A Case Study on Dual-Track Development in Agile Software Development

Henning Rückborn
CEWE Stiftung & Co. KGaA - Research & Development - Web Applications
Oldenburg, Germany *henning.rueckborn@cewe.de*

Eva-Maria Schön
University of Applied Sciences Emden/Leer - Faculty of Business Studies
Emden, Germany *eva-maria.schoen@hs-emden-leer.de*

Abstract

Dual-track development offers a promising strategy for integrating User Experience (UX) into cross-functional agile software development. This case study explores the practical implementation and application of dual-track development within a real-world context, aiming to identify effective practices, challenges, and impacts on workflows. A development team was supported throughout its adoption of dual-track development. Findings indicate positive influences on user-centricity, stakeholder integration, and concept work efficiency. Specifically, establishing a dedicated discovery track and systematically involving stakeholders proved beneficial. This study provides empirical evidence for dual-track development's advantages in integrating UX within agile, cross-functional teams.

Keywords: dual-track, agile software development, user experience, UCD, HCD

1. Introduction

Agile software development (ASD) [1] and Human-Centered Design (HCD) [7] originated from distinct motivations. Agile aims to boost customer satisfaction through incremental releases and high adaptability to changing requirements without compromising software quality, contrasting with plan-driven approaches like the waterfall model [3]. HCD, conversely, focuses on ensuring usability and UX, aspects often underrepresended in traditional and early agile methods [2], [13]. The lack of an explicit UX role in agile development can lead to HCD goals being neglected [16].

In 2007, Sy [18] highlighted the necessity of integrating HCD work with development through iterative processes, proposing parallel "Interaction Designer" and "Developer" tracks to circumvent the waterfall approach and enhance products. Building on this, Dual-Track Development (DTD) emerged. This agile model combines the benefits of ASD with a strong emphasis on user-centricity, integrating the HCD process and meeting the criteria for user-centered agile software development (UCASD) [4].

Agile product development is conceptually divided into product discovery (defining the right product through new ideas) and product delivery (building the product right, focusing on how to deliver what) [14]. DTD is an approach to implement this via two parallel tracks within a team: the discovery track and the delivery track (see figure 1), using a product backlog as their interface. The discovery track iteratively conducts the HCD process, exploratively generating prototypes or implementation specifications. The delivery track then implements these software features, delivering them as new product increments [4], [9], [15], [19].

This case study analyzes the strategies a team at CEWE Stiftung & Co. KGaA (CEWE) used to adopt a dual-track approach for developing an online shop search function. The study aims to identify best practices and challenges, and to examine how DTD facilitated the integration

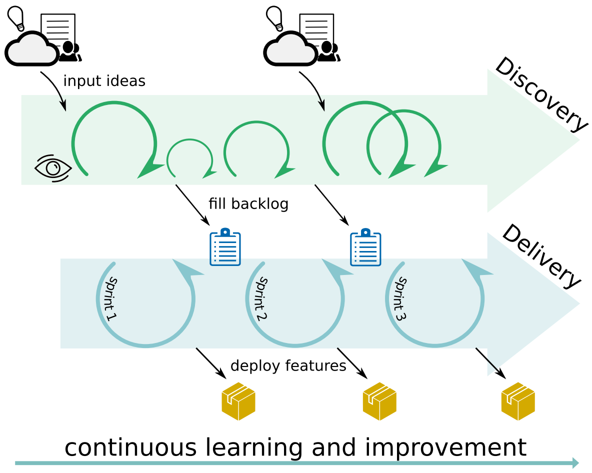

Fig. 1. Schematic representation of dual-track development

of UX work into the team's agile software development. This study addresses the following research questions (RQ1, RQ2, RQ3):

RQ1 What are the best practices and challenges of working with dual-track development?
RQ2 How does dual-track development strengthen user-centricity and the UX role?
RQ3 How does the dual-track development approach promote stakeholder involvement?

2. Related Work

This section provides an overview of related work. Since the publication of DTD in 2007 by Sy [18], it has become standard practice. Agile teams use DTD to maintain continuous synchronization of discovery work and software delivery throughout the entire product development process. However, research on this topic is rather limited.

Silva da Silva et. al [16] conducted a Systematic Literature Review (SLR) in 2011. As a result of their study, they proposed a framework for integrating agile and HCD. This framework comprises two tracks, one for discovery work and one for software delivery, and is based on the model proposed by Sy. Another SLR was conducted in 2015 by Brhel et al. [4]. Their results explicitly state that separating product discovery from product creation is a key principle to integrating Agile and HCD.

In 2019, a study by Péraire [9] investigated the application of DTD as a component of a software requirements and interaction design course in academia. The study provides a thorough overview of the course, including task descriptions for both the discovery and delivery tracks, as well as insights into the real-world application in an academic setting.

Sedano et al. (2020) [15] conducted a study using a grounded theory research approach, observing teams working on agile transformation projects over three years. The authors provide a clear description of DTD as it is applied today, focusing on a specific set of practices. The authors state their contribution more as *"transferable insight of organizing development as two parallel tracks than the specific constellation of practices"* [15].

In 2021, a study by Trieflinger et al. [19] conducted a grey literature review. The authors identified five models for implementing DTD in a practical environment, including the discovery-delivery cycle model, now-next-later product roadmaps, lean sprints, product kata, and dual-track Scrum.

Analyzing the related work we can summarize that early publications (*e.g.,* [4], [16]) emphasize the importance of separating product discovery and delivery and provide general guidance on how the two tracks can be applied in practice. More recent studies focus on the application of DTD from an organizational perspective (*e.g.,* [15], [19]) and offer concrete practices for its implementation. One study [9] presents experiences with DTD in academia as part of a study program. In contrast, our study presents an in-depth analysis of a specific case in industry, focusing on a software company. In particular, we investigate best practices, challenges, user-centricity, and stakeholder involvement that arise in the context of applying DTD.

3. Methodology

The organization selected for this study faced specific challenges, namely *Dealing with design and concept work in Scrum* and *Consideration of UX-driven development in sprints*. These challenges often led to difficulties in balancing UX design with the agile development process, especially within the limited timeframes of Scrum sprints. DTD was considered a promising solution to address these issues, as it enables a parallel stream of design and development work, ensuring that UX considerations are continuously integrated throughout the development cycle.

For this case study, we adopted a qualitative research approach to investigate a team's experiences with DTD [10]. We conducted semi-structured interviews with five out of six team members to gain detailed insights into their perspectives. The collected data underwent qualitative content analysis following Mayring and Fenzl's methodology [8] to identify patterns and themes. A comprehensive description of the research process is available in the research protocol [11].

3.1. Case study context

CEWE is a leading European provider of customized photo products, including photo books, calendars and gifts, offering continuously developed, user-friendly software. The web platform CEWE Online Photo Service (COPS) is central to its marketing strategy for online product dissemination. COPS version 5, relaunched in 2019, introduced a new technical foundation and vertical cross-functional teams [17], aligning development with customer journey domains. Due to prior technical limitations and to enhance user-friendliness and meet customer expectations, a comprehensive full-text search initiative began in early 2021. Initially, the responsible team for the new full-text search employed Scrum [12], focusing on evaluating software components and developing an operating concept. Despite creating a backlog (epics, stories, bugs) and implementing all Scrum events and artifacts, two fundamental issues emerged over time, posing significant challenges.

Dealing with design and concept work in Scrum: It became evident that the team members' estimates of story points for design and concept stories lacked reliability. The inherent complexity of these tasks often prevented scope reduction, leading to unbalanced team utilization and frequently overloaded sprints.

Consideration of UX-driven development in sprints: The question arose as to how the creation of prototypes, the planning of user tests, and the consideration of user experiences could be adequately taken into account and estimated in the sprints in order to be able to make reliable statements regarding the duration.

These two problems quickly made it clear that UX research and conceptual work could not be optimally handled in Scrum iterations for that team.

3.2. The Team

The team was cross-functional, comprising expertise in front-end development, full-stack development, UX research, and product ownership. Individual team members, their company affiliation, team roles, and prior knowledge are detailed in Table 1.

The initial team, comprising P2, P4, P5, and P6, possessed a strong agile mindset and a history of successful collaboration, which aided rapid team building and workflow establishment.

Table 1. Role, company affiliation and experience of team members

P1	Junior Developer	2022	Theoretical agile knowledge, no professional experience
P2	Product Owner	2008	Previously Developer
P3	Team coach	2023	Agile Coach, no software development experience
P4	UX Researcher	2003	Primary focus on UX aspects, also UI design
P5	Frontend Developer	2009	Clear focus on frontend development
P6	Fullstack Developer	2019	Backend, system architecture, author of this case study

In late 2022, P1 joined as a young professional and was continuously integrated. Prior to team coach P3's arrival, the team received support from two external coaches (April 2022 - December 2022; January 2023 - April 2023). This self-organized team showed minimal need for direct management.

3.3. Implementation of Dual-Track Development in the Team

In order to meet the project requirements, the team implemented a DTD approach structured according to the model shown in figure 1. Under the UX leadership of P4, the discovery track generated, verified, and refined new ideas into actionable delivery stories, forming the basis for the delivery backlog. The delivery track, managed by developers P1, P5, and P6, focused on implementing these stories using Scrum with 14-day sprints. Product Owner P2 maintained and prioritized the delivery backlog, collaborating with P4 on stakeholder management. To ensure close collaboration, continuous exchange between UX and development was maintained throughout the implementation. Atlassian Jira was used to visualize and manage both tracks: the discovery track as a Kanban board and the delivery track as a Scrum board.

3.4. Data collection

To answer the research questions RQ1-RQ3, semi-structured interviews were conducted.The qualitative research approach was chosen due to the small number of interview partners. In addition, semi-structured interviews are a widely used, differentiated and methodologically well-developed method for generating qualitative data [6].

Development of the interview guide: We developed a structured interview guide using the SPSS method (collect, check, sort, subsum) to balance openness with the necessary research structure [6]. Initial questions and topics were compiled, then reviewed against Helfferich's criteria [6] to ensure they were open-ended, narrative-stimulating, understandable, and non-suggestive. To refine the guide, a test interview was conducted with an individual not part of the team mentioned in section 3.2. The revised interview guide is organized into sections: introduction, personal details, expectations, experiences, summary questions, open questions, and conclusion.

Conducting the interviews: Interviews with P1 and P5 were conducted on-site in a meeting room, with only the interviewee and interviewer (P6) present. Each interview lasted 45 to 55 minutes. Questions were posed in German, and responses were given in German (four participants) or English (one participant). Conducted between November 20 and November 28, 2024, all interviews were recorded with consent and subsequently transcribed.

3.5. Data Analysis and Synthesis

Interview analysis followed the qualitative content analysis approach by Mayring and Fenzl [8]. The MaxQDA software was utilized for efficient material management, with its integrated arti-

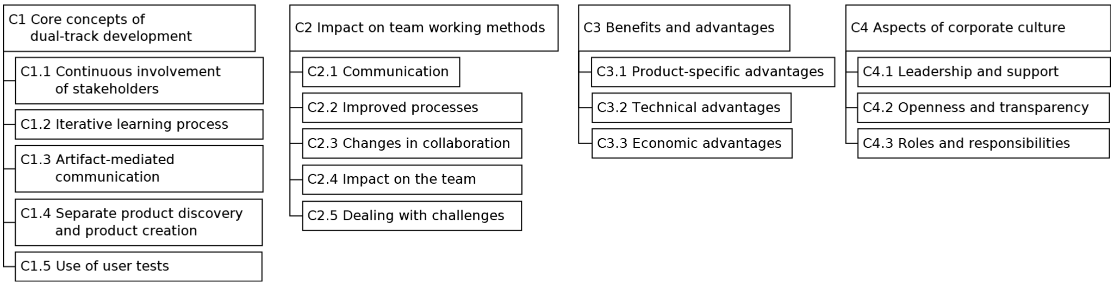

Fig. 2. Representation of the consolidated code system

ficial intelligence (AI) functions assisting in the identification of potentially overlooked codes during evaluation.

Deductive Coding: First, a deductive coding of the interview transcripts was performed. The generation of deductive codes was informed by extant literature, and each transcript was subsequently coded with these codes. The UCASD principles [4] were selected because they represent central aspects of ASD in combination with HCD.

Inductive Coding: In the subsequent phase, the data material was subjected to inductive coding. The objective of this phase was to also take into account topics and aspects that arose directly from the material itself and had not already been coded in advance. A subsequent review was conducted to ensure that all interview transcripts were reviewed with all available codes.

Consolidation: The results of the deductive and inductive coding processes were subsequently merged in a comparative manner to establish a consolidated code system. This resulted in four main categories (Core concepts of dual-track development (C1), Impact on team working methods (C2), Benefits and advantages (C3), Aspects of corporate culture (C4)), which were further refined with additional codes. A representation of the code system is shown in figure 2.

Data Synthesis: Subsequent to the content analysis, a synthesis of the data was conducted with the objective of consolidating the statements and findings derived from the interviews into a comprehensive representation, thereby facilitating the formulation of conclusions.

4. Results

The following section presents the evaluation's key findings, which are organized according to the codes of the consolidated code system in figure 2. Explicit statements made by respondents have been edited in some cases to improve readability.

4.1. Core concepts of dual-track development (C1)

The analysis of the interviews indicates that the principles of UCASD [4] were indeed considered by the team through the implementation of DTD.

Continuous involvement of stakeholders (C1.1): A critical factor in stakeholder engagement was their involvement throughout the entire development process, not just the design phase. The team fostered a deliberative process, discussing and integrating ideas and suggestions directly into implementation. "*The ideas were brought to the Product Owner and then tickets were simply created on the Discovery Track [Board] and prioritized within the team.*" (P4) Crucially, this effort ensured consideration for both stakeholders and management representatives, alongside end-users, whose input was incorporated through user tests (P5). Stakeholder coordination was primarily achieved through regular meetings and workshops, facilitating discussions on topics from both discovery and delivery tracks. This approach helped align expectations and ideas, and identify necessary compromises. "*There were barriers, with specific customer's needs or stakeholder wishes. People would say, if we do it this way it will take* *too long. It will be too much work. Where is the added value? This is why we started direct communication with the stakeholders.*" (P3) This close exchange fostered transparency, mutual understanding, and respectful interaction.

This approach led to a substantial enhancement in the quality and efficiency of feature development (P3). Furthermore, stakeholder feedback and requests were repeatedly aligned with the product vision: "*Sometimes it's difficult to communicate to stakeholders that user-driven design doesn't necessarily focus on them as users. But that they are the stakeholders, and the users are a separate entity who will actually use the software.*" (P5) To that end, regular coordination meetings were scheduled with the relevant stakeholders, during which the findings from the UX research were compiled and prioritized together (P3).

Iterative learning process (C1.2): The interviews further illuminate the iterative learning process that the team went through. The interviews state that "*you didn't have to talk about the nitty gritty details*" instead, "*you could just start [and] try it out*" (P4). This put the focus on the essentials and the team was able to contribute "*ideas not only from a UX perspective*", but also work together on concepts (P4). It also meant that "*everyone, including the developers, was involved in an exchange [with the] stakeholder*" (P3). This facilitated a more comprehensive understanding of the stakeholders' needs by the team members, who were then able to incorporate them into their work (P3, section 4.1). This iterative learning process also meant that ideas could be implemented more quickly than would otherwise have been the case (P5) and solutions could be found that all team members and stakeholders identified with (P5). The initial hypothesis, which posited that the delivery track was the sole responsibility of the UX role P4, proved challenging at first (P4). This dynamic has evolved over time, underscoring the necessity for ongoing clarification and coordination of roles and responsibilities within the team.

Artifact-mediated communication (C1.3): Seamless idea exchange among team members was facilitated by various artifacts, including Jira boards, user stories, prototypes, design drafts, and user tests. Special internal workshops were crucial for incorporating diverse perspectives, encompassing UX, development, and web analytics. "*There was a special workshop for the search [, in which] conceptual work was carried out not only with people outside the team, but also with other departments. That really helped, because it brought in different perspectives not only between UX and development, but also from other departments.*" (P1) The use of prototypes and design drafts provided concrete product visualizations, supporting iterative development (P5) and enhancing the quality of user stories by integrating prototypes. Addressing the research question regarding challenges, it was observed that despite frequent communication in daily stand-ups, the discovery track was occasionally overlooked. This led to the team having incomplete awareness of all tasks (P4, P1). The primary reason for this was the confusing presentation of the two tracks as separate Jira boards (P1).

Separate product discovery and product creation (C1.4): The Discovery Track placed primary emphasis on conceptual work and the identification of products. "*To develop a rough plan for concepts, i.e., what you want to implement. It worked very well for that.*" (P4) Concurrently, the delivery track concentrated on product development and implementation. "*Delivery has always been relatively clear: it is what is delivered. In other words, the finished product, which in our case was the software.*" (P1) The separation of responsibilities "*was very clear, everything [was] very well structured. The responsibilities were clearly defined, clarified, it ran smoothly.*" (P3) This focus was evident in the fact that "*not everything was on one board. That meant you didn't have that overload of thinking, oh, I have to do think about that now too.*" (P1) In addition to the focus, the Discovery Track also provided "*sufficient time to intensively develop the design concept*" without any time pressure (P2). This made it possible to test concepts extensively. In the discovery track, "*the concepts could be made as big as necessary*", "*but as small as possible*", without having to squeeze them into a scrum rhythm (P4). "*The separation between the two tracks was like the flywheel in a car. One track could work on something, and if* *it ran into difficulties, the other track could [still] work independently.*" (P5) The fact that "*the topic of UX and concept work was really given its own space, which is no matter of course*" (P2) is "*already a key factor in ensuring that a good result was achieved in the end*" (P2). Initially, the separation of tracks presented challenges, particularly the assumption that UX research in the discovery track would be solely handled by the UX role. Through continuous learning and adaptation, the team evolved this approach. "*It has changed in that the concepts are no longer created by the UX role alone, but are worked on by the entire team.*" (P4) Another challenge involved maintaining a balanced exchange between the two tracks. "*Sometimes the discovery board got a little lost because the focus in daily work was more on development topics, and those were processed more quickly or received changes or a different status more quickly.*" (P1)

Use of user tests (C1.5): The team "*tested extensively at the time, conducted many user tests, and was able to continuously incorporate user feedback.*" (P4) In this particular context, "*the process definitely helped to be able to react quickly when issues came up in a concept that needed to be verified with users or with results from user tests before a decision could be made.*" (P4) In addition, the use of user tests is "*a good tool for seeing which direction you are going in without letting personal opinions come into play*" (P4). This does not always seem to be the case, because "*in the past, there were situations [in which] people conducted user tests [and the results] were simply questioned if they did not fit in with their own world view of our software landscape*" (P2). The challenges encountered include dependencies on external providers for user testing and the difficulty of estimating the time required for this process (P4). Furthermore, a paucity of trust in user testing poses a challenge. This issue was addressed through a process of clarification and prioritization. Achieving this balance necessitated a careful consideration of user interests and the development effort required. (P3)

4.2. Impact on team working methods (C2)

The DTD approach promoted more intensive communication within the team and with stakeholders. This, in turn, contributed to improved processes and a change in collaboration.

Communication (C2.1): The division into a discovery track and a delivery track ensured structured and transparent communication, as "*the communication channels were clearly defined and regulated*" (P3). The regular daily meetings, which were subdivided into sections for the two tracks, facilitated effective exchange of information and a shared understanding of the product's progress (P5). The close interdisciplinary cooperation fostered a high degree of mutual trust and understanding among the participants (P5). The developers were involved in the conceptual work at an early stage and the "*technical skills were used to evaluate the actual possibilities and environment*" (P5). Concurrently, the UX role had a direct link to the developers to exchange feedback and ideas (P4, P5). Stakeholder communication also benefited from the dual-track approach, as concepts could be presented earlier and developed iteratively (P1, P5). Regular reviews ensured that everyone involved remained informed (P1). Regarding the communication with the stakeholders we observed one challenge. It is imperative to obtain a comprehensive understanding of both tracks, as these were mapped in disparate Jira boards and not all information was readily accessible at first glance (P1).

Improved processes (C2.2): The implementation of a DTD approach resulted in numerous enhancements to the team's work processes. The approach enabled "*better communication and collaboration within the team, which led to higher product quality and fewer iterations for debugging*" (P3). The process of creating stories for the delivery track was enhanced by the involvement of development in the discovery track, leading to an improvement in the quality of the stories. The developers knew "*the scope , everything became much clearer. The stories were simpler. And with simpler stories, the work is much more fun. [They] were also able to reduce the scope of work because [they] had a document that stated what needed to be done.*" (P5) In addition, the process of involving development in the discovery track resulted in "*a clearer* *roadmap , what can come in what order, [which] topics can come up.*" (P1) It was hypothesized in advance that the challenge would be that "*either the conceptual track has nothing to do at the moment or the development track has nothing to do because everyone is waiting for each other.*" (P2) Contrary to the initial concern, this did not materialize in the team under investigation (P2).

Changes in collaboration (C2.3): The DTD approach yielded several enhancements in teamwork, including: It led to greater cohesion within the team and a shared understanding of the product, as "*the entire team worked on the concepts and was always available at certain stages to discuss them.*" (P4) It is emphasized that "*there was a very good exchange within the team and then ideas always came not only from a UX perspective, but from everyone in the team.*" (P4) The developers' increased involvement in the design process enabled them to contribute their technical expertise. Overall, teamwork was improved, ensuring that "*the development stories were derived directly from the concepts.*" (P5) The roles and responsibilities of the various tracks and boards were clearly defined, facilitating effective communication and coordination among the members (P3). Collaboration was supported by regular exchange meetings (section 4.1), such as daily meetings, reviews, and dedicated concept meetings. In addition, there was a shared view of the product (P1, P2, P5). The biggest challenge in the collaboration was described as "*experiencing many situations, both within the team and with other stakeholders, where opinions clash. It is actually never easy, or rarely easy, to bring these together so that in the end a common idea emerges that everyone likes or can support*" (P2).

Impact on the team (C2.4): The delineation of roles and responsibilities between the discovery and the delivery track resulted in enhanced transparency and self-organization. "*Everything was very clear and well structured. Responsibilities were clearly defined and clarified, so everything ran smoothly.*" (P3) The division into two tracks facilitated a more efficient work process by decoupling concept work and development, thereby avoiding periods of waiting. "*It allowed people to focus on what they wanted to do and were they are good at.*" (P5) The team members were able to concentrate on their respective areas of expertise and work independently on their tasks, which strengthened mutual trust. "*Everyone expressed their opinion and was heard. That's exactly how trust was possible.*" (P4) The incorporation of developers into the discovery track further fostered the team's capacity for innovation, as ideas and proposed solutions emanated not only from the UX perspective but from the collective perspective of the entire team (P4).

Dealing with challenges (C2.5): To overcome challenges, the team implemented a strategy of enhanced communication, both with stakeholders and internally among team members. Occasional disagreements were addressed in an objective manner, and conflicting product decisions were supported by user test results. As time progressed, the team came to discern which issues merited contention and where it was preferable to exercise compromise (P2, P3).

4.3. Benefits and advantages (C3)

The evaluation of the interviews indicates that the DTD approach brought particular product-related advantages to the fore, such as user-centricity and a clear product vision. The resulting risk minimization also had a positive impact on economic benefits. In the field of software engineering, technical advantages in terms of code quality, software architecture, and maintainability were often secondary considerations.

Product-specific advantages (C3.1): The iterative work, the involvement of developers in the discovery track, and the consideration of user test results resulted in "*fewer missing features, higher quality, no or few subsequent iterations for bug fixing or adaptation to customer requirements, because these were identified in advance and implemented with high quality in the work.*" (P3) In addition, the working method supported the development of a product vision. The collaborative iterative work "*helped us to really have a focused [product] vision for the search*" and not fall into "*endless cycles*" (P1). In terms of challenges encountered, the primary issue was the alignment of user test results with the interests of the stakeholders. This was perceived as a difficult task at the outset. Nevertheless, the team was able to address this issue through open and objective discussions with the affected parties, "*particularly when supported by empirical evidence.*" (P2)

Technical advantages (C3.2): Intensive collaboration in the discovery track and joint creation of concepts and user stories for implementation resulted in increased modularity and interchangeability of the system (P5). "*The different parts are much more replaceable and modular because we had more time and more direction to work on them.*" (P5) It has been established that, in general, the enhancement in code quality is not regarded as being impacted by the process. Developers identified their increased integration into the discovery track as a challenge. Primary concerns centered on the initiative and effort required to provide status updates on concepts, alongside the frequent context switches between development and conceptual work.

Economic advantages (C3.3): "*Due to the fact that all relevant stakeholders, including developers, were involved in the discovery track and were able to contribute their technical expertise to the development*" (P3), risks were identified and eliminated at an early stage. The discovery track facilitated extensive testing and iteration of concepts and designs, unencumbered by the time constraints imposed by development sprints. This led to increased responsiveness to changes, which can be attributed to "*the extensive testing that was already taken into account in the concepts beforehand*" (P4). This also resulted in fewer iterations for bug fixes. The implementation of development stories could be executed with greater concreteness and detail, given the availability of visual concepts and prototypes (P1). The artifact-mediated communication (section 4.1) used in this way led to fewer iterations in error correction and a faster start of development work, "*since there was usually already a presentable prototype, which was then created not via HTML , but mostly via Adobe XD*" (P1).

4.4. Aspects of corporate culture (C4)

The delineation of roles and responsibilities was instrumental in fostering open and transparent communication. Leadership and support, akin to that provided by a team coach, centered more on fostering autonomy and facilitating self-organization among team members.

Leadership and support (C4.1): The interviewees emphasized that the team demonstrated a high degree of personal responsibility and autonomy in the DTD process, enabling them to make decisions and develop solutions independently. It was emphasized that the team "*actually had a relatively high degree of freedom to decide and try things out*" (P4). The managers exhibited a high degree of confidence in the team members' expertise, recognizing their ability to contribute their unique strengths and provide mutual support. This trust is identified as a pivotal factor contributing to the success of the project (P3). The team further emphasized that an open communication culture and tolerance for mistakes engendered a constructive working atmosphere in which mistakes were regarded as opportunities for improvement (P3). Furthermore, it was asserted that all individuals were at liberty to articulate their perspectives without impediment and were also given due consideration (P4). The team did not encounter any challenges in their interactions with management. The management respected the team's scope for action and working methods and supported them in their efforts. The team was entrusted with the responsibility of implementing the process successfully (P4).

Openness and transparency (C4.2): The team's collaborative efforts were marked by a commitment to openness and transparency. It is emphasized that "*these were always highlight moments when everyone could contribute their ideas and demonstrate their expertise on the topics.*" (P2) The team, comprising stakeholders, collaborated on the conceptualization, thereby fostering a collective understanding and transparency regarding the requirements (P3). Despite the existence of explicit role delineations and responsibilities with respect to the discovery track and concept development, the track was conceptualized from the outset to be open, thereby enabling all team members to participate and fostering collaborative concept development (section 4.1). Furthermore, the team exhibited a high level of trust in each other and placed significant value on the expertise of each member. The team's capacity to respond flexibly to changes was attributable to its members' diverse strengths (P3, P4, P5, section 4.4). The respondents indicated that the most significant challenges pertained to disagreements regarding the direction of the product, although these were primarily personal opinions. In such cases, it was also decided to involve users through user tests in order to create an objective basis for decision-making (P2, P3, P4, section 4.2).

Roles and responsibilities (C4.3): The delineation of discovery and delivery tracks has resulted in a more precise allocation of roles and responsibilities within the team. "*They had specific responsibilities. They took responsibility for the stories. Each role was separate, whether UX, UI, developer, or Product Owner.*" (P3) The UX role P4 was more focused on conceptual work in the discovery track, while the developers concentrated mainly on the delivery track and the implementation of the stories. The Product Owner P2 managed the product backlog and wrote the delivery track stories. This division of tasks can be attributed to a very committed UX role, which administered the discovery track independently (P2). The communication with relevant parties was also divided up, with the Product Owner serving as the primary point of contact. "*[As UX role], I did [the communication] for the concepts themselves, and otherwise the Product Owner did it.*" (P4) The team coaches assumed a more supportive role, as the experienced team demonstrated the capacity to manage the processes independently (P4, P5). Initially, there was a tendency within the team to perceive the UX role as the owner of the discovery board (P4) and not primarily as the person in charge of the discovery track. However, over time, this dynamic shifted, leading to the emergence of a more expansive set of expectations and an augmentation in the scope of tasks, including stakeholder communication (P3). It was also necessary to actively shape the exchange and communication between the tracks to avoid the formation of knowledge silos.

5. Discussion

This case study offers substantiated evidence regarding the effects of DTD on the working methods of the team studied. However, these positive effects are not reflected in an explicit increase in product quality (section 4.3), but rather in the optimized processes and behaviors of the team, as well as in the team's approach to requirements and user-centricity, with these elements being incorporated into the product to be delivered. This phenomenon engenders economic advantages, including the mitigation of risk in the domain of software development (section 4.3).

In response to the inquiry RQ1 regarding optimal methodologies, it becomes evident that the fundamental principles (section 4.1) assume a particularly significant role. The utilization of these tools has been demonstrated to promote and accelerate communication, thereby enhancing processes within the team and with stakeholders. The achievement of this objective is predominantly facilitated by artifact-mediated communication (section 4.1) and the implementation of the discovery track (section 4.1). This establishes a framework for the continuous involvement of stakeholders in the development process and also fosters an iterative learning process (section 4.1) in which all participants can engage with the product, thereby cultivating a shared product vision and building mutual understanding.

The implementation of user tests (section 4.1) has been demonstrated to enhance stakeholder understanding and introduce an objective decision-making dimension, such as for prioritizing functions. A beneficial approach involves the utilization of prototypes or design drafts as templates for user stories. These measures have been shown to enhance quality, mitigate the likelihood of unfavorable developments, and prevent the need for frequent rework.

In addressing the research question RQ1 regarding challenges encountered in DTD, it is evident that maintaining a balance between stakeholder requirements and user interests is of particular significance. The integration of stakeholders into the discovery track and the continuous learning process serves to effectively address these concerns. It is imperative to elucidate the roles and responsibilities of each team member. These roles and responsibilities must be clearly defined and subsequently adapted and established as necessary.

Research question RQ2 aimed to investigate how DTD strengthens user-centricity and the UX role. Findings indicate that DTD enhances user-centricity by integrating user testing with the discovery track and promoting continuous user feedback. This is achieved through two main approaches: user-centered prototype development guided by user testing, and increased developer involvement in the discovery track, fostering a deeper understanding of user needs. The discovery track consequently establishes a central position for the UX role, which becomes responsible for concept work, stakeholder coordination, and validating user interest, thereby comprehensively contributing its expertise to the product's user-centered orientation.

The discovery track is fundamental for effective stakeholder integration, addressing research question RQ3. It creates a protected space for collaborative idea generation and product development with stakeholders, both in daily work and dedicated workshops (section 4.1). Regular joint meetings, where new findings (e.g., from user tests or analytics) are discussed, foster a continuous, iterative learning process. This approach demonstrably enhances the shared understanding between stakeholders and the team. This contributes to the enhancement of the acceptance and recognition of the expertise of the various roles.

The findings of this study align with the principles of UCASD [4], thereby substantiating the efficacy of DTD as a practical framework for implementing these principles. Specifically, this framework functions as an organizational model for teams and provides an overarching structure for the entire development process. In this structure, teams in the discovery track may utilize methods such as HCD [7] or Lean UX [5] to ensure UX work is iterative and validation-driven, while Scrum [12] has established itself as the preferred process model in the delivery track.

5.1. Limitations

As with any research, certain limitations arise from the methodological approach chosen. For the discussion of our limitations we use the threats to validity schema according to Wohlin et al. [20] and Runeson and Hoest [10].

Construct validity: One limitation of this study is that the research questions aim to explore how DTD can improve the way of working in cross-functional teams. However, the study adheres closely to an interpretative paradigm, which leads to subjective statements that may not fully capture the complexity of the phenomena. This subjective perspective can limit the clarity of the cause-and-effect relationships between DTD and the observed improvements.

Internal validity: The study cannot conclusively determine whether DTD alone is responsible for the improvements observed. Other factors, such as the organizational environment or the agile mindset, might have contributed to some of the changes. As such, the internal validity of the findings is constrained by the inability to isolate the specific influence of DTD from other potential contributing factors. The first author of the study was part of the team and actively contributed to the software development process. To mitigate the risk of bias, the second author of the paper, a senior researcher, supervised the study and conducted quality control.

External validity: The study's external validity is limited by the fact that it focuses on a single team within a highly particular environment. This makes it difficult to generalize the findings to other teams or settings. Although the results offer valuable insights, the transferability of the conclusions to other contexts remains challenging.

6. Conclusion

The results of the study emphasize the necessity of allocating dedicated space, specifically the discovery track, for exploratory activities such as user research and idea development, and to utilize this space in a systematic manner. In practice, this means:

Open space for the discovery track: Organizations that aspire to implement DTD in teams are advised to allocate sufficient resources and designated space to the discovery track. This dedicated space enables teams to make well-founded decisions based on user research and concept development, thereby significantly increasing the quality and user-centricity of the product.

Promoting systematic stakeholder integration in the discovery track: It is imperative to recognize the pivotal function of stakeholder engagement in ensuring the success of a product. In order to achieve this objective, teams should proactively integrate stakeholders into the discovery track. This will allow for the understanding of their perspectives and expectations at an early stage, and for the incorporation of these elements into product development. Regular formats for discussion and open communication are essential for this.

Cultivate a culture of transparent and artifact-mediated communication: As previously mentioned, an open and transparent communication culture within the team and with stakeholders is crucial for successfully utilizing DTD. The targeted use of artifacts, such as prototypes, has proven to be an effective means of facilitating communication, visualizing requirements, and promoting a common understanding within the development team. These artifacts offer a significant foundation for effective coordination between the UX and development teams.

Using user evidence as a basis for decision-making in the discovery track: It is imperative for teams to methodically incorporate user testing into the discovery process and utilize the insights derived from these tests in conjunction with stakeholder requirements as a foundation for development decisions. This approach enables a more objective prioritization of features and the validation of concepts, thereby reducing the risk of misdevelopment.

Nevertheless, it is imperative to underscore that while DTD establishes a framework for integrating user-centric evidence, business objectives and frameworks must nevertheless be considered. In conclusion, the concept of DTD can be categorized as a framework for agile software development that improves collaboration between UX and development. Future research endeavors may concentrate on the impact of particular factors, such as the manner in which software architecture affects the incorporation of UX work.

References

[1] Beck, K., Beedle, M., Bennekum, A.v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Sutherland, J., Thomas, D., van Vliet, K.: Manifesto for agile software development (february 2001), `https://agilemanifesto.org/`

[2] Beyer, H.: User-Centered Agile Methods. Morgan & Claypool Publishers (2010)

[3] Boehm, B.: A survey of agile development methodologies 45, pp. 119

[4] Brhel, M., Meth, H., Maedche, A., Werder, K.: Exploring principles of user-centered agile software development: A literature review. Information and Software Technology 61, pp. 163–181 (2015)

[5] Gothelf, J., Seiden, J.: Lean UX: Applying Lean Principles to Improve User Experience. "O'Reilly Media, Inc."

[6] Helfferich, C.: Die Qualität qualitativer Daten: Manual für die Durchführung qualitativer Interviews. SpringerLink Bücher, VS Verlag für Sozialwissenschaften, 4. auflage edn.

[7] International Organization for Standardization: Iso 9241-210:2019(en) ergonomics of human-system interaction — part 210: Human-centred design for interactive systems (2019)

[8] Mayring, P., Fenzl, T.: Qualitative Inhaltsanalyse, pp. 691–706. Springer Fachmedien Wiesbaden, Wiesbaden (2022)

[9] Péraire, C.: Dual-track agile in software engineering education. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). pp. 38–49 (2019)

[10] Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering 14(2), pp. 131–164

[11] Rückborn, H., Schön, E.M.: Research protocol for the paper entitled: A case study on dual-track development in agile software development (Apr 2025), https://zenodo.org/records/15703844

[12] Schwaber, K., Sutherland, J.: The scrum guide the definitive guide to scrum: The rules of the game (november 2020), https://scrumguides.org/, 2020 Edition

[13] Schön, E.M., Silva da Silva, T., Hinderks, A., Sharp, H., Thomaschewski, J.: Introduction to special issue on agile ux: challenges, successes and barriers to improvement. Information and Software Technology 158, pp. 107193 (2023)

[14] Schön, E.M., Thomaschewski, J., Escalona, M.J.: Lean user research for agile organizations. IEEE Access 8, pp. 129763–129773 (2020)

[15] Sedano, T., Ralph, P., Peraire, C.: Dual-track development. IEEE Software 37(6), pp. 58–64 (2020)

[16] Silva da Silva, T., Martin, A., Maurer, F., Silveira, M.: User-centered design and agile methods: A systematic review. In: 2011 Agile Conference. pp. 77–86 (2011)

[17] Skelton, M., Pais, M.: Team topologies: organizing business and technology teams for fast flow. IT Revolution

[18] Sy, D.: Adapting usability investigations for agile user-centered design 2, pp. 112–132

[19] Trieflinger, S., Münch, J., Heisler, B., Lang, D.: Essential approaches to dual-track agile: Results from a grey literature review. In: Klotins, E., Wnuk, K. (eds.) Software Business. pp. 55–69. Springer International Publishing, Cham (2021)

[20] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer (2012)

Acknowledgments

AI tools were used for translation in the preparation of this submission, and AI features built into MaxQDA were used to review the coding process.