaboration with the user. To address this, it is important to cultivate a culture of open communication among designers, developers, and users, fostering a collaborative approach to feature development.
- Realism: When designing a user interface for an app, visual aesthetics are often given less priority. However, it is recommended to focus on creating an interaction design that establishes an emotional connection with users, such as animations, to make it engaging and memorable.
- User Contentment: When testing a feature, the team typically emphasizes verifying the requirements outlined in the report. However, incorporating user feedback from the prototype phase can greatly facilitate this process, as it helps generate test-based scenarios that align with user needs and preferences.

REFERENCES

[1] Mergel, Ines, Sukumar Ganapati, and Andrew B. Whitford. ‘Agile: A new way of governing.’ Public Administration Review 81, no. 1 (2021): 161-165.

[2] Ciric, Danijela, Bojan Lalic, Danijela Gracanin, Nemanja Tasic, Milan Delic, and Nenad Medic. ‘Agile vs. Traditional approach in project management: Strategies, challenges and reasons to introduce agile.’ Procedia Manufacturing 39 (2019): 1407-1414.

[3] Hayat, Faisal, Ammar Ur Rehman, Khawaja Sarmad Arif, Kanwal Wahab, and Muhammad Abbas. ‘The influence of agile methodology (Scrum) on software project management.’ In 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 145-149. IEEE, 2019.

[4] Nawaz, Zahid. ‘Proposal of Enhanced FDD Process Model.’ (2021).

[5] Aftab, Shabib, Zahid Nawaz, Madiha Anwar, Faiza Anwer, Muhammad Salman Bashir, and Munir Ahmad. ‘Comparative Analysis of FDD and SFDD.’ International Journal of Computer Science and Network Security 18, no. 1 (2018): 63-70.

[6] Hunt, John. ‘Feature-driven development.’ Agile Software Construction (2006): 161-182.

[7] Alhazmi, Alhejab, and Shihong Huang. ‘Integrating DT into scrum framework in the context of requirements engineering management.’ In Proceedings of the 3rd International Conference on Computer Science and Software Engineering, pp. 33-45. 2020.

[8] Sohaib, Osama, Hiralkumari Solanki, Navkiran Dhaliwa, Walayat Hussain, and Muhammad Asif. ‘Integrating DT into extreme programming.’ Journal of Ambient Intelligence and Humanized Computing 10 (2019): 2485-2492.

[9] Pillay, Narendren, and Jeanette Wing. ‘Agile UX: Integrating good UX development practices in Agile.’ In 2019 Conference on Information Communications Technology and Society (ICTAS), pp. 1-6. IEEE, 2019.

[10] Dingsøyr, Torgeir, Davide Falessi, and Ken Power. ‘Agile development at scale: the next frontier.’ IEEE Software 36, no. 2 (2019): 30-38.

[11] Dobrigkeit, Franziska, and Danielly de Paula. ‘DT in practice: understanding manifestations of DT in software engineering.’ In Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering, pp. 1059-1069. 2019.

[12] Towards an Understanding of Benefits and Challenges in the Use of DT in Requirements Engineering..

[13] Senapathi, Mali, and Meghann L. Drury-Grogan. ‘Systems thinking approach to implementing kanban: A case study.’ Journal of Software: Evolution and Process 33, no. 4 (2021): e2322.

[14] Julio Cesar Pereira, Rosaria de F.S.M. Russo, DT Integrated in Agile Software Development: A Systematic Literature Review, Procedia Computer Science, Volume 138, 2018, Pages 775-782, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.10.101.

[15] (2006). Feature-Driven Development. In: Agile Software Construction. Springer, London. https://doi.org/10.1007/1-84628-262-4_9.

[16] Coad, Peter, Eric Lefebvre and Jeff De Luca. ‘Java modeling in color with UML: enterprise components and process.’ (1999).

[17] Dobrigkeit, F., & de Paula, D. (2019). DT in practice: understanding manifestations of DT in software engineering. Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. doi:10.1145/3338906.3340451.

[18] Corral, L., & Fronza, I. (2018). DT and Agile Practices for Software Engineering. Proceedings of the 19th Annual SIG Conference on Information Technology Education - SIGITE ‘18. doi:10.1145/3241815.3241864

[19] Zhou, Q. (2020). Building DT into Content Strategy. Proceedings of the 38th ACM International Conference on Design of Communication. doi:10.1145/3380851.3416738.

[20] Pereira, L., Parizi, R., Prestes, M., Marczak, S., & Conte, T. (2021). Towards an understanding of benefits and challenges in the use of DT in requirements engineering. Proceedings of the 36th Annual ACM Symposium on Applied Computing. doi:10.1145/3412841.3442008.

[21] Eickhoff, F. L., McGrath, M. L., Mayer, C., Bieswanger, A., & Wojciak, P. A. (2018). Large-scale application of IBM DT and Agile development for IBM z14. IBM Journal of Research and Development, 62(2/3), 1:1–1:9. doi:10.1147/jrd.2018.2795879.

[22] Darrin, M. A. G., & Devereux, W. S. (2017). The Agile Manifesto, DT and systems engineering. 2017 Annual IEEE International Systems Conference (SysCon). doi:10.1109/syscon.2017.7934765.

[23] Hehn, J., & Uebernickel, F. (2018). The Use of DT for Requirements Engineering: An Ongoing Case Study in the Field of Innovative Software-Intensive Systems. 2018 IEEE 26th International Requirements Engineering Conference (RE). doi:10.1109/re.2018.00-18.

[24] Newman, P., Ferrario, M. A., Simm, W., Forshaw, S., Friday, A., & Whittle, J. (2015). The Role of DT and Physical Prototyping in Social Software Engineering. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. doi:10.1109/icse.2015.181.

[25] Senapathi, M., & Drury-Grogan, M. L. (2020). Systems Thinking Approach to Implementing Kanban: A case study. Journal of Software: Evolution and Process. doi:10.1002/smr.2322.

[26] Kitchenham, B. and Charters, S. (2007) Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

[27] Aline de Oliveira Sousa and Natasha Malveira Costa Valentim. 2019. Prototyping Usability and User Experience: A Simple Technique to Agile Teams. In Proceedings of the XVIII Brazilian Symposium on Software Quality (SBQS ‘19). Association for Computing Machinery, New York, NY, USA, 222–227. https://doi.org/10.1145/3364641.3364667.

[28] OpenAI. “ChatGPT.” OpenAI, v3.5, https://www.openai.com.