Exploring Principles of User-Centered Agile Software Development: A Literature Review

Systematic Review Paper

Manuel Brhel[a], Hendrik Meth[b], Alexander Maedche[a,b], and Karl Werder[a]

[a] Chair of Information Systems IV
[b] Institute for Enterprise Systems
University of Mannheim
L 15, 1-6
68131 Mannheim
Germany
{brhel, meth, maedche, werder}@es.uni-mannheim.de

Corresponding Author: Hendrik Meth
Phone +49 621 181-3579 | Fax +49 621 181-3627

1. Introduction

The question of how to meet the challenge of organizing software development activities has concerned researchers and practitioners ever since software engineering (SE) emerged as an independent scientific discipline in the 1970s [1, 2]. In the last decade, the mechanistic view of software development prevalent in earlier phase-based linear approaches has been replaced by an understanding of development activities as a dynamic process characterized by iterative cycles and the active involvement of all stakeholders [3]. This is reflected in *agile software development* (ASD), which responds to unpredictable change by relying on people and their creativity rather than on processes [4], and limit software development strictly to activities that add business value for the customer [5–7].

With agile methods becoming mainstream even for large-scale organizations in the software industry [8,9], software is being delivered on time and in budget, and customer demands are being met increasingly often [10,11]. Nevertheless, agile methods focus on the question of how *useful* software can be developed, with customer value being understood as primarily driven by providing an appropriate functional scope. They do not necessarily focus on developing software that is considered *usable* [12–14], i.e. usability defined as the extent to which a software can be used by specified users to achieve specified goals effectively, efficiently, and satisfactorily in a specified use context [15]. While usability is not a central topic in SE, in which it is considered one of many non-functional requirements and quality attributes [16], it has become crucial for economic success in highly competitive markets and can be used to set the product apart from that of the competition [17–19]. *User-centered design* (UCD) ensures that the goals and needs of the system's end-users is the focus of the product's development. In the field of human-computer interaction (HCI), terms like UCD, usability engineering, and interaction design often have a very similar meaning [20]. UCD is driven by continuous end-user evaluation and the iterative refinement of design concepts and prototypes [21].

Given the need to deliver business value to the customer in a rapidly changing environment while taking the needs of end-users into account, the integration of UCD and ASD seems to be a promising endeavor and has received increasing attention in recent years [20,22,23]. In contrast to plan-driven SE, whose properties often impede the integration of UCD [25,26], similarities between ASD and UCD provide a common ground, which eases integration. Moreover, owing to these similarities, a multitude of integration approaches has been suggested in the literature [20,21,24]. Thereby, some authors focus on the benefits of integrating particular usability practices into ASD [23, 47] or vice versa, while other authors propose integrated *user-centered agile software development* (UCASD) approaches [20,22].

Several literature reviews on UCASD research have been conducted. Although we have identified a plethora of existing works in these reviews, they suffer from several shortcomings: First, existing reviews lack a comprehensive coverage of the different dimensions of UCASD. They mainly focus on the actual software development practices to be used, and rarely describe further dimensions, such as the overall process to be followed, the people and social aspects involved or the technology that may be leveraged. Second, findings have been insufficiently abstracted and systematized, impeding a generalization of the results and applicability in specific contexts. For example, while it might be helpful to know the most elaborated practice to conduct end-user evaluations in a usability lab, this specific practice might be inappropriate when an according infrastructure is not available. In these cases, a more generic principle, i.e. a rule based on the examination of underlying concepts [6], might be a more suitable starting point. Finally, from a methodological point of view, the conducted literature reviews on UCASD lack clear quality criteria for paper selection and need to be complemented by current findings. In order to address these research gaps, this paper aims at capturing and analyzing the current state of the art in UCASD. More specifically, we investigate the following research question: *Which principles constitute a user-centered agile software development approach?* Thus, following an approach similar to the *agile manifesto* [134], we derive a set of grounded principles for UCASD from the literature.

The remainder of the paper is organized as follows: In Section 2, we establish the foundations of our work by summarizing related work and outlining the existing research gap. Section 3 introduces the research method applied in this paper, describing the sourcing and search strategy, the paper selection process, and the final analysis. In the following sections, we present and discuss the results of the systematic review: First, Section 4 presents an overview of the results. The identified principles of UCASD are subsequently discussed in Section 5. Finally, Section 6 provides a summary of the paper and outlines the limitations and contributions of our work.

2. Foundations

ASD and UCD evolved from different motivations. Software engineers aim to satisfy customers through timely releases and responsiveness to change requests without compromising software quality. These goals are difficult to achieve through plan-driven SE approaches, resulting in proposals for ASD [27, 28]. UCD aims at ensuring appropriate usability of the implemented software, a characteristic that has not been considered sufficiently in traditional, plan-driven approaches or in agile approaches. UCD addresses this issue but does not consider agile principles. Therefore, first attempts to integrate ASD and UCD approaches were made about a decade ago. In the following, we present and analyze existing literature reviews on UCASD. We subsequently introduce a comprehensive framework of UCASD dimensions.

2.1. Summary of Existing Literature Reviews

Sohaib and Khan [23], da Silva *et al.* [20], and Barksdale and McCrickard [22] provide comprehensive overviews of approaches to UCASD. The reviews, which are briefly summarized in the following, each have a unique focus and pursue different objectives.

Sohaib and Khan's [23] review aims to identify tension fields between UCD and ASD. They find that the most important questions are whether software development should focus on the customer or on the end-user of the software, whether developers' aim should be to create useful or usable software, and whether unit testing or usability testing is more important. They provide references that highlight different aspects of these questions. Furthermore, they address discrepancies between the use of little and extensive up-front design. Based on their findings, Sohaib and Khan [23] suggest an integrated approach, bridging the gap between the two fields. The approach focuses on the recommendation to use specific practices. In their review, Sohaib and Khan [23] do not attempt to give an exhaustive overview of extant publications on UCASD. Instead, they use references to illustrate each of the identified tension fields. Consequently, they describe neither the search process nor the criteria for inclusion or exclusion of primary sources.

In contrast, da Silva *et al.* [20] explicitly address their work as a systematic literature review and follow the research method described by Kitchenham [29] to cover existing literature on the integration of UCD and ASD as completely as possible. Following Kitchenham's [29] recommendation, they explicitly define their inclusion criteria for publications. Moreover, having conducted an automated search in four databases and a manual search in three conference proceedings, their search scope is extensive. The systematic approach resulted in the analysis of 58 articles, whereby they provide extensive descriptive statistics on various aspects of the retrieved publications. Furthermore, Silva *et al.* [20] identify key aspects of UCASD, again focusing only on relevant practices. They argue that the most significant principles for integrating UCD and ASD are "little design up front" (LDUF) – i.e. the design of a small proportion of the overall system prior to the start of system development – a close collaboration between usability and development experts, and a deferred development with the designers working one sprint ahead of the developers. Based on their findings, Silva *et al.* [20] propose a high-level integrated framework with parallel design and development activities. Additionally, the most commonly recommended practices when integrating UCD into ASD are identified and mapped to their proposed framework.

The main purpose of Barksdale and McCrickard's [22] literature review is to point out an insufficiently covered area of research, highlighting the need for their own research project while placing it in a broader context. Thus, the main contribution of their review lies in the organization of extant research efforts concerning the integration of UCD and ASD along five dimensions. Barksdale and McCrickard [22] thus distinguish between a focus on practices, the process itself, technological means to support software development, and the people, or social aspects, involved in the process. They adopt a broad view of practices, referring to "the whole art of performance" [141], covering all actions of performing or actually doing something within the software development process. We followed this generic classification approach. Consequently, we assigned all included publications with a clear focus to one of these research areas. Furthermore, we provide illustrative examples of each integration strategy, i.e. depicting the relationships of important concepts from both fields. Barksdale and McCrickard [22] indicate the criteria they used to select studies, but solely drew on Google Scholar to conduct an automated search.

2.2. Gap Analysis of Existing Literature Reviews

Analyzing the related work, we observe three major shortcomings: First, none of the existing reviews broadly cover UCASD across its different dimensions, such as practices, process, technology, people, and social aspects. While the works of Sohaib and Khan [23] and da Silva *et al.* [20] focus strongly on practices, Barksdale and McCrickard [22] are most interested in the people/social dimension. Therefore, none of the reviews that cover all relevant dimensions of UCASD in an integrated manner.

Second, the existing reviews do not sufficiently abstract and generalize the results. While recommendations for specific practices are given, these practices are not sufficiently abstracted. As the applicability of specific methods and practices depends on the given context, it is difficult to implement the proposed UCASD approaches in a real-life software development project. A further analysis of existing works, aiming at a clear separation of generic principles and their implementation through specific instantiations would increase the applicability of the findings.

Third, while each review provides summarized information on the included primary articles, none of them includes a quality assessment of the encompassed primary studies, as Kitchenham [29] recommends. In an attempt to update the findings of prior reviews, a quality analysis of the used sources should be included, and the results of this analysis should be taken into account when interpreting the results. Finally, the systematic literature review by da Silva *et al.* [20], which can be considered the most extensive and rigorous review to date, was conducted in 2010. Given the dynamics of the analyzed field of research, major contributions might have been published since then, which was also confirmed in a preliminary assessment of recent publications in the field of interest.

Summarizing, the motivation for conducting a systematic literature review tailored to the scope of the envisioned UCASD approach is sufficiently substantiated. Still, the findings of the three assessed reviews have to be included in this work, and a crosscheck of the retrieved literature with the literature contained in the three published reviews should be performed.

3. Research Method

The literature review was carried out by following the established guidelines for conducting systematic literature reviews suggested by Kitchenham [29], which are a proven means to arrive at a complete and thorough census of existing research within a domain [30–32]. Before conducting the actual review, it was planned in detail by establishing a review protocol. The review protocol guided each of the following steps in detail in order to ensure rigor and transparency in the research process [33, 34]. It was based on the objective of finding guidance and recommendations for UCASD, which reflects the research question introduced earlier (see Section 1), and included the following elements:

(1) A *search strategy*, i.e. the approach to identify appropriate search terms for querying scientific databases as well as the resources to be searched.

(2) *Selection criteria and procedures,* which are suitable to identify relevant primary studies in the full list of results emerging from the automated database search. In the

selection process, the sample of included studies is narrowed down in multiple stages based on previously defined inclusion and exclusion criteria.

(3) The attributes used for a *quality analysis* of the included primary studies, which allows an assessment of the existing knowledge base.

(4) A *data extraction and synthesis* method, representing the systematic approach to consolidate and integrate existing knowledge.

The review protocol was developed in cooperation with the first and second authors, and validated by the third author prior to conducting the review. In the following, we describe the implementation of the review protocol.

3.1. Data Sources and Search Strategy

Contributions to the research topic at hand may be found in different domains (i.e. information systems [IS], and computer science [CS]), and in different sub-domains within these domains (e.g. HCI). For each of these domains, the most relevant databases were selected for the search. The IS literature has focused on three of these databases, namely ProQuest, Elsevier ScienceDirect, and EBSCO Host. Additionally, we queried three databases, namely IEEE Xplore, ACM Digital Library, and Springer Link, which mainly focus CS topics. The search string was composed to cover both fields of interest, i.e. UCD and ASD. Key words relevant to the first field were derived from an exploratory literature review, while key words for the second field were adopted from Dybå and Dingsøyr [31], who performed a systematic literature review of empirical studies on ASD. Additionally, the terms “software development” and “systems development” were added to reflect the activity of interest and restrict the search space. Table 1 lists the key words and their corresponding logical operators.

AND	OR	Ergonomics, Human-Computer Interaction, Computer-Human Interaction, Interaction Design, Usability, User Experience, User-Centered Design, UI Design, Interface Design
	OR	Agile, Scrum, Extreme Programming, Lean, Crystal Clear, Feature Driven Development, Dynamic Software Development
	OR	Software Development, Systems Development

Table 1: Composition of the search string

3.2. Paper Selection and Pre-Assessment

The selection process involved four stages. In each stage, the sample size was reduced based on the inclusion criteria applicable in this stage as depicted in Figure 1. In the first stage, relevant publications were retrieved by querying the databases mentioned above with the search string depicted in Table 1. All database queries were made in the first two weeks of October 2012. This yielded a total of 1152 initial results, and 1034 publications after the removal of duplicates, which were included in the second stage.

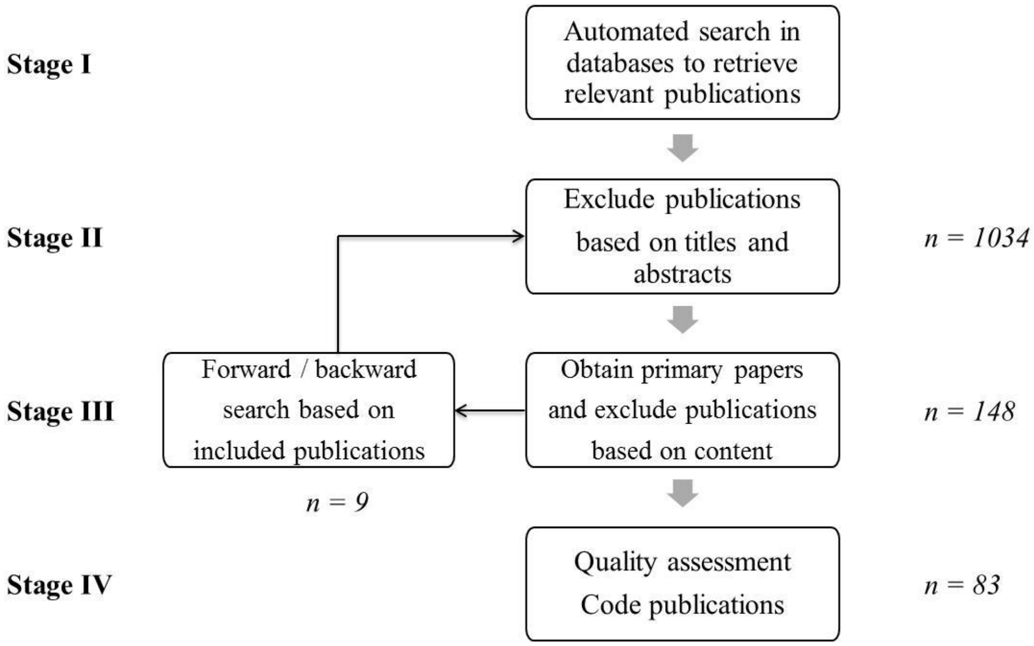

Figure 1: Selection process (based on [31]).

In the second stage, publications were excluded based on their titles and abstract. Criteria for inclusion in the following stage were that the article focuses on the integration of UCD and ASD. Special formats, such as editorials, prefaces, article summaries, interviews, news, correspondence, discussions, and readers' letters, were excluded. Moreover, we excluded articles summarizing tutorials, workshops, panels, or poster sessions. As Brereton *et al.* [30] note, titles and abstracts of IS, CS, and SE publications are often of low quality. Thus, articles were included in the next stage only if the evaluation of titles and abstract did not lead to a conclusive decision as Dybå and Dingsøyr [31] demonstrate. After the analysis of titles and abstracts, 148 publications were included in the following stage.

The third stage involved a more detailed analysis of the articles' content. Thus, the full text of each publication in this stage was retrieved for local storage. Each article in the sample's introduction, conclusion, and, in case of doubt, content was skimmed to assess its relevance. In this stage, stricter criteria were applied to determine the relevance of articles, as studies or practitioner reports should give recommendations or requirements for software development integrating UCD and ASD. These recommendations or requirements may be given explicitly, for example: "UCD practitioners must be given ample time in order to discover the basic needs of their users before any code gets released into the shared coding environment" [35]. On the other hand, they might emerge as a result of the study, for example, if a positive impact of applying a certain principle or practice has been observed. Based on these criteria, 74 articles were considered relevant for inclusion in the next stage. Following the strategy suggested by Webster and Watson [36], we conducted a forward and backward search based on these key publications. The retrieved new and potentially relevant articles were fed to the second stage for further processing, resulting in an iterative extension of the sample. This resulted in an additional set of nine articles for inclusion in stage four. It has to be noted that both da Silva *et al.'s* [20] and Sohaib and Khan's [23] reviews were included in the stage three sample. However, they were excluded from the detailed analysis in stage four as they do not present the results of primary research. In total, 83 papers were selected for the final sample[1].

In the fourth stage, the final sample passed a first categorization and a quality assessment. During the categorization, each paper was assigned to one of the four research foci, which

[1] The full list of papers and their corresponding classifications can be accessed at https://madata.bib.uni-mannheim.de/80/

will be introduced in the next section. A quality assessment of each paper was subsequently conducted to obtain additional information supporting the interpretation of the paper's recommendations. This assessment was based on four questions, which are given in Table 2. Each of the four criteria was graded on a dichotomous scale as "yes" or "no," while question 3b was answered by means of the respective method, which was either explicitly stated in the paper or determined by the researcher based on the available information.

1. Is there a clear statement of the research goals (e.g. in an explicitly verbalized research question)?
2. Is there an adequate description of the context in which the research was carried out?
3. (Only applicable to empirical research papers):
    a. Is the research method explicitly stated?
    b. Which research method was chosen?
4. Are there explicit recommendations, which could be aggregated as principles?

Table 2: Criteria for quality assessment.

3.3. Paper Analysis

Identification of Dimensions

To analyze prior research, a comprehensive, hierarchical coding system was established. UCASD is essentially about the integration of UCD and ASD. Therefore, to identify the basic dimensions of the coding system, it is helpful to investigate on which levels integration may be accomplished. Barksdale and McCrickard [22] present an approach, in which the existing literature is classified into five types of integration for UCASD: Process integration, practices integration, people integration, social integration, and technology integration. *Process integration* is understood as the merging and synchronizing of independent UCD and ASD processes, providing a unified process incorporating both perspectives. *Practices integration* represents the incorporation of UCD practices into ASD and vice versa. *People integration* is understood as changes in the team composition to bring experts of the two different disciplines together (e.g. adding a designer to a team of developers). *Social integration* reflects social interaction and the joint creation of knowledge. Finally, *technology integration* entails the use of technological means to support and coordinate activities.

The different levels of integration presented by Barksdale and McCrickard [22] allow for a comprehensive and differentiated classification of existing works on UCASD. In our literature analysis, we therefore used these integration levels as a starting point to determine the basic dimensions of our coding system. In contrast to Barksdale and McCrickard [22], we did not differentiate between people integration and social integration, as we consider the two aspects almost inseparable.

Identification of Codes

We used the qualitative data analysis software *MAXQDA*[2] to code the papers. As a universal tool for qualitative data analysis, *MAXQDA* is usually employed to analyze textual data, such as interview transcripts, and supports a variety of qualitative research methods. Various authors (e.g. [31,37,38]) have reported positive experiences concerning the use of similar software (e.g. NVivo[3]) to synthesize data from a corpus of texts emerging from a systematic literature review. Motivated by these experiences, we adopted this approach to process a large amount of textual data rigorously and transparently. This seems especially useful as evidence found in extant literature, which may be used to derive principles, can be retrieved easily in later stages of the research process. Using *MAXQDA*, codes can be

[2] http://www.maxqda.de/
[3] http://www.qsrinternational.com/products_nvivo.aspx

assigned to text segments or images in documents. Besides descriptive statistics on the occurrence of codes in documents, *MAXQDA* provides a convenient way to extract coded text passages, allowing for easy data synthesis. Moreover, MAXQDA allowed us to conduct a quality analysis of the included papers. To be able to retrieve papers within a certain category in a convenient manner, the categorization was done by assigning a code reflecting the appropriate category to the title and abstract of each paper. Furthermore, if a criterion given in Table 2 was assessed to be fulfilled, the text segment providing the information was coded accordingly. The analysis of the document corpus was aimed at identifying aspects that have to be taken into account when integrating UCD and ASD, with the overarching aim of deriving principles. During the analysis of the document corpus, observations made concerning different aspects of integration were assigned a meaningful code, with the aim of consolidating the perspective different authors take on the same issue later on.

The dimensions introduced above formed the basis of a first set of high-level codes. For simplification reasons, we use the terms "process," "practices," "people/social," and "technology" as high-level codes in the following. Based on our analysis and comparison of the existing studies, as presented in Section 2, we assigned initial sub-codes to each high-level code (see Table 3). Each of the papers in the final sample was carefully read, analyzed, and coded according to this system. While the initial set of codes proved to be exhaustive, we added additional lower level codes and hierarchy levels during the analysis of the literature to document the insights reported in these studies. This process becomes omnipresent when comparing the two coding systems. The initial coding system reflects 21 codes with two hierarchy levels, whereas the final coding system contains 73 codes in four hierarchy levels. Various reasons, such as the addition of specific practices as codes, drove the extension of the coding system. During the coding process, text passages were related to one of the codes. While the group assignment according to the main research focus was mutually exclusive, this does not mean that only aspects concerning this stream of research were discussed in the paper. For example, when the authors of a paper with a clear process focus recommend the use of personas, the corresponding text passage was coded as "personas," which was applied to every document in the corpus discussing personas independently of its research focus. Moreover, throughout the coding process, potential new codes were challenged against existing codes and vice versa. Initially, this led to many changes in the coding system. However, the codes became more stable as they developed organically.

Level 1	**Process**	**Practices**	**People/Social**	**Technology**
Level 2	Little Design Up Front	Prototyping	Close Collaboration	Data Exchange
	User Testing*	User Testing*	Cross-Functionality	IDE Integration
	One Sprint Ahead	User Stories	Knowledge Transfer	
	Cohesive Overall Design	Scenarios		
	Parallel Tracks	Personas		
	Iterative Design / Development			
	Incremental Design / Development			

Table 3: Initial coding system.

Identification of Candidate Principles

In order to identify candidate principles, each code was investigated concerning two characteristics. First, the *frequency of occurrence* was assessed to gain an impression of the relative importance of each aspect. As an example, the code "prototyping" was assigned in 40 (48.2%) articles in the literature review, indicating that the use of prototypes was discussed in this document. In contrast, the idea of “remote usability testing” was only found in one (1.2%) article, leading to the conclusion that, while the former aspect has a high significance for the scientific community, the latter is a marginal topic. For this assessment, we also considered the insights reported in existing literature reviews (cf. Section 2.1).

Second, we collected *guidance and recommendations* for the conceptualization of an integrated UCASD approach. While 17 (20.5%) of the articles included in the review contain explicit advice on how to integrate UCD and ASD, implicit recommendations can be derived too if applying a certain practice has been observed as having a positive impact. Thus, the content of the coded text passages was analyzed for either explicit or implicit guidance concerning the integration of UCD and ASD. Finally, to obtain an overview of the recommendations in the extant literature, the coded text passages were reviewed once again and organized along emerging patterns. These patterns represent related ideas or concepts presented in different publications along the four dimensions, leading to the merger of independent aspects in the coding system. The identification of such patterns was based on a qualitative analysis and the expertise of the researchers in this domain. In this step, also the findings of the three assessed reviews were drawn upon to identify themes for potential principles [33].

4. Results

4.1. Number and Distribution of Publications

When analyzing the result set, it became apparent that, while the integration of UCD and ASD was first discussed a decade ago, the topic gained momentum in 2007 and, since then, a constantly high number of relevant articles have been published every year, as Figure 3 illustrates. This reflects that, while the idea of integrating UCD and ASD, has been around for some time, many integration issues are still unresolved and research is ongoing. In particular, at least 17 relevant articles have been published since da Silva *et al.’s* [20] systematic literature was conducted in late 2010, justifying an update of their findings.

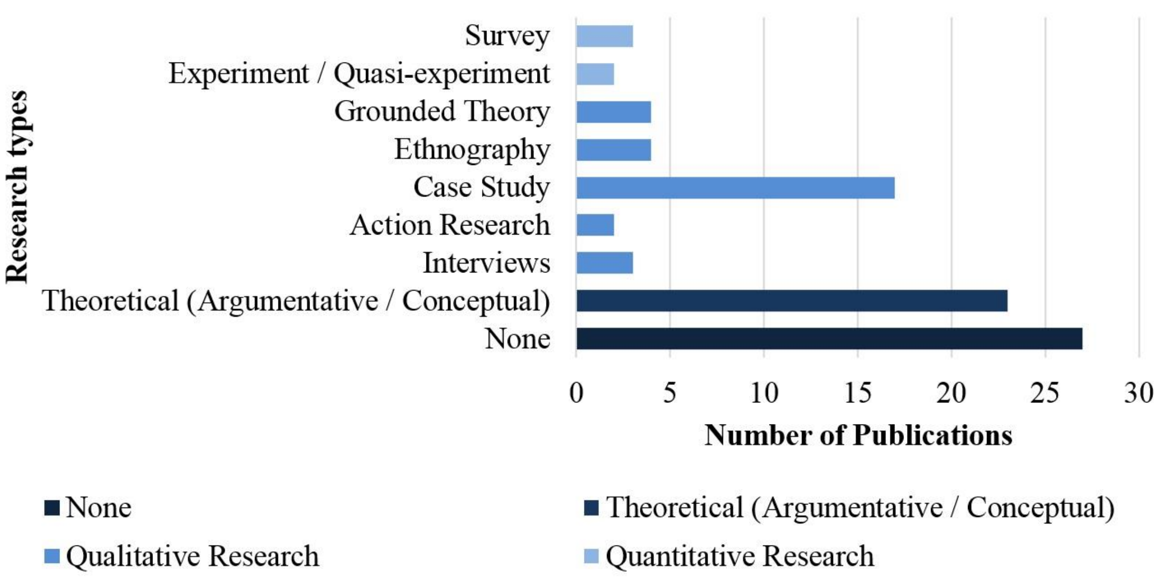

Figure 2: Number of publications by research type.

Moreover, we identified the research type of each paper as presented in Figure 2. We classified each paper according to one of four research types, which are non-exclusive of each other. The types identified are qualitative research (N=30), (i.e. action research (N=2), case study (N=17), ethnography (N=4), grounded theory (N=4) and interview (N=3)), quantitative research (N=4), theoretical work (N=23), and none (N=27) of the aforementioned (e.g. practice reports). Industrial experience reports refer to work that present practical experience. This is supported by existing evidence that industrial experience strongly influences agile research [20, 46].

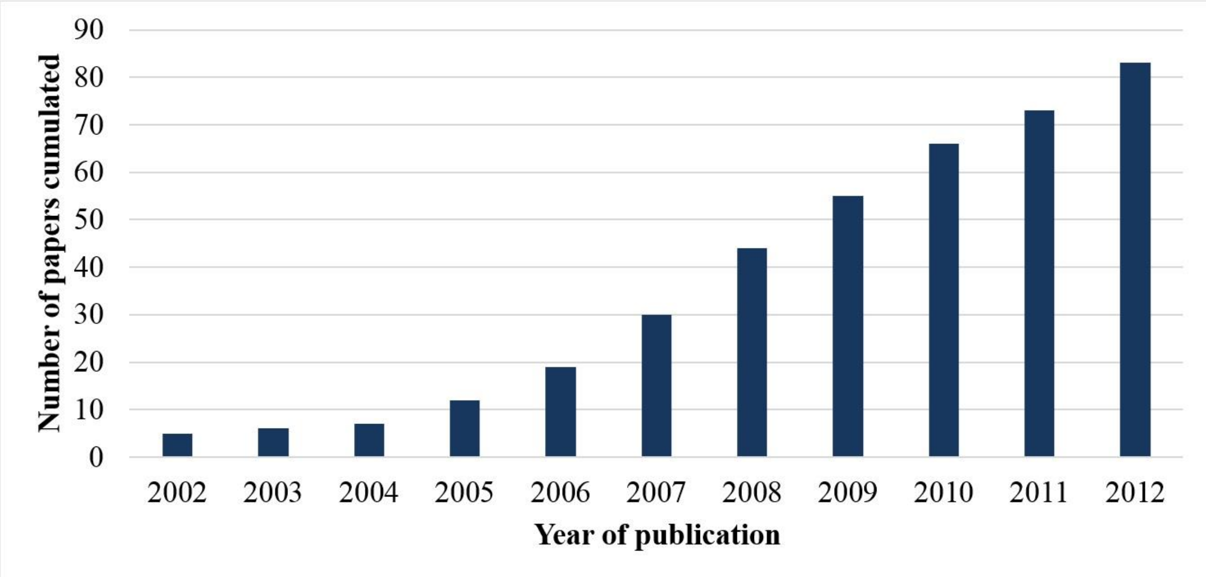

Figure 3: Number of publications by year.

Table 4 presents the results of categorizing the 83 publications included in the final sample according to their main research focus based on the dimension of our coding system. The analysis revealed that both process and practice integration aspects are strongly represented in the final sample. While the former are discussed in 24 (28.9%) publications, the latter form the biggest group with 38 (45.8%) of the included articles. However, the high number of publications discussing process integration aspects does not mean that a variety of mature integrated approaches exists. Among the 24 publications that focus on process integration, only 15 present novel approaches, while the remaining articles discuss only approaches that were published earlier by the same authors. In general, it is difficult to distinguishing between a focus on process or practice integration aspects, as authors frequently describe their work as integrated approaches, but only discuss the integration of UCD into ASD while neglecting the developer’s perspective, which did not meet the definition of the process focus taken here. People and social integration aspects were discussed in 14 (16.9%) papers, while technological integration received the least attention with seven (8.4%) of the included publications looking into it.

Dimension	**Included Publications**	**Number of Publications**
Process	Benigni *et al.* [39], Budwig *et al.* [40], Felker *et al.* [41], Ferreira *et al.* [14], Ferreira *et al.* [42], Fox *et al.* [21], Holzinger *et al.* [43], Hussain *et al.* [44], Hussain *et al.* [45], Kuusinen *et al.* [46], Lee & McCrickard [47], Lee *et al.* [48], Lee *et al.* [49], Losada *et al.* [50], Losada *et al.* [51], Losada *et al.* [52], Memmel *et al.* [53], Memmel *et al.* [54], Miller [55], Najafi & Toyoshiba [56], Paelke & Nebe [57], Paelke & Sester [58], Wolkerstorfer *et al.* [59], Zhang *et al.* [60]	24 (28.9%)
Practices	Adikari *et al.* [61], Beyer *et al.* [62], Broschinsky & Baker [63], Carvalho [64], Chamberlain *et al.* [35], Cho [65], Constantine [13], Constantine & Lockwood [66], da Silva *et al.* [67], Detweiler [68], Düchting *et al.* [69], Evnin & Pries [70], Ferré *et al.* [71], Fisher & Bankston [72], Haikara [73], Hansson *et al.* [74], Hellmann *et al.* [75], Hellmann *et al.* [76], Hennings [77], Hodgetts [78], Humayoun *et al.* [79], Hussain *et al.* [80], Illmensee & Muff [81], Isomursu *et al.* [82], Kane [83], Lárusdóttir [84], Lárusdóttir *et al.* [85], Lee *et al.* [86], Medina-Medina *et al.* [87], Memmel *et al.* [88], Meszaros & Aston [89], Obendorf & Finck [90], Obendorf *et al.* [91], Patton [92], Patton [93], Petrovic & Siegmann [94], Rafla *et al.* [95], Rittenbruch *et al.* [96]	38 (45.8%)
People & Social	Barksdale & McCrickard [22], Barksdale & McCrickard [97], Barksdale *et al.* [98], Brown *et al.* [99], Brown *et al.* [100], Ferreira *et al.* [101], Ferreira *et al.* [102], Kollmann *et al.* [103], Leszek & Courage [104], Lievesley & Yee [105], Singh [106], Ungar [107], Ungar & White [108], Williams & Ferguson [109]	14 (16.9%)
Technology	Feiner & Andrews [110], Gonçalves & Santos [111], Hosseini-Khayat *et al.* [112], Humayoun *et al.* [113], Lee [114], Peixoto [115], Peixoto [116]	7 (8.4%)
Total		83 (100.0%)

Table 4: Overview of publications in the final sample.

4.2. Synthesized View

A key objective of the literature review was the synthesis of evidence found in extant literature in order to derive principles by applying a coding system to each article's content. Figure 4 depicts the final version of the coding system, which evolved for the "process," "people/social," "technology," and "practices" dimensions. Some codes were omitted to improve readability (see complete coding system in Appendix A).

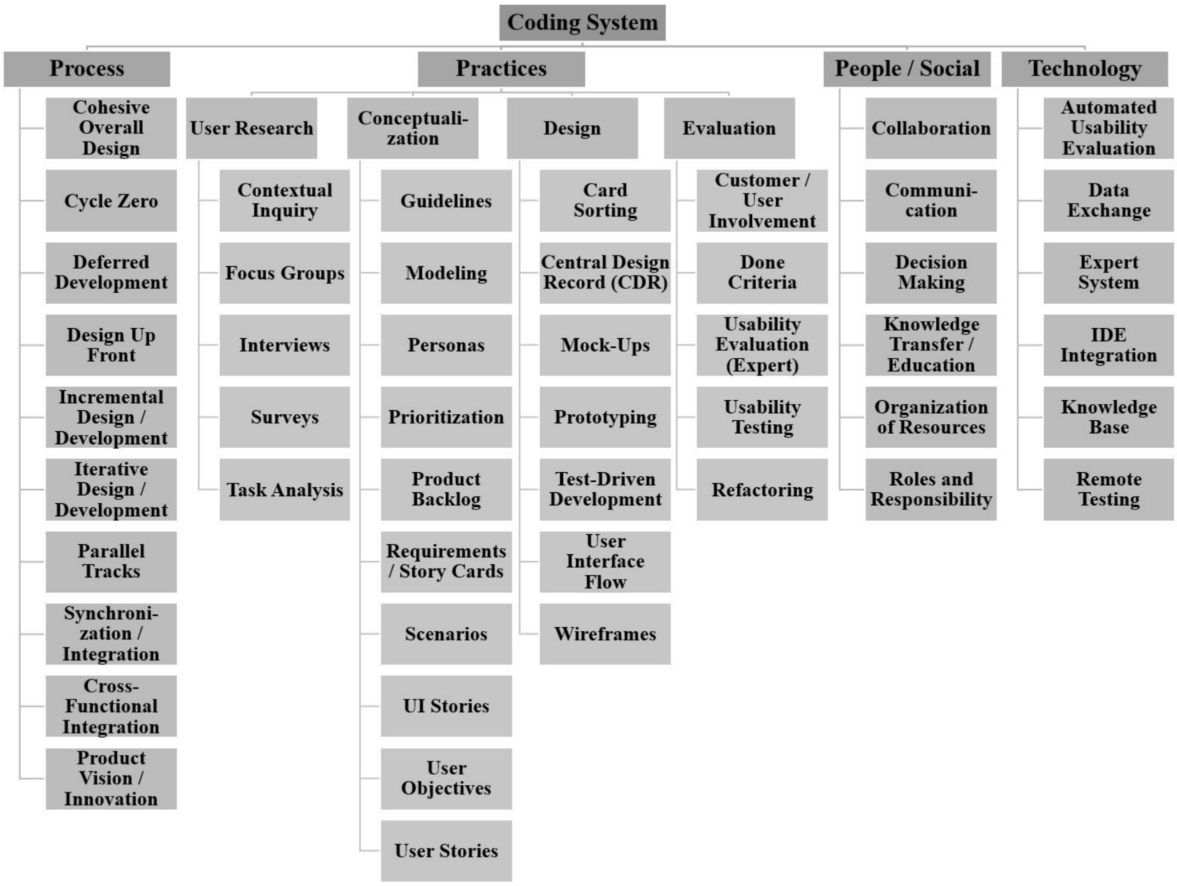

Figure 4: Final coding system for the process, people/social, technology, and practices dimensions.

The coding system shows that, while some codes explicitly emerged from the integration of ASD and UCD approaches (e.g. "synchronization" and "collaboration"), other codes represent aspects of the individual, original approaches (e.g. "iterative design" or "incremental design"). This result illustrates that integrated approaches strive to keep the best aspects of the original approaches on the one hand, while trying to enable a smooth merge of the two worlds on the other hand. Interestingly for the process dimension, many more codes have been identified for the other two dimensions. Moreover, the codes in the process dimension are mainly specific to software development, while the majority of the codes in the other dimensions are rather generic (e.g. codes like "collaboration" or "data exchange"). This might be due to the overall number of papers assigned to the "process" dimension being larger than in the other dimensions. Apart from that, it might also be an indicator of the depth and maturity of discourse in each of the dimensions. While aspects of process integration have been intensively discussed and refined, the discussion in the people/social and technology dimension seems to be rather limited in UCASD research.

Owing to the large variety of practices discussed in the review, we sub-categorized the practices dimension based on the process phases. While strict process phases are uncommon in ASD, UCD proceeds along pre-defined (but not necessarily sequential) phases. Our classification follows the four phases defined in the ISO standard for human-centered design: research, specification, design, and evaluation [117]. The four aforementioned phases are considered generic enough for identifying sub-codes within the practices dimension, One example is the inclusion of user research, which is absent in other development methods. Abstracting these phases from the general activities to be done (investigate the context, specify requirements, design the software, and, finally, test its congruence with the

requirements), this classification also allows for mapping typical agile practices (such as user stories). Figure 4 displays all codes for the practices dimension and their assignment to sub-dimensions.
The results confirm that, especially in UCD, there is a plethora of different methods as others (cf. [118]) have shown. Recommendations for individual practices differ across the studies, resulting in a heterogeneous picture of practices with an overlapping, if not identical, scope. For example, in the specification phase, scenarios and UI stories, as well as mock-ups and wireframes could be used alternatively. As suggested earlier, it is therefore helpful to abstract phases from specific practices.

The final coding system contained in Figure 4 provides a comprehensive overview of aspects relevant to UCASD. Consequently, these codes, respectively the underlying text passages, were the key takeaway from the literature review, whose overarching aim is to derive principles to guide the integrated UCASD approach. Throughout the data gathering process, we identified a lack of content for the technology dimensions. While we found aspects concerning technological means to support user-centered agile processes (N=10), the text passages were of a general nature and did not provide guidance on technology support. Therefore, no principles concerning technology integration were derived. A different challenge emerged for the people/social dimension. Although we found more sources, the recommendations were contradictory. While ASD suggests the collective accountability of the team [119], UCD methodologies usually suggest individual responsibility, for example, allocating the responsibility of developing a usable product to the UCD expert [122-124]. Furthermore, the team setup led to contradictory evidence. On the one hand, two *dedicated teams*, which handle design and development activities respectively, can be formed. On the other hand, a *cross-functional team*, handling both tasks, might be established [101,102]. Therefore, we derived no principles for the people/social dimension. The paper's conclusion discusses how future work might derive corresponding principles for the technology and people/social dimensions.

5. Principles of UCASD

In the following, we discuss the identified principles for the integration of UCD and ASD organized along the two central dimensions processes and practices.

5.1. Process Principles

First, a process perspective is necessary to decide how activities involved in the software development process are organized on a fundamental level. Various aspects focusing on the process perspective were frequently mentioned in the identified publications and appeared to have a high level of maturity as a consensus was identified for most of them. In Figure 5, the codes assigned to the process perspective are listed along with their occurrence in the identified publications. The items have been consolidated into three key principles forming a UCASD approach.

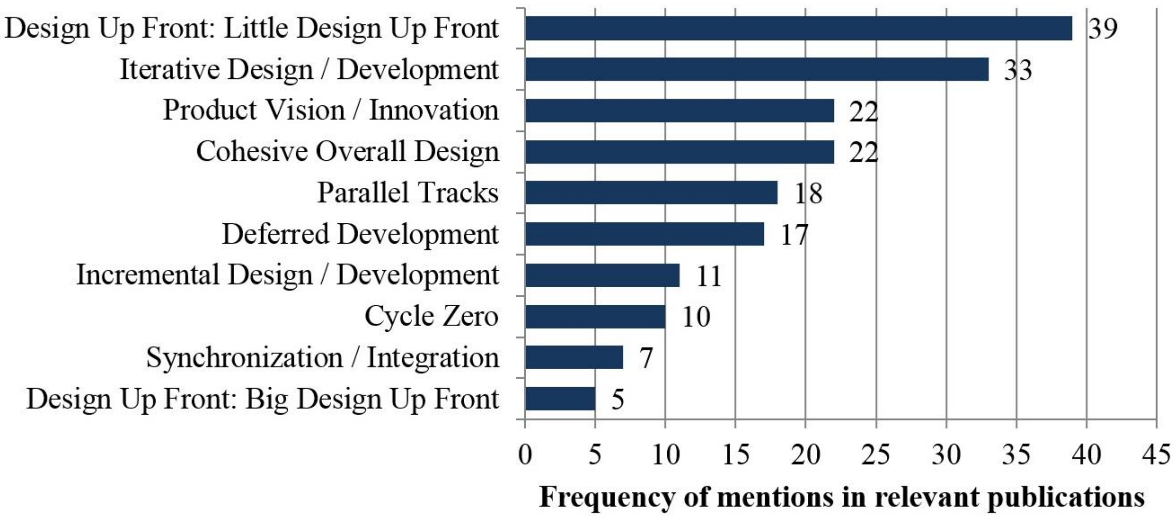

Figure 5: Codes and articles related to the process dimension

Separate Product Discovery and Product Creation

Focus Area. The valuation of up-front analysis and design was recognized as one of the main tension points between UCD and ASD early on. While the former promotes the extensive up-front analysis of user requirements and the design of the users' interaction with the system, the latter focuses on delivering a working code at the expense of an exhaustive planning and design phase. Moreover, there is no common definition of what constitutes sufficient preparation, and the effort put into initial analysis and design activities ranges from days (e.g. [13]) to weeks (e.g. [40]).

Evidence. The results of our literature review confirm the importance of an up-front analysis and design: 42 (50.6%) of the included publications discuss the necessity of up-front design efforts. Seven of these present empirical evidence. A total of 39 articles argue that, while an extensive up-front design is against agile principles, *little design up front* (LDUF) is needed for successful UCD efforts in an agile environment. As an example, Miller [55] and Sy [24] suggest that interaction designers use an up-front cycle for planning and collecting customer input. Fox *et al.* [21] are more specific with regard to methods and reserve an up-front phase for contextual inquiry and iterative low-fidelity prototyping, resulting in a preliminary design and a list of requirements that are handed over for development. This is in line with the results of da Silva *et al.* [20], who find that 31 of the 58 papers they reviewed recommend a strictly limited up-front design effort.

In particular, 10 (12.0%) of the included publications implement the LDUF concept by reserving a *cycle zero*, or sprint zero, for analysis and design activities before actual agile development iterations start. The idea of time-boxing initial analysis and design for the duration of a development cycle was first suggested in Miller's [55] integrated process model and has since been adopted and modified by various authors, such as Sy [24], Fox *et al.* [21], and da Silva *et al.* [20].

There is evidence that a limited up-front design effort is particularly necessary to provide a consistent and cohesive user interface and navigation structure as it supports designers in finding a suitable design concept from the very beginning [42,61]. Patton [93] states that user needs and goals have to be clearly defined before conducting design and development efforts in order to ensure the usability and usefulness of the software in ASD. Various authors have confirmed this view, presenting user research as an essential aspect of up-front design efforts in agile environments [20,21,24,35,49,55]. In total, 22 (26.5%) of the papers in the review confirm a relationship between LDUF and a *cohesive overall design* of the final product, which is challenging to achieve in an agile environment. Meszaros and Aston [89] assert that evolutionary design is not suitable for user interfaces when insufficient conceptual guidance is given. Moreover, Constantine [13] argues that late revisions of the user interface are unacceptable as such revisions would be highly disruptive for users already familiar with the application.

While there is no broad agreement in the literature on the extent and outcome of LDUF activities, it is evident that what is commonly called up-front "design" also has the character of a constituting analysis involving extensive stakeholder interactions. Other than the actual UI design itself, user research does not yet involve any commitment, thus maintaining flexibility. Consequently, up-front design activities should first and foremost be understood as a preliminary exploration phase, harnessing the potential of UCD to generate new ideas. Twenty-two (26.5%) of the publications included in the review address this issue, and see UCD as a means to establish and communicate a *product vision* in order to "improve the line of sight between strategy and team-level execution" [72]. While UCD provides suitable concepts for approaching ill-defined problems and delivering a product with a high degree of innovation, exploratory activities are insufficiently addressed in ASD [125]. Kettunen [119] points out that, while agile methodologies are useful means to iteratively select and refine product features, large-scale product innovations are beyond their scope [119]. They answer the question of how to build a product correctly, i.e. delivering a valuable product to the market as soon as possible, while maintaining flexibility along the way. However, product scoping or ideation in general, which would address the question of building the right product, is not part of agile methodologies. This shortfall can be countervailed by drawing on UCD, in which collecting stakeholder needs, expectations, and ideas are core functions. We consequently conclude that the usefulness focus of SE has to be complemented by usability concerns not only during product development, but also during product planning. In order to deliver a consistent and cohesive design throughout software product development, such concerns should be included in the product vision [40,42,46,72,89,103,105,106].

Suggestion. A shift from an up-front design to up-front analysis concept is identified. In order to deliver a product that is valuable to the customer and end-user in that it is both usable and useful, product discovery and product creation should be separated in UCASD. Overall, as listed in Table 5, four codes from our coding system support our suggestion to include an additional principle in this specific context.

Included Codes	*Resulting Principle*
Little Design Up Front Cycle Zero Cohesive Overall Design Product Vision / Innovation	Separate Product Discovery and Product Creation

Table 5: Codes related to the separate product discovery and product creation principle

Besides making room for sufficient up-front design activities as a prerequisite for delivering a usable product with a consistent user interaction concept, it allows for mitigating agile shortcomings with regard to product discovery, fostering the delivery of a useful product. Thus, we formulate the first principle as:

> ***Principle 1 (Separate Product Discovery and Product Creation):*** *User-centered agile software development should be based on separated product discovery and product creation phases*

While the literature clearly evidences the importance of product discovery in general, there is currently a knowledge gap with regard to the extent of product discovery activities. This is also visible within the ten listed items identified in the publications, in which a pair of antipodes can be found, namely the contrast between little or extensive up-front design efforts. Thus, further empirical research is required to investigate the extent of product discovery activities in respect of the different contingency factors on the team, product, and organizational levels. For example, it would be of interest to understand how different software product characteristics, such as the maturity, complexity, and degree of interaction, influence the extent and outcome of product discovery activities.

Iterative and Incremental Design and Development

Focus Area. Both ASD and UCD promote an iterative and incremental approach to software development. This allows for using feedback collected in previous iterations to enhance the emerging product in future iterations. Thus, intermediate solutions to the development problem are used as a pathway to finding a complete and desirable solution for the customer or end-user.

Evidence. The main commonalities of ASD and UCD have been identified in earlier literature reviews by da Silva *et al.* [20], and Sohaib and Khan [23]. Specifically, their reviews confirm the significance of iterative and incremental design and development. There is general agreement among UCD researchers that an *iterative* approach is essential for developing a product with high usability [123,126]. According to Mao *et al.* [118], design iterations are the most commonly used UCD paradigm; however, feedback is also an essential aspect in ASD. In ASD, the overall project is typically composed of subsequent self-contained cycles, each of which comprises analysis, design, programming, and test activities [127]. Among the included papers, 33 (39.8%) explicitly discuss design or development iterations. In accordance with UCD approaches, empirical feedback is used to revise designs in the next iteration. The iterative refinement and evaluation is continued until the user's needs are met, thus ensuring that an adequate level of usability is achieved [124,128]. In particular, all publications presented in the process dimension (see Table 3) promote an iterative process setup.

An iterative approach allows for the product under development to be refined based on feedback collected at an earlier stage. In contrast, an *incremental* strategy means that system functionality is partitioned into thin vertical slices from the user interface to the database layer. Each slice is functionally coherent and demonstrable. Instead of delivering the required functionality all at once, the scope of the system grows progressively with the addition of consecutive slices [129,130]. Solutions to software development problems are often complex and exhibit many degrees of freedom. Thus, eliciting requirements from users often results in what Boehm [131] calls the IKIWISI symptom: "I can't tell you in advance, but I'll know it when I see it" [131]. Especially for interaction features, users might not be able to specify their needs unless confronted with a tangible representation of the system, such as a working product increment that would allow them to clarify and refine their vision of the software [9,127,131]. Drawing on this insight, the aspect of incremental development was mentioned in 11 (13.3%) articles, typically with reference to functional tests or usability evaluations.

The associated papers identified in the literature provide no empirical evidence for the outcome of iterative and incremental design and development. This may be because the empirical studies not only consider the user-centered design or agile development perspective independently from each other, but as a combination of the two [21,35,42,46,49,103,105].

Suggestion. In summary, developing software in an iterative manner creates short feedback cycles, while an incremental development in vertical slices allows for evaluating working product increments. Additionally, adapting the product scope after each iteration embraces change and is a prerequisite to deliver valuable software.

Included Codes	*Resulting Principle*
Iterative Design / Development Incremental Design / Development	Iterative and Incremental Design and Development

Table 6: Codes related to the iterative and incremental design and development principle

Beyond the frequent occurrence of the two aspects in the literature, a general acceptance of this strategy can be inferred from references to iterative and incremental concepts for ASD (e.g. Scrum) or UCD (e.g. ISO 9241-210). Even though the reviewed papers do not explicitly discuss an iterative and incremental approach, none of them challenge this paradigm. Thus, the second principle is articulated as follows:

> ***Principle 2 (Iterative and Incremental Design and Development):*** *User-centered agile approaches should support software design and development in short iterations and in an incremental manner.*

Parallel Interwoven Creation Tracks

Focus Area. Extending the separate product discovery and product creation principle, the third principle shapes the subsequent course of action after the start of regular design and development activities. As a restricted up-front phase of analysis and design allows for specifying the user interaction only for the most important features of a system, features with a lower priority have to be considered in later iterations in parallel to development. Thus, it is necessary to conduct user research and prepare designs for the upcoming development cycle at least one iteration (or sprint) ahead of the development team.

Evidence. In total, 18 (21.7%) of the publications included in the literature review support this principles, whereas five are empirically grounded. On the one hand, the literature explicitly combines the LDUF concept with the organization of continuous analysis and design activities. On the other hand, development activities need to be organized in *parallel tracks*. The literature offers no alternative approach for the latter. The incremental nature of ASD enables UCD experts to prepare the interaction concept for envisioned system features consecutively according to their priority. Thus, an initial interaction concept for the most important system features can be prepared in an upstream iteration prior to development. Enhancing this concept in parallel to development allows for maintaining a suitable balance between necessary up-front design and flexibility, while designs are prepared for implementation just in time. Additionally, engaging in continuous analysis and design activities in parallel to development allows for incorporating changing user and customer needs, thus leading to an increased degree of software usability and usefulness.

The concept of designing *one sprint ahead* is commonly found in existing UCASD approaches, and was mentioned in 17 (20.5%) publications in the review. However, only two publications support this argument with empirical evidence [49, 56]. As examples, Miller's [55] and Sy's [24] proposals schedule activities for the design team to gather customer data and develop user interface specifications for the next cycle, while the development team simultaneously implements specifications prepared in the previous cycle. Fox *et al.* [21] moreover propose that the development team implements the design concept prepared in the preceding iteration, while the UCD staff conducts a contextual inquiry and prepares a design prototype for the next iteration.

Despite the need to work in parallel, seven (11.86%) of the publications in the review point out that a deferred development necessitates mechanisms for the *synchronization and integration* of design and development work. However, the scheduling of resources in practice is a non-trivial task and can be identified as one constraint. Researchers investigating the topic of *cross-functional integration* have discussed the question of how to integrate the development and design functions since the inception of UCASD more than a decade ago. The potential of cross-functional integration has also been intensively investigated from an empirical point of view in product development research [e.g. 142, 143]. Two specific dimensions of cross-functional integration in the context of UCASD are identifiable in the literature. On the one hand, the necessity of functional diversity on the team level, i.e. the degree to which team members differ with regard to their functional background and experiences [132], results in an organizational challenge to integrate design and development experts. On the other hand, team members with multi-knowledge, individual knowledge, and experience across different functional areas [133] can be important enablers of cross-functional work in software development projects. Concerning the former aspect, the preferable organizational setup for UCASD is yet to be determined. The options are to appoint two dedicated teams to handle design and development activities, respectively, or to establish a cross-functional team that handles both tasks [101,102]. The latter is in line with ASD, which promotes the idea of a team including “people with all the skills and perspectives necessary for the project to succeed” [5]. Schwaber and Sutherland [121] emphasize that an agile team should have “all competencies needed to accomplish the work without depending on others not part of the team” [121]. Similarly, 34 (41.0%) of the papers included in the literature review describe a cross-functional setup, in which UCD expertise is available within an agile development team. According to Kuusinen *et al.* [46], this is also practitioners’ most frequent proposal to improve the cooperation between UCD experts and developers. As it is frequently mentioned, a cross-functional team allows for a direct and unmediated exchange of knowledge and information between design and development experts [22,100,102]. In particular, it allows for maximizing collaborative activities in order to create possibilities for knowledge transfers. As an example, Ungar and White [108] suggest workshops in which designers, developers, and stakeholders collaborate and explore design alternatives as a forum for ad-hoc knowledge exchanges between the involved parties.

Suggestion. Design and development activities as part of the product creation phase are often described as parallel tracks. It has to be kept in mind that an interaction concept destined for implementation in an upcoming iteration has to be finished prior to the start of its development. Table 7 lists the codes that built the foundation for suggesting one further principle, i.e. parallel interwoven creation tracks, from a process perspective.

Included Codes	*Resulting Principle*
Parallel Tracks Design One Sprint Ahead Synchronization / Integration	Parallel Interwoven Creation Tracks

Table 7: Codes related to the parallel interwoven creation tracks principle

To summarize, design activities need to start one sprint ahead of the development activities. Given the parallelization, mechanisms need to be adopted to assure the synchronization and integration of such tracks.

> ***Principle 3 (Parallel Interwoven Creation Tracks):*** *In user-centered agile approaches, design and development should proceed in parallel interwoven tracks.*

Further empirical research is required to investigate different cross-functional UCASD creation track setups under consideration of contingency factors. Existing literature providing empirical insights into cross-functional integration in product development might serve as a useful starting point. Such insights would require further refinement to focus on UCASD and its outcomes on a team and product level.

5.2. Practices for Principles

The practices perspective considers concrete methods that are executed within the coding system. The literature included in our review discusses a large variety of practices (see Figure 8 in Appendix A). In an effort to consolidate this diversity of practices, principles for the practices dimension of UCASD should capture the rationale behind these practices rather than pay attention to the practices themselves. Consequently, practices conducted in an UCASD approach have been classified along two holistic principles.

Continuous Stakeholder Involvement

Focus Area. While ASD is inherently people-centric and the active involvement of stakeholders is one of its essential elements [3,135], it lacks a clear distinction between the customer and the end-user of the software [35]. It is not unusual for one person, for example a domain expert or a product manager from the client organization, to fill both the customer and user roles [136]. Such roles are not able to represent all stakeholders in the system [62]. However, UCD approaches require direct and unmediated contact with the end-user of the software [123,126].

Evidence. Various publications in the literature review discuss the decision between direct or mediated stakeholder relations from a general perspective without reference to specific concepts. Thereby, 17 (20.5%) of the included publications promote the *direct involvement* of stakeholders, i.e. direct and unmediated contact between stakeholders and design or development experts. In their ethnographic study focusing on the commonalities of agile methods and user-centered development, Chamberlain *et al.* [35] confirm the importance of continuous stakeholder integration. In contrast, eight (9.6%) of the articles recognize different motivations to involve *stakeholder representatives*, which might be preferable considering resource constraints or difficulties to gain direct access to prospective users. Beyond the specific scope of UCASD, empirical literature has intensively researched the importance of user integration in software development [e.g. 144, 145].

Going beyond the general perspective, two focal areas of stakeholder involvement in the context of UCASD became apparent in the review. The evaluation of the system can be recognized as the most important motivation to establish direct contact with customers, i.e. end-users. In particular, 37 (44.6%) of the reviewed publications discuss *usability testing* with end-users. Nevertheless, practices used in the early phases of software development, in which interactions with stakeholders are necessary to establish system requirements, are equally important. While 12 (14.5%) of the articles only take a generic perspective on *user research* by highlighting the necessity to conduct user research or by discussing the inclusion of user research in an agile development process, 10 (12.0%) publications give explicit attention to *contextual inquiry* in an agile environment. Other practices for stakeholder involvement, including *task analysis* (6 publications, 7.2%), *focus groups* (4 publications, 4.8%), *interviews* (2 publications, 2.4%), and *surveys* (2 publications, 2.4%) are mentioned, but receive less attention.

Suggestion. Stakeholders' continuous involvement in the design and development process is one of the main commonalities of UCD and ASD. Both emphasize the human aspect of software development.

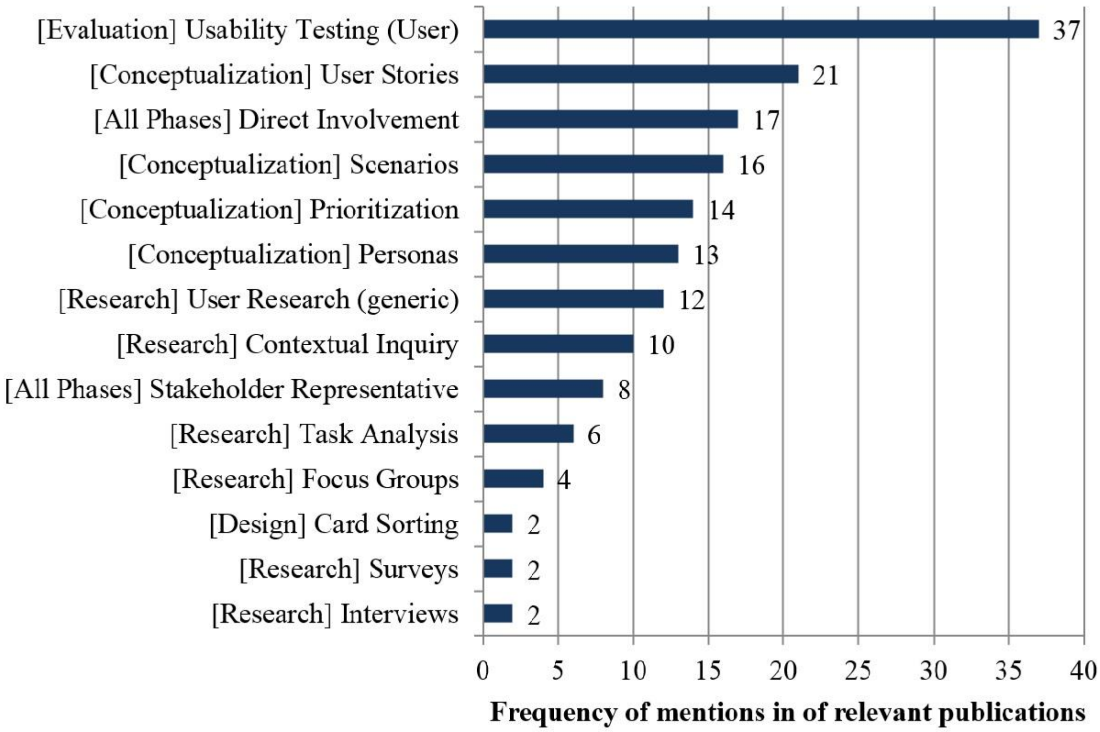

Figure 6: Codes and number of articles related to the continuous stakeholder involvement principle

The value of individuals and interactions is documented in the *agile manifesto* [134]. According to UCD, understanding users and their tasks should be the focus of software development [126]. Moreover, UCD and ASD both encourage customer or user participation in the development of new systems or software. Thus, we suggest the following principle:

> ***Principle 4 (Continuous Stakeholder Involvement):*** *Stakeholders should be actively involved in user-centered agile approaches early on and should remain involved throughout the entire development process to collect input and feedback.*

In the context of UCASD, there is a lack of empirical studies systematically investigating the continuous stakeholder involvement principle. Building on literature on user involvement [144, 145], the extent and outcome of continuous stakeholder integration in UCASD requires further empirical research. From a practical point of view, empirical insights could be leveraged to give context-specific recommendations for applying different stakeholder integration strategies, using associated practices. A deeper understanding of contingency factors and their influence is required to derive corresponding recommendations for successfully applying this principle.

Artifact-Mediated Communication

Focus Area. In contrast to traditional plan-based approaches, ASD discourages the use of formal documentation, as it values "[w]orking software over comprehensive documentation" [134], and knowledge about products and processes is supposed to become tacit and should be exchanged through interactions between team members [137]. In order to reduce "waste," documentation should be reduced to a minimum, with the team only documenting what is absolutely necessary [3,138]. As Kuusinen *et al.* [46] state, there are as yet no commonly agreed upon concepts for communicating design or documenting user requirements in UCASD. Nevertheless, while analyzing the document corpus in the literature review, it became apparent that, besides stakeholder involvement, the use of various *artifacts* related to the design and development process is most prominently discussed in research publications. Artifacts are defined as an "… aspect of the material world that has been modified over the history of its incorporation into goal-directed human action." [139]. This definition incorporates both the tangible and mutable nature of artifacts and their purposeful use. Artifacts, such as those described below, have a long tradition of being used for documentation purposes in design and development activities [99]. Besides supporting creative processes, they are an important means for organizing communication and collaboration among internal and external parties.

Evidence. Discussed in 40 (48.2%) of the publications included in the literature review, prototypes, i.e. visual instantiations of the design concept [140], is the most frequently occurring artifact type. However, only three papers provide empirical evidence supporting this principle [46,48,99]. Remaining very general, Lee *et al.* propose a new approach that combines user-centered and agile software development using central design records (CDRs) as artifacts [48]. Kuusinnen *et al.* focus on the interaction of the UX team and the development team in a large software development organization, identifying the creation and communication of designs as an important task of the UX team [46]. Going into more detail, an ethnographic study by Brown *et al.* [99] analyzes the role of artifacts during the interaction between design and development.

While Brown *et al.* [99] mention prototypes with different levels of fidelity, they usually recommend transitioning from low-fidelity to high fidelity prototypes in design iterations. Additionally, 17 (20.5%) of the included publications consider the use of sketches or mock-ups as artifacts to support the early ideation stage of the design process [140], and nine (10.8%) articles mention the use of *wireframes* to depict the envisioned layout of a user interface. Kuusinen *et al.* [46] describe challenges during the interaction of design and development efforts based on the example of a multi-continental development team. Two studies (2.4%) further discuss the use of multiple wireframes in a *user interface flow*, which is a set of wireframes that visualize which elements of a user interface are used in an interaction path through the system.

A second group of artifacts is inherently linked to stakeholder involvement, as the documentation of stakeholder properties and needs are also of interest to the user-centered agile community. The employment of *user stories* to describe features providing business value to the customer is popular, and mentioned in 21 (25.3%) of the publications in the review. While user stories stem from the agile world, the idea of modeling the system's value proposition in *scenarios* providing a step-by-step narrative on how a prospective user will benefit from the envisioned product is a user-centered development concept, and is discussed in 15 (19.3%) articles. Furthermore, the use of *personas* as concrete representations of user archetypes is highly popular in UCASD, and discussed in 13 (15.7%) publications. Finally, four (4.8%) of the included publications describe the idea of a CDR as a central artifact linking design goals, design decisions, and usability testing results [48]. The use of a CDR supports the goal of reaching a cohesive design.

Suggestion. Artifacts are central means of communication for agile and user-centered software development. Figure 7 depicts the codes and number of articles discussing different types of artifact within the software development process.

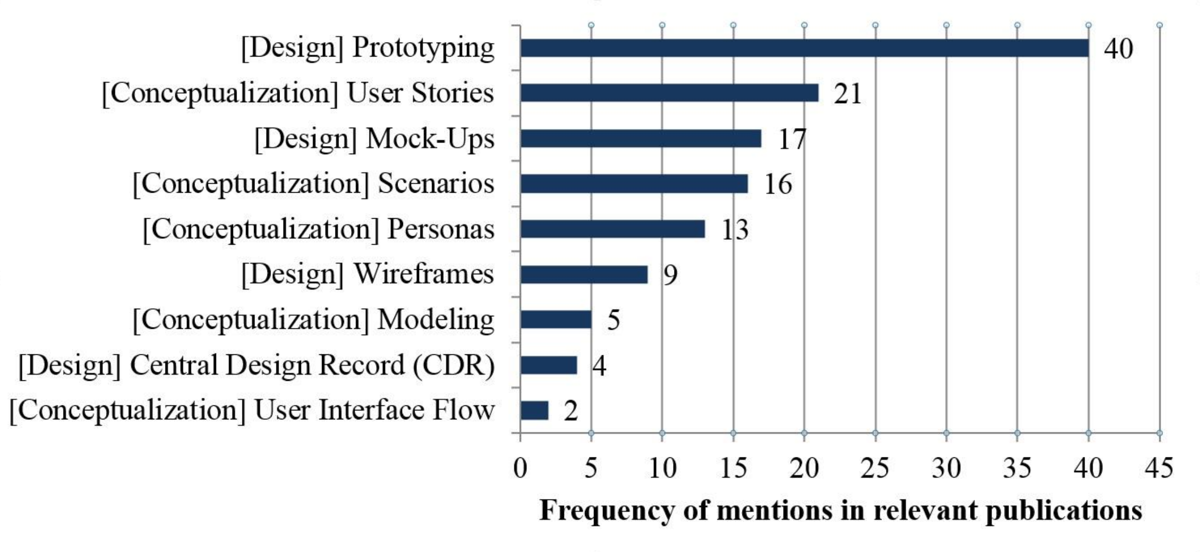

Figure 7: Codes and number of articles related to the artifact-mediated communication principle

While ASD focuses on working prototypes, UCD provides different artifacts to document stakeholder needs and communicate design ideas. On the one hand, user stories, personas, and scenarios are examples of conceptualizations of stakeholder needs. On the other hand, prototypes, wireframe, and mock-ups are established artifacts for communicating designs. To generalize these concepts on a higher abstraction level, we suggest the following principle:

> ***Principle 5 (Artifact-Mediated Communication):*** *In user-centered agile approaches, tangible and up-to-date artifacts should be used to document and communicate product and design concepts, and should be accessible to all involved stakeholders.*

For many years, artifact-mediation communication has been leveraged intensively in other industries, for example, clay modeling in the automotive industry when designing new cars or scale models used by architects to realize a new construction. Surprisingly, empirical evidence for this important principle in the context of UCASD is limited to three publications in our literature review. Following the line of argumentation in principle 4, empirical insights could be leveraged to make context-specific recommendations for applying corresponding artifact-mediated communication strategies and associated practices from a practical point of view. Again, a deeper understanding of contingency factors and their influence on design and development outcomes is needed to derive recommendations for successfully applying this principle.

6. Conclusion

This paper captures the current state of ASD and UCD integration and identifies generic principles that constitute an integrated UCASD approach. To achieve this, we conducted a systematic review of existing literature. In total, 83 publications were identified as relevant and were analyzed along four dimensions. The analysis resulted in a differentiated coding system and created the foundation for suggesting the five principles as summarized in Table 8.

Principle	*Description*
Principle 1	Separate Product Discovery and Product Creation
Principle 2	Iterative and Incremental Design and Development
Principle 3	Parallel Interwoven Creation Tracks
Principle 4	Continuous Stakeholder Involvement
Principle 5	Artifact-Mediated Communication

Table 8: Principles for UCASD.

In order to interpret the implications of our findings adequately, the following limitations of the study need to be considered. Even though a systematic review should ensure a relatively complete census of the relevant literature, completeness can never be guaranteed. Accordingly, we most likely overlooked some relevant articles [36]. While the terminology used in the database query is commonly accepted and used within the scientific community, different terms may have been used to describe relevant methods. Moreover, although clear criteria have been established for assessing the selected articles' relevance (see Section 3), the evaluation was based on the judgment and experience of the authors. Other scholars might have judged these articles differently. The same limitation applies to the coding of each paper using the presented coding system. During the last step of our literature analysis, we derived generic principles based on the identified codes. As discussed earlier, we could only identify principles within the process and practices dimensions. The results for the people/social dimension were contradictory and therefore impeded generalization. The results for the technology dimension were thin and contained hardly any explicit recommendations.

Regardless of the depicted limitations, our literature review has led to the following theoretical contributions. First, we derived a comprehensive coding system, which allows for a broader classification of existing works on UCASD, that scholars who are working in the area of UCASD can use. Besides the application in this paper, the coding system might be used to classify and evaluate future research in this area. Second, the current state of the art in UCASD has been depicted based on the derived coding system. Providing a broad overview of existing work, this compilation can be used as a starting point for scholars who want to research this area. From a practical point of view, the derived principles and their association with specific practices and processes might help scholars apply UCASD in specific contexts. Consequently, our work might contribute to reaching the major goal of UCASD: The delivery of useful *and* usable software.

Our systematic literature review did focus on the 10 years' time frame of 2002 to 2012. As user-centered agile software development is an active research field, we did a crosscheck of our results with most recent publications in the years 2013 and 2014 using the same search strings and the same databases. Consequently, a set of 487 papers has been identified. After a screening of the titles and abstracts, 26 papers actually focusing on the subject of user-centered agile software development have been analyzed further and classified along the identified dimensions as depicted in the table below.

Dimension	**Included Publications**
Process	Salvador et al. [147], Butt et al. [150], da Silvia et al. [152], Plonka et al. [162], Kuusinen [166], Ahmad et al. [167], Maurer & Hellmann [169], Humayoun et al. [171]
Practices	Salvador et al. [147], Arnowitz [148], da Silvia et al. [152], Peres et al. [153], Ardito et al [155], Inayat [156], Salah et al. [157], Cajander et al. [159], Bertholdo et al. [160], Caballero et al. [161], Plonka et al. [162], Lizano et al. [165], Kuusinen [166], Maurer & Hellmann [169], Häger et al. [170]
People & Social	Arnowitz [148], Kropp & Koischwitz [149], Jurca et al. [151], Kuusinen & Mikkonen [154], Raison & Schmidt [158], Cajander et al. [159], Bertholdo et al. [160], Plonka et al. [162], da Silva et al. [163], Heimgärtner & Solanki [164], Lizano et al. [165], Kuusinen [166], Wale-Kolade et al. [168], Häger et al. [170]
Technology	Salvador et al. [147], Humayoun et al. [171]

Table 9 – Overview of recently published papers in 2013/2014 along the four dimensions

In the process dimension the identified papers describe integrated approaches combining the two fields, confirm earlier prescriptions and discuss key challenges with regards to this dimension: E.g., models addressing the trade-offs between usability and agile methods [150] and models supporting the collaboration between software engineering and usability experts are suggested [167,171] Moreover, established concepts, such as sprint 0, iterative evaluation and designing one sprint ahead are further supported [152, 160, 162, 169]. It is recognized that usability methods are often used too late in the development process. Managing the product vision and time boxing the user-centered design work are well-known continuous challenges [147,162,166]. Overall, one can conclude that the three principles identified in our systematic literature review have been confirmed by recent research in the field.

In similar vein, the identified papers related to the practices dimension provide further contextualized and enhanced integration concepts and describe specific challenges. While practices from user-centered design in Scrum improve the understanding of actual user's needs, challenges remain in the actual application of usability practices [165, 166]. Recent work has suggested patterns for the integration of ASD and UCD [161] and a framework for integrating practices [153]. However, the prioritization of non-functional usability requirements in comparison to functional requirements remains challenging [156, 162]. Fast prototyping, individual inquiry, formal tests, and heuristic evaluations are recognized as the most frequent usability practices [147]. Specifically, Low-Fi prototypes and user story maps are suggested as strategic and cost-effective concepts for creating artifacts [169]. Specifically looking at distributed teams, digital low-fidelity prototypes are suggested as a powerful concept to improve the communication of designers and developers [152, 155]. Beyond these practices for stakeholder integration and artifact-mediated communication, the establishing of explicit usability goals and acceptance criteria has been discussed [148, 152]. In sum, recent research was conducted that helps to further refine and contextualize UCASD practices. However, the translation of UCASD practices from academia into practice remains challenging [155], future empirically grounded research is required.

Recently, there has been increasing interest in the people/social dimension of UCASD. A clear definition for the responsibility of the user perspective within a scrum team is needed

[159], as such role influences the product owners' satisfaction [154]. It is recognized that the usability professional needs to be a full member of the agile team [163] with clearly defined responsibilities and authority [159, 166]. The expected contribution may change throughout different development stages [149], being one of the key challenges for its integration [165]. While additional usability training of developers might be a tempting solution [165], usability professionals require encouragement, authority, and enough time to accomplish her tasks [148,151,166]. Moreover, her co-location with the development is a key enabler to improve overall communication. It is important to break old habits in the development team [149].
Overall, contextual factors such as organizational support and cultural change are not to be neglected, as they may lead to power struggles between usability professionals and developers, or a lack of required usability resources for the actual work [151]. Strategic and ongoing support, and a collaborative culture are recognized as important factors within an organization facilitating the integration of ASD and UCD [158,159]. Particularly managers and product owners should be drivers of this cultural change [159, 163]. As a result of an adaptation process in mindset and culture, a team's creativity, pro-activity and speed can be improved [164]. In sum, we see a growing interest in the research community to look at the people/social dimension of UCASD. Still, as discussed earlier in our literature review, it is challenging to organize the manifold contextual factors into stable principles to enable meaningful guidance for successfully applying UCASD in practice.

Finally, only very limited new suggestions have been identified regarding the technology dimension of UCASD. One approach for the automation of usability methods application to reduce cost is described in [147]. Furthermore, a prototypical example for the integration of an integrated UCSAD approach within an IDE to provide guidance to developers [171]. As mentioned earlier, research with regards to the technology dimension of UCASD seems to be currently still in an early stage.

Future research might extend the work presented in this paper. It became obvious during the deeper analysis of the identified UCASD publications in our systematic literature review that only a limited number of papers present rigorous empirical findings. Future research may specifically investigate the identified UCASD principles from an empirical point of view. In this context, we deem it highly relevant to gain a better understanding of the influence of contingency factors, such as software product, team, organization or further characteristics. We also believe that future research should focus on identifying further principles in the people/social as well as the technological dimension of UCASD. Specifically, instead of analyzing existing UCASD literature, subsequent research could explore whether knowledge from other domains (e.g. organizational science or sociology for the people/social dimension and other research streams of computer science for the technology dimension) is applicable to UCASD. This may result in additional, people- and technology-related principles, which could further extend our results. Second, complementing and building on empirical research, we suggest deploying the identified principles in real-world software development projects following an (action) design research approach [146]. Expected outcomes of this type of research could be the creation and evaluation of a UCASD procedure model supporting the execution of software development projects targeting the delivery of useful *and* usable software.

References

[1] F.P. Brooks, The Mythical Man-Month, Addison-Wesley, Reading, MA, USA, 1975.
[2] W.W. Royce, Managing the Development of Large Software Systems, in: Proc. IEEE WESTCON, 1970.
[3] S. Nerur, Theoretical Reflections on Agile Development Methodologies, Communications of the ACM 50 (2007) 79–83.
[4] A. Cockburn, Agile software development: The cooperative game, 2nd ed., Addison-Wesley, Upper Saddle River, NJ, USA, 2006.
[5] K. Beck, C. Andres, Extreme programming eXplained: Embrace change, 2nd ed., Addison-Wesley, Boston, MA, USA, 2004.
[6] K. Conboy, Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development, Information Systems Research 20 (2009) 329–354.
[7] K. Schwaber, M. Beedle, Agile Software Development with Scrum, Prentice Hall, Upper Saddle River, NJ, USA, 2001.
[8] B. Blau, T. Hildenbrand, Product Line Engineering in Large-Scale Lean and Agile Software Product Development Environments - Towards a Hybrid Approach to Decentral Control and Managed Reuse, in: Sixth International Conference on Availability, Reliability and Security (ARES 2011), Aug. 22-26, 2011 Vienna, AT, 2011, pp. 404–408.
[9] B.W. Boehm, Some Future Software Engineering Opportunities and Challenges, in: S. Nanz (Ed.), The Future of Software Engineering, Springer, Berlin / Heidelberg, DE, 2011, pp. 1–32.
[10] D. Batra, W. Xia, D.E. van der Meer, K. Dutta, Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry, Communications of the AIS 27 (2010) 379–394.
[11] B. Fitzgerald, Software Crisis 2.0, IEEE Computer 45 (2012) 89–91.
[12] S. Blomkvist, Towards a Model for Bridging Agile Development and User-Centered Design, in: A. Seffah, J. Gulliksen, M.C. Desmarais (Eds.), Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, Springer Netherlands, Dordrecht, NL, 2005, pp. 219–244.
[13] L.L. Constantine, Process Agility and Software Usability: Toward Lightweight Usage-Centered Design, Information Age 8 (2002) 1–10.
[14] J. Ferreira, J. Noble, R. Biddle, Agile Development Iterations and UI Design, in: J. Eckstein, F. Maurer, R. Davies, G. Melnik, G. Pollice (Eds.), 2007 Agile Conference (AGILE2007), Aug. 13-17, 2007, Washington, DC, USA, 2007, pp. 50–58.
[15] ISO, ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability, 1998.
[16] A. Seffah, M.C. Desmarais, E. Metzker, HCI, usability and software engineering integration: present and future, in: A. Seffah, J. Gulliksen, M.C. Desmarais (Eds.), Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, Springer Netherlands, Dordrecht, NL, 2005, pp. 37–57.
[17] D.J. Mayhew, M.M. Tremaine, A Basic Framework, in: R.G. Bias, D.J. Mayhew (Eds.), Cost-justifying usability: An update for an Internet age, 2nd ed., Morgan Kaufmann, Boston, MA, USA, 2005, pp. 41–101.
[18] F. Scheiber, D. Wruk, A. Oberg, J. Britsch, M. Woywode, A. Maedche, F. Kahrau, H. Meth, D. Wallach, M. Plach, Software Usability in Small and Medium Sized Enterprises in Germany - An Empirical Study, in: A. Mädche, A. Botzenhardt, L. Neer (Eds.), Software for People, Springer, Berlin / Heidelberg, DE, 2012, pp. 39–52.

[19] A. Seffah, E. Metzker, The Obstacles and Myths of Usability and Software Engineering, Communications of the ACM 47 (2004) 71–76.
[20] T.S. da Silva, A. Martin, F. Maurer, M. Silveira, User-Centered Design and Agile Methods: A Systematic Review, in: 2011 Agile Conference (AGILE2011), Aug. 8-12, 2011, Salt Lake City, UT, USA, 2011, pp. 77–86.
[21] D. Fox, J. Sillito, F. Maurer, Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 63–72.
[22] J.T. Barksdale, D.S. McCrickard, Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management, International Journal of Agile and Extreme Software Development 1 (2012) 52–77.
[23] O. Sohaib, K. Khan, Integrating Usability Engineering and Agile Software Development: A Literature Review, in: 2010 International Conference On Computer Design And Applications (ICCDA 2010), June 25-27, 2010, Qinhuangdao, CN, 2010, pp. 32–38.
[24] D. Sy, Adapting Usability Investigations for Agile User-centered Design, Journal of Usability Studies 2 (2007) 112–132.
[25] M.F. Costabile, Usability in the software life cycle, in: S.K. Chang (Ed.), Handbook of Software Engineering & Knowledge Engineering, World Scientific Publishing Co., Singapore, SG, 2001, pp. 179–192.
[26] B. Göransson, M. Lif, J. Gulliksen, Usability Design - Extending Rational Unified Process with a New Discipline, in: Interactive Systems. Design, Specification, and Verification, 10th International Workshop (DSV-IS 200), June 11 - 13, 2003, Funchal, Madeira Island, PT, 2003, pp. 316–330.
[27] C. Clegg, C. Axtell, L. Damodaran, B. Farbey, R. Hull, R. Lloyd-Jones, J. Nicholls, R. Sell, C. Tomlinson, Information Technology: A Study of Performance and the Role of Human and Organizational Factors, Ergonomics 40 (1997) 851–871.
[28] K. Petersen, C. Wohlin, The effect of moving from a plan-driven to an incremental software development approach with agile practices, Empirical Software Engineering 15 (2010) 654–693.
[29] B.A. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering: EBSE Technical Report EBSE-2007-01. Keele University, Keele, UK, 2007.
[30] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (2007) 571–583.
[31] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: A systematic review, Information and Software Technology 50 (2008) 833–859.
[32] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - A systematic literature review, Information and Software Technology 51 (2009) 7–15.
[33] S.A. Carlsson, S. Henningsson, S. Hrastinski, C. Keller, Socio-Technical IS Design Science Research: Developing Design Theory for IS Integration Management, Information Systems and e-Business Management 9 (2011) 109–131.
[34] J. vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, A. Cleven, Reconstructing the giant: On the importance of rigour in documenting the literature search process, in: S. Newell, E.A. Whitley, N. Pouloudi, J. Wareham, L. Mathiassen (Eds.), 17th European Conference on Information Systems (ECIS 2009), June 8-10, 2010, Verona, IT, 2009, pp. 2206–2217.

[35] S. Chamberlain, H. Sharp, N.A.M. Maiden, Towards a Framework for Integrating Agile Development and User-Centred Design, in: 7th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2006), June 17-22, 2006, Oulu, FI, 2006, pp. 143–153.

[36] J. Webster, R.T. Watson, Analyzing the Past to Prepare for the Future: Writing a Literature Review, MIS Quarterly 26 (2002) xiii–xxiii.

[37] E. Hossain, M.A. Babar, H.-y. Paik, Using Scrum in Global Software Development: A Systematic Literature Review, in: Fourth IEEE International Conference on Global Software Engineering (ICGSE 2009), July 13-16, 2009, Limerick, IE, 2009, pp. 175–184.

[38] S. Wang, S. Zheng, L. Xu, D. Li, H. Meng, A literature review of electronic marketplace research: Themes, theories and an integrative framework, Information Systems Frontiers 10 (2008) 555–571.

[39] G. Benigni, O. Gervasi, F.L. Passeri, T.-H. Kim, USABAGILE_Web: A Web Agile Usability Approach for Web Site Design, in: D. Taniar, O. Gervasi, B. Murgante, E. Pardede, B.O. Apduhan (Eds.), Proceedings of the International Conference on Computational Science and Its Applications (ICCSA 2010), March 23-26, 2010, Fukuoka, JP, 2010, pp. 422–431.

[40] M. Budwig, S. Jeong, K. Kelkar, When user experience met agile: a case study, in: Dan R. Olsen Jr., R.B. Arthur, K. Hinckley, M.R. Morris, S.E. Hudson, S. Greenberg (Eds.), Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI 2009), April 4-9, 2009, Boston, MA, USA, 2009, pp. 3075–3084.

[41] C. Felker, R. Slamova, J. Davis, Integrating UX with scrum in an undergraduate software development project, in: L.A. Smith King, D.R. Musicant, T. Camp, P.T. Tymann (Eds.), Proceedings of the 43rd ACM technical symposium on Computer science education (SIGCSE 2012), Feb. 29 - Mar. 3, Raleigh, NC, USA, 2012, pp. 301–306.

[42] J. Ferreira, J. Noble, R. Biddle, Up-Front Interaction Design in Agile Development, in: 8th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2007), June 18-22, 2007, Como, IT, 2007, pp. 9–16.

[43] A. Holzinger, M. Errath, G. Searle, B. Thurnher, W. Slany, From Extreme Programming and Usability Engineering to Extreme Usability in Software Engineering Education (XP+UE->XU), in: 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), July 25-28, 2005, Edinburgh, UK, 2005, pp. 169–172.

[44] Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, T. Vlk, P. Wolkerstorfer, User Interface Design for a Mobile Multimedia Application: An Iterative Approach, in: First International Conference on Advances in Computer-Human Interaction (ACHI 2008), Feb. 10-15, 2008, Sainte Luce, FR, 2008, pp. 189–194.

[45] Z. Hussain, M.M.H. Lechner, W. Slany, M. Umgeher, P. Wolkerstorfer, Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application, in: A. Holzinger (Ed.), Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB 2008), Nov. 20-21, Graz, AT, 2008, pp. 313–330.

[46] K. Kuusinen, T. Mikkonen, S. Pakarinen, Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact, in: M. Winckler, P. Forbrig, R. Bernhaupt (Eds.), Proceedings of the 4th International Conference on Human-Centered Software Engineering (HCSE 2012), Oct. 29-31, 2012, Toulouse, FR, 2012, pp. 94–111.

[47] J.C. Lee, D.S. McCrickard, Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development, in: J. Eckstein, F. Maurer, R.

Davies, G. Melnik, G. Pollice (Eds.), 2007 Agile Conference (AGILE2007), Aug. 13-17, 2007, Washington, DC, USA, 2007.

[48] J.C. Lee, D.S. McCrickard, K.T. Stevens, Examining the Foundations of Agile Usability with eXtreme Scenario-Based Design, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 3–10.

[49] J.C. Lee, T.K. Judge, D.S. McCrickard, Evaluating eXtreme scenario-based design in a distributed agile team, in: D.S. Tan, S. Amershi, B. Begole, W.A. Kellogg, M. Tungare (Eds.), Proceedings of the International Conference on Human Factors in Computing Systems (CHI 2011), May 7-12, 2011, Vancouver, BC, CA, 2011, pp. 863–877.

[50] B. Losada, M. Urretavizcaya, Isabel Fernández de Castro, An Integrated Approach to Develop Interactive Software, in: J.A. Jacko (Ed.), Proceedings of the 14th International Conference on Human-Computer Interaction. Design and Development Approaches. (HCI International 2011), July 9-14, Orlando, FL, USA, 2011, pp. 470–474.

[51] B. Losada, M. Urretavizcaya, J.-M. López-Gil, I. Fernández-Castro, Combining InterMod agile methodology with usability engineering in a mobile application development, in: Proceedings of the 13th International Conference on Interacción Persona-Ordenador (INTERACCION ’12), Oct. 3-5, 2012, Elche, ES, 2012, pp. 39:1-39:8.

[52] B. Losada, M. Urretavizcaya, I. Fernández-Castro, A guide to agile development of interactive software with a “User Objectives”-driven methodology, Science of Computer Programming (in press).

[53] T. Memmel, F. Gundelsweiler, H. Reiterer, CRUISER: A Cross-Discipline User Interface and Software Engineering Lifecycle, in: J.A. Jacko (Ed.), Proceedings of the 12th International Conference on Human-Computer Interaction. Interaction Design and Usability (HCI International 2007), July 22-27, Beijing, CN, 2007, pp. 174–183.

[54] T. Memmel, F. Gundelsweiler, H. Reiterer, Agile human-centered software engineering, in: T.C. Ormerod, C. Sas (Eds.), Proceedings of the 21st British HCI Group Annual Conference on HCI 2007: HCI…but not as we know it (BCS HCI 2007), Sept. 3-7, 2007, Lancaster, UK, 2007, pp. 167–175.

[55] L. Miller, Case Study of Customer Input For a Successful Product, in: 2005 Agile Conference (AGILE 2005), July 24-29, Denver, CO, USA, IEEE Computer Society, 2005, pp. 225–234.

[56] M. Najafi, L. Toyoshiba, Two Case Studies of User Experience Design and Agile Development, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 531–536.

[57] V. Paelke, K. Nebe, Integrating Agile Methods for Mixed Reality Design Space Exploration, in: Proceedings of the 7th ACM conference on Designing interactive systems (DIS 2008), Feb. 25-27, 2008, Cape Town, ZA, 2008, pp. 240–249.

[58] V. Paelke, M. Sester, Augmented paper maps: Exploring the design space of a mixed reality system: Theme issue “Visualization and exploration of geospatial data”, ISPRS Journal of Photogrammetry and Remote Sensing 65 (2010) 256–265.

[59] P. Wolkerstorfer, M. Tscheligi, R. Sefelin, H. Milchrahm, Z. Hussain, M. Lechner, S. Shahzad, Probing an agile usability process, in: Extended Abstracts Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI 2008), April 5-10, 2008, Florence, Italy, 2008, pp. 2151–2158.

[60] P. Zhang, J. Carey, D. Te'eni, M. Tremaine, Integrating Human-Computer Interaction Development into the Systems Development Life Cycle: A Methodology, Communications of the Association for Information Systems 15 (2005) Article 29.

[61] S. Adikari, C. Mcdonald, J. Campbell, Little Design Up-Front: A Design Science Approach to Integrating Usability into Agile Requirements Engineering, in: J.A. Jacko (Ed.), Proceedings of the 13th International Conference on Human-Computer

Interaction. New Trends. (HCI International 2009), July 19-24, San Diego, CA, USA, 2009, pp. 549–558.
[62] H.R. Beyer, K. Holtzblatt, L. Baker, An Agile Customer-Centered Method: Rapid Contextual Design, in: C. Zannier, H. Erdogmus, L. Lindstrom (Eds.), Proceedings of the 4th Conference on Extreme Programming and Agile Methods (XP/Agile Universe 2004), Aug. 15-18, 2004, Calgary, CA, 2004, pp. 50–59.
[63] D. Broschinsky, L. Baker, Using Persona with XP at LANDesk Software, an Avocent Company, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 543–548.
[64] C.R.M. de Carvalho, MEX experience boards: a set of agile tools for user experience design, in: Proceedings of the IX Brazilian symposium on Human Factors in Computing Systems (IHC ’10), Oct. 5-8, 2010, Belo Horizonte, BR, 2010, pp. 213–216.
[65] L. Cho, Adopting an Agile Culture, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 400–403.
[66] L.L. Constantine, L.A.D. Lockwood, Usage-Centered Engineering for Web Applications, IEEE Software 19 (2002) 42–50.
[67] B.S. da Silva, V.C.O. Aureliano, S.D.J. Barbosa, Extreme designing: binding sketching to an interaction model in a streamlined HCI design approach, in: Proceedings of the VII Brazilian symposium on Human factors in computing systems (IHC '06), Nov. 19 - 22, 2006, Natal, BR, 2006, pp. 101–109.
[68] M. Detweiler, Managing UCD within agile projects, interactions 14 (2007) 40–42.
[69] M. Düchting, D. Zimmermann, K. Nebe, Incorporating User Centered Requirement Engineering into Agile Software Development, in: J.A. Jacko (Ed.), Proceedings of the 12th International Conference on Human-Computer Interaction. Interaction Design and Usability (HCI International 2007), July 22-27, Beijing, CN, 2007, pp. 58–67.
[70] J. Evnin, M. Pries, Are You Sure? Really? A Contextual Approach to Agile User Research, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 537–542.
[71] X. Ferré, N.J. Juzgado, A.M. Moreno, Framework for Integrating Usability Practices into the Software Process, in: F. Bomarius, S. Komi-Sirviö (Eds.), Proceedings of the 6th International Conference on Product Focused Software Process Improvement (PROFES 2005), June 13-15, 2005, Oulu, FI, 2005, pp. 202–215.
[72] K.G. Fisher, A. Bankston, From Cradle to Sprint: Creating a Full-Lifecycle Request Pipeline at Nationwide Insurance, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 223–228.
[73] J. Haikara, Usability in Agile Software Development: Extending the Interaction Design Process with Personas Approach, in: 8th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2007), June 18-22, 2007, Como, IT, 2007, pp. 153–156.
[74] C. Hansson, Y. Dittrich, D.W. Randall, How to Include Users in the Development of Off-the-Shelf Software: A Case for Complementing Participatory Design with Agile Development, in: 39th Hawaii International Conference on Systems Science (HICSS 2006), Jan. 4-7, 2006, Kauai, HI, USA, 2006.
[75] T.D. Hellmann, A. Hosseini-Khayat, F. Maurer, Agile Interaction Design and Test-Driven Development of User Interfaces - A Literature Review, in: T. Dingsøyr, T. Dybå, N.B. Moe (Eds.), Agile Software Development, Springer, Berlin / Heidelberg, DE, 2010, pp. 185–201.
[76] T.D. Hellmann, A. Hosseini-Khayat, F. Maurer, Supporting Test-Driven Development of Graphical User Interfaces Using Agile Interaction Design, in: Third International Conference on Software Testing, Verification and Validation (ICST 2010), April 7-9, 2010, Paris, FR, 2010, pp. 444–447.

[77] L. Hennings, Making Design Tangible in Software Development Projects, in: A. Maedche, A. Botzenhardt, L. Neer (Eds.), Software for People, Springer, Berlin / Heidelberg, DE, 2012, pp. 151–167.

[78] P. Hodgetts, Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes, in: 2005 Agile Conference (AGILE 2005), July 24-29, Denver, CO, USA, IEEE Computer Society, 2005, pp. 235–242.

[79] S.R. Humayoun, Y. Dubinsky, T. Catarci, A Three-Fold Integration Framework to Incorporate User - Centered Design into Agile Software Development, in: Human Centered Design - Second International Conference (HCD 2011), Held as Part of HCI International 2011, July 9-14, 2011, Orlando, FL, USA, 2011, pp. 55–64.

[80] Z. Hussain, H. Milchrahm, S. Shahzad, W. Slany, M. Tscheligi, P. Wolkerstorfer, Integration of Extreme Programming and User-Centered Design: Lessons Learned, in: 10th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2009), May 25-29, 2009, Pula, Sardinia, IT, 2009, pp. 174–179.

[81] T. Illmensee, A. Muff, 5 Users Every Friday: A Case Study in Applied Research, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 404–409.

[82] M. Isomursu, A. Sirotkin, P. Voltti, M. Halonen, User Experience Design Goes Agile in Lean Transformation - A Case Study, in: 2012 Agile Conference (AGILE 2012), Dallas, TX, USA, August 13-17, 2012, 2012, pp. 1–10.

[83] D. Kane, Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet, in: 2003 Agile Development Conference (ADC 2003), June 25-28, 2003, Salt Lake City, UT, USA, 2003, pp. 40–46.

[84] M.K. Lárusdóttir, Usability Evaluation in Software Development Practice, in: P. Campos, T.C.N. Graham, J.A. Jorge, N.J. Nunes, P.A. Palanque, M. Winckler (Eds.), 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2011), Sep. 5-9, 2011, Lisbon, PT, 2011, pp. 430–433.

[85] M.K. Lárusdóttir, E.R. Bjarnadottir, J. Gulliksen, The Focus on Usability in Testing Practices in Industry, in: P. Forbrig, F. Paternó, A. Mark Pejtersen (Eds.), Human-Computer Interaction, Springer, Berlin / Heidelberg, DE, 2010, pp. 98–109.

[86] S.-H. Lee, I.-Y. Ko, S. Kang, D. Lee, A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features, in: S.I. Ahamed, D.-H. Bae, S.D. Cha, C.K. Chang, R. Subramanyan, E. Wong, H.-I. Yang (Eds.), Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference (COMPSAC 2010), July 19-23, 2010, Seoul, KR, 2010, pp. 317–326.

[87] N. Medina-Medina, J. Burella, G. Rossi, J. Grigera, E.R. Luna, An Incremental Approach for Building Accessible and Usable Web Applications, in: L. Chen, P. Triantafillou, T. Suel (Eds.), Proceedings of the 11th International Conference on Web Information Systems Engineering (WISE 2010), Dec. 12-14, 2010, Hong Kong, CN, 2010, pp. 564–577.

[88] T. Memmel, H. Reiterer, A. Holzinger, Agile Methods and Visual Specification in Software Development: A Chance to Ensure Universal Access, in: J.A. Jacko (Ed.), Proceedings of the 12th International Conference on Human-Computer Interaction. Interaction Design and Usability (HCI International 2007), July 22-27, Beijing, CN, 2007, pp. 453–462.

[89] G. Meszaros, J. Aston, Adding usability testing to an agile project, in: 2006 Agile Conference (AGILE 2006), July 23-28, 2006, Minneapolis, MN, USA, 2006, pp. 6 pp.

[90] H. Obendorf, M. Finck, Scenario-based usability engineering techniques in agile development processes, in: Extended Abstracts Proceedings of the 2008 Conference on

Human Factors in Computing Systems (CHI 2008), April 5-10, 2008, Florence, Italy, 2008, pp. 2159–2166.
[91] H. Obendorf, A. Schmolitzky, M. Finck, XPnUE - defining and teaching a fusion of eXtreme programming and usability engineering, in: HCI Educators Workshop 2006 (HCIEd 2006), March 23-24, 2006, Limerick, IE, 2006.
[92] J. Patton, Designing Requirements: Incorporating Usage-Centered Design into an Agile SW Development Process, in: D. Wells, L.A. Williams (Eds.), Extreme Programming and Agile Methods, Second XP Universe and First Agile Universe Conference (XP/Agile Universe 2002), Aug. 4-7, 2002, Chicago, IL, USA, 2002, pp. 1–12.
[93] J. Patton, Hitting the Target: Adding Interaction Design to Agile Software Development, in: ACM SIGPLAN Object Oriented Programming Systems Languages and Applications Conference (OOPSLA 2002) - Practitioners Reports, Nov. 4 - 8, 2002, Seattle, WA, USA, 2002, pp. 1–7.
[94] K. Petrovic, M. Siegmann, Make Space for the Customer: The Shift towards Customer Centricity, in: A. Marcus (Ed.), Proceedings of the First International Conference on Design, User Experience, and Usability. Theory, Methods, Tools and Practice (DUXU 2011), Held as Part of HCI International 2011, July 9-14, 2011, Orlando, FL, USA, 2011, pp. 485–490.
[95] T. Rafla, P.N. Robillard, M. Desmarais, A method to elicit architecturally sensitive usability requirements: its integration into a software development process, Software Quality Journal 15 (2007) 117–133.
[96] M. Rittenbruch, G. McEwan, N. Ward, T. Mansfield, D. Bartenstein, Extreme participation - moving extreme programming towards participatory design, in: Participation and Design: Inquiring Into the Politics, Contexts and Practices of Collaborative Design Work (PDC 2002), June 23 - 25, 2002, Malmø, SE, 2002, pp. 23–25.
[97] J.T. Barksdale, D.S. McCrickard, Concept mapping in agile usability: a case study, in: E.D. Mynatt, D. Schoner, G. Fitzpatrick, S.E. Hudson, W.K. Edwards, T. Rodden (Eds.), Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010), April 10-15, 2010, Atlanta, GA, USA, 2010, pp. 4691–4694.
[98] J.T. Barksdale, E.D. Ragan, D.S. McCrickard, Easing Team Politics in Agile Usability: A Concept Mapping Approach, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 19–25.
[99] J. Brown, G. Lindgaard, R. Biddle, Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 39–50.
[100] J.M. Brown, G. Lindgaard, R. Biddle, Collaborative Events and Shared Artefacts: Agile Interaction Designers and Developers Working Toward Common Aims, in: 2011 Agile Conference (AGILE2011), Aug. 8-12, 2011, Salt Lake City, UT, USA, 2011, pp. 87–96.
[101] J. Ferreira, H. Sharp, H. Robinson, Values and Assumptions Shaping Agile Development and User Experience Design in Practice, in: A. Sillitti, A. Martin, X. Wang, E. Whitworth (Eds.), Proceedings of the 11th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2010), June 1-4, 2010, Trondheim, NO, 2010, pp. 178–183.
[102] J. Ferreira, H. Sharp, H. Robinson, Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice, in: 2012 Agile Conference (AGILE 2012), Dallas, TX, USA, August 13-17, 2012, 2012, pp. 11–20.

[103] J. Kollmann, H. Sharp, A. Blandford, The Importance of Identity and Vision to User Experience Designers on Agile Projects, in: 2009 Agile Conference (AGILE2009), Aug. 24-28, 2009, Chicago, IL, USA, 2009, pp. 11–18.
[104] A. Leszek, C. Courage, The Doctor is "In" - Using the Office Hours Concept to Make Limited Resources Most Effective, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 196–201.
[105] M.A. Lievesley, J.S.R. Yee, The role of the interaction designer in an agile software development process, in: G.M. Olson, R. Jeffries (Eds.), Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006), April 22-27, 2006, Montréal, CA, 2006, pp. 1025–1030.
[106] M. Singh, U-SCRUM: An Agile Methodology for Promoting Usability, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 555–560.
[107] J. Ungar, The Design Studio: Interface Design for Agile Teams, in: 2008 Agile Conference (AGILE2008), August 4-8, 2008, Toronto, ON, CA, 2008, pp. 2167–2178.
[108] J. Ungar, J. White, Agile user centered design: enter the design studio - a case study, in: Extended Abstracts Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI 2008), April 5-10, 2008, Florence, Italy, 2008, pp. 2167–2178.
[109] H. Williams, A. Ferguson, The UCD Perspective: Before and After Agile, in: J. Eckstein, F. Maurer, R. Davies, G. Melnik, G. Pollice (Eds.), 2007 Agile Conference (AGILE2007), Aug. 13-17, 2007, Washington, DC, USA, 2007, pp. 285–290.
[110] J. Feiner, K. Andrews, Usability Reporting with UsabML, in: M. Winckler, P. Forbrig, R. Bernhaupt (Eds.), Proceedings of the 4th International Conference on Human-Centered Software Engineering (HCSE 2012), Oct. 29-31, 2012, Toulouse, FR, 2012, pp. 342–351.
[111] J. Gonçalves, C. Santos, POLVO - Software for Prototyping of Low-Fidelity Interfaces in Agile Development, in: P. Campos, T.C.N. Graham, J.A. Jorge, N.J. Nunes, P.A. Palanque, M. Winckler (Eds.), 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2011), Sep. 5-9, 2011, Lisbon, PT, 2011, pp. 63–71.
[112] A. Hosseini-Khayat, T.D. Hellmann, F. Maurer, Distributed and Automated Usability Testing of Low-Fidelity Prototypes, in: S. Freudenberg, J. Chao (Eds.), 2010 Agile Conference (AGILE2010), Aug. 9-13, 2010, Orlando, FL, USA, 2010, pp. 59–66.
[113] S.R. Humayoun, Y. Dubinsky, T. Catarci, E. Nazarov, A. Israel, A model-based approach to ongoing product evaluation, in: G. Tortora, S. Levialdi, M. Tucci (Eds.), Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI ’12), May 22-25, 2012, Capri Island, Naples, IT, 2012, pp. 596–603.
[114] J.C. Lee, Embracing agile development of usable software systems, in: G.M. Olson, R. Jeffries (Eds.), Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006), April 22-27, 2006, Montréal, CA, 2006, pp. 1767–1770.
[115] C.S.A. Peixoto, Human-computer interface expert system for agile methods, in: V. Luzar-Stiffler, I. Jarec, Z. Bekic (Eds.), Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces, June 22-25, 2009, Cavtat/Dubrovnik, HR, 2009, pp. 311–316.
[116] C.S.A. Peixoto, A.E.A. da Silva, A Conceptual Knowledge Base Representation for Agile Design of Human-Computer Interface, in: Third International Symposium on Intelligent Information Technology Application (IITA 2009), Nov. 21, 2009, Nanchang, CN, 2009, pp. 156–160.
[117] ISO, ISO 9241-210:2010 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, 2010.

[118] J.-Y. Mao, K. Vredenburg, P.W. Smith, T. Carey, The State of User-Centered Design Practice, Communications of the ACM 48 (2005), pp. 105–109.

[119] P. Kettunen, Adopting key lessons from agile manufacturing to agile software product development - A comparative study, Technovation 29 (2009), pp. 408–422.

[120] J.A. Highsmith, Agile project management: Creating innovative products, 2nd ed., Addison-Wesley, Upper Saddle River, NJ, USA, 2010.

[121] K. Schwaber, J. Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, 2011, http://www.scrum.org/scrum-guides, accessed 1 April 2013.

[122] A. Cooper, The inmates are running the asylum, Sams, Indianapolis, IN, USA, 2004.

[123] J.D. Gould, C. Lewis, Designing for Usability: Key Principles and What Designers Think, Communications of the ACM 28 (1985), pp. 300–311.

[124] H. Sharp, Y. Rogers, J. Preece, Interaction Design: Beyond Human-Computer Interaction, 2nd ed., John Wiley & Sons Inc., West Sussex, UK, 2007.

[125] X. Ferré, N. Medinilla, How a Human-Centered Approach Impacts Software Development, in: J.A. Jacko (Ed.), Proceedings of the 12th International Conference on Human-Computer Interaction. Interaction Design and Usability (HCI International 2007), July 22-27, Beijing, CN, 2007, pp. 68–77.

[126] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, Å. Cajander, Key principles for user-centred systems design, Behaviour & Information Technology 22 (2003), pp. 397–409.

[127] C. Larman, Agile and iterative development: A manager's guide, Addison-Wesley, Boston, MA, USA, 2003.

[128] A. Cooper, R. Reimann, D. Cronin, About Face 3: The Essentials of Interaction Design, 3rd ed., John Wiley & Sons Inc., Indianapolis, IN, USA, 2007.

[129] M. Cohn, User stories applied: For agile software development, Addison-Wesley, Boston, MA, USA, 2004.

[130] R.C. Martin, Iterative and Incremental Development (IID). Engineering Notebook Column, 1999.

[131] B.W. Boehm, Requirements that Handle IKIWISI, COTS, and Rapid Change, IEEE Computer 33 (2000) 99–102.

[132] D. Gebert, S. Boerner, E. Kearney, Cross-functionality and innovation in new product development teams: A dilemmatic structure and its consequences for the management of diversity, European Journal of Work and Organizational Psychology 15 (2006), pp. 431–458.

[133] M.H.-J. Park, J.W. Lim, P.H. Birnbaum-More, The Effect of Multiknowledge Individuals on Performance in Cross-Functional New Product Development Teams, Journal of Product Innovation Management 26 (2009), pp. 86–96.

[134] M. Fowler, J. Highsmith, The Agile Manifesto, 2001, http://agilemanifesto.org/, accessed 1 April 2013.

[135] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, J. Still, The impact of agile practices on communication in software development, Empirical Software Engineering 13 (2008), pp. 303–337.

[136] A. Martin, R. Biddle, J. Noble, The XP Customer Role in Practice: Three Studies, in: 2004 Agile Development Conference (ADC 2004), June 22 - 26, 2004, Salt Lake City, UT, USA, 2004, pp. 42–54.

[137] S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, Communications of the ACM 48 (2005), pp. 72–78.

[138] T. Dingsøyr, S. Nerur, V. Balijepally, N.B. Moe, A decade of agile methodologies: Towards explaining agile software development, Journal of Systems and Software 85 (2012), pp.1213–1221.

[139] M. Cole, Cultural psychology: A once and future discipline, Harvard University Press, Cambridge, MA, USA, 1996.
[140] W. Buxton, Sketching user experiences: Getting the design right and the right design, Morgan Kaufmann, Boston, MA, USA, 2007.
[141] P. Bourdieu, Outline of a Theory of Practice (Vol. 16). Cambridge University Press, 1977.
[142] A. Botzenhardt, H. Meth, A. Maedche, Cross-Functional Integration of Product Management and Product Design in Application Software Development: Exploration of Success Factors, ICIS 2011 Proceedings, 2011.
[143] L. C. Troy, T. Hirunyawipada, A. K. Paswan, Cross-functional integration and new product success: an empirical investigation of the findings, Journal of Marketing 72:6 (2008), pp. 132–146.
[144] R. Rasmussen, A. S. Christensen, T. Fjeldsted, M. Hertzum, Selecting users for participation in IT projects: Trading a representative sample for advocates and champions?, Interacting with Computers, 23:2 (2011), pp. 176-187.
[145] M. A. Harris, H. R. Weistroffer, A New Look at the Relationship between User Involvement in Systems Development and System Success. Communications of the Association for Information Systems, 24:42 (2009), pp. 739-756.
[146] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, R. Lindgren, Action Design Research, MIS Quarterly, MIS Quarterly, 35:2 (2011), pp. 1-20.
[147] C. Salvador, A. Nakasone, J. A. Pow-Sang. A systematic review of usability techniques in agile methodologies. In Proceedings of the 7th Euro American Conference on Telematics and Information Systems (2014), p. 17.
[148] J. Arnowitz. Taking the fast RIDE: designing while being agile. Interactions 20:4 (2013): 76-79.
[149] E. Kropp, K. Koischwitz. User-centered-design in agile RE through an On-site User Experience Consultant. In 2nd International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE 2014), , pp. 9-12.
[150] S. M. Butt, W. F. W. Ahmad, L. Rahim (2014, June). Handling tradeoffs between agile and usability methods. In International Conference on Computer and Information Sciences (ICCOINS 2014), pp. 1-6.
[151] G. Jurca, T. D. Hellmann, F. Maurer. Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX. In Agile Conference (AGILE 2014), pp. 24-32.
[152] T. S. da Silva, M. S. Silveira, F. Maurer. Ten Lessons Learned from Integrating Interaction Design and Agile Development. In Agile Conference (AGILE 2013), pp. 42-49..
[153] A. L. Peres, T. S. da Silva, F. Selleri Silva, F. Furtado Soares, C. Rosemberg Maia De Carvalho, S. Romero De Lemos Meira. AGILEUX Model: Towards a Reference Model on Integrating UX in Developing Software Using Agile Methodologies. In Agile Conference (AGILE 2014), pp. 61-63. [154] Kuusinen, Kati, and Tommi Mikkonen. Designing User Experience for Mobile Apps: Long-Term Product Owner Perspective. In 20th Asia-Pacific Software Engineering Conference (APSEC, 2013) pp. 535-540..
[155] C. Ardito, P. Buono, D. Caivano, M. F. Costabile, R. Lanzilotti. Investigating and promoting UX practice in industry: An experimental study. International Journal of Human-Computer Studies 72:6 (2014): 542-551.
[156] I. Inayat, S. Salwah Salim, S. Marczak, M. Daneva, S. Shamshirband. A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior (2014).

[157] D. Salah, R. Paige, P. Cairns. Integrating Agile Development Processes and User Centred Design-A Place for Usability Maturity Models?. In Human-Centered Software Engineering, Springer, Berlin / Heidelberg, DE, 2014, pp. 108-125

[158] C. Raison, S. Schmidt. Keeping user centred design (UCD) alive and well in your organisation: taking an agile approach. In Design, User Experience, and Usability. Design Philosophy, Methods, and Tools, Springer, Berlin / Heidelberg, DE 2013, pp. 573-582.

[159] Å Cajander, M. Larusdottir, J. Gulliksen. Existing but not explicit-The user perspective in scrum projects in practice. In Human-Computer Interaction–INTERACT 2013, Springer, Berlin / Heidelberg, 2013, pp. 762-779.

[160] A. P. O. Bertholdo, T. S. da Silva, C. de O. Melo, F. Kon, M. Selbach Silveira. Agile Usability Patterns for UCD Early Stages. In Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience, Springer International Publishing,Cham, CH 2014, pp. 33-44.

[161] L. Caballero, A. M. Moreno, A. Seffah. Persona as a Tool to Involving Human in Agile Methods: Contributions from HCI and Marketing. In Human-Centered Software Engineering, Springer, Berlin / Heidelberg, 2014, pp. 283-290.

[162] L. Plonka, H. Sharp, P. Gregory, K. Taylor. UX Design in Agile: A DSDM Case Study. In Agile Processes in Software Engineering and Extreme Programming, Springer International Publishing, Cham, CH, 2014, pp. 1-15.

[163] T. S. da Silva, M. Selbach Silveira, C. de O. Melo, L. C. Parzianello. Understanding the UX Designer's Role within Agile Teams. In Design, User Experience, and Usability. Design Philosophy, Methods, and Tools, Springer, Berlin / Heidelberg, 2013, pp. 599-609.

[164] R. Heimgärtner, A. Solanki. Using Agile Methods in Intercultural HCI Design Projects. In Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience, Springer International Publishing, Cham, CH, 2014, pp. 123-129.

[165] F. Lizano, M. M. Sandoval, J. Stage. Integrating Usability Evaluations into Scrum: A Case Study Based on Remote Synchronous User Testing. In Human-Computer Interaction. Theories, Methods, and Tools, Springer International Publishing, Cham, CH, 2014, pp. 500-509.

[166] K. Kuusinen. Improving UX Work in Scrum Development: A Three-Year Follow-Up Study in a Company. In Human-Centered Software Engineering, Springer, Berlin / Heidelberg, 2014, pp. 259-266.

[167] W. F. W. Ahmad, S. M. Butt, L. Rahim. Usability Evaluation of the Agile Software Process. In Advances in Visual Informatics, Springer International Publishing, Cham, CH, 2013, pp. 640-651.

[168] A. Wale-Kolade, P. A. Nielsen, T. Päivärinta. Usability Work in Agile Systems Development Practice: A Systematic Review. In Building Sustainable Information Systems, Springer, NY, US, 2013, pp. 569-582.

[169] F. Maurer, T. D. Hellmann. People-Centered Software Development: An Overview of Agile Methodologies. In Software Engineering, Springer, Berlin / Heidelberg, 2013, pp. 185-215.

[170] F. Häger, T. Kowark, J. Krüger, C. Vetterli, F. Übernickel, M. Uflacker. DT@ Scrum: Integrating Design Thinking with Software Development Processes. In Design Thinking Research, Springer International Publishing, Cham, CH, 2015, pp. 263-289.

[171] S. R. Humayoun, Y. Dubinsky, T. Catarci. User Evaluation Support Through Development Environment for Agile Software Teams. In Smart Organizations and Smart Artifacts, Springer International Publishing, Cham, CH, 2014, pp. 183-191.

Appendix A: Practices for UCASD

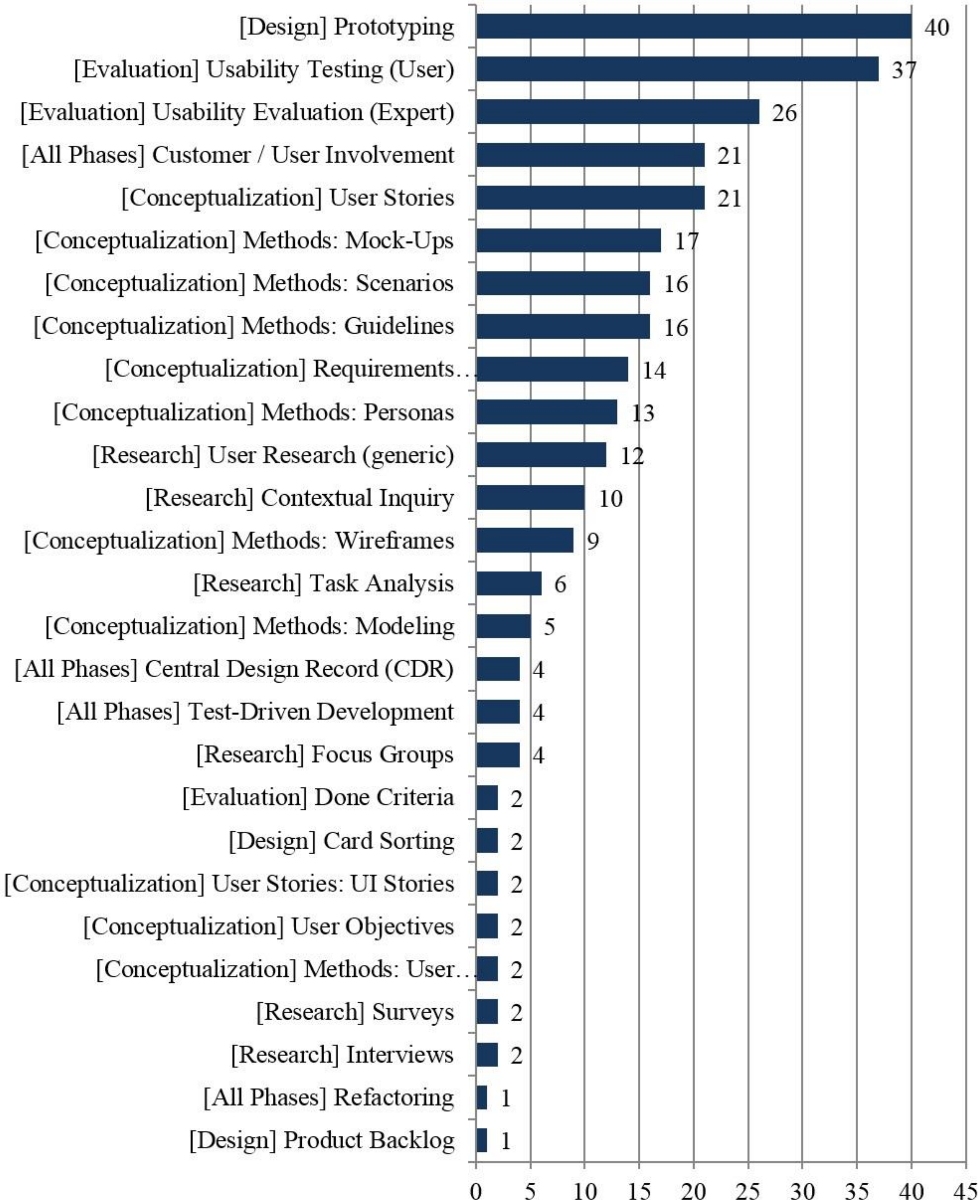

Figure 8: Codes and number of articles related to dimension "Practices"

Appendix B: Additional Codes

Level 1	Level 2	Level 3	Level 4
Process	Design Up Front	Big Design Up Front	-
Process	Design Up Front	Little Design Up Front	-
Practices	Evaluation	Customer / User Involvement	Participatory Design
Practices	Evaluation	Customer / User Involvement	Direct Involvement
Practices	Evaluation	Customer / User Involvement	User Representative
Practices	Evaluation	Usability Evaluation (Expert)	Cognitive Walkthrough
Practices	Evaluation	Usability Evaluation (Expert)	Heuristic Evaluation
Practices	Evaluation	Usability Testing	RITE
Practices	Evaluation	Usability Testing	Thinking Aloud
Practices	Evaluation	Usability Testing	Wizard-of-Oz

Table 9: List of additional codes that were omitted in Figure 4 and their respective hierarchy level