AIS Transactions on Human-Computer Interaction

Volume 11
Issue 3 *Special Issue on HCI in a Sharing Society*

Article 3

9-30-2019

Scope of Usability Tests in IS Development

Kimmo Tarkkanen
Turku University of Applied Sciences, kimmo.tarkkanen@turkuamk.fi

Ville Harkke
University of Turku, ville.harkke@utu.fi

Transactions on Human-Computer Interaction

Volume 11 Issue 3

9-2019

Scope of Usability Tests in IS Development

Kimmo Tarkkanen

Department of ICT, Turku University of Applied Sciences, kimmo.tarkkanen@turkuamk.fi

Ville Harkke

Information Systems Science, University of Turku, ville.harkke@utu.fi

Recommended Citation

Tarkkanen, K., & Harkke, V. (2019). Scope of usability tests in IS development. *AIS Transactions on Human-Computer Interaction*, *11*(3), pp. 136-156.

DOI: 10.17705/1thci.00117

Available at http://aisel.aisnet.org/thci/vol11/iss3/3

Transactions on Human-Computer Interaction

Research Paper DOI: 10.17705/1thci.00117 ISSN: 1944-3900

Scope of Usability Tests in IS Development

Kimmo Tarkkanen
Department of ICT, Turku University of Applied Sciences

Ville Harkke
Information Systems Science, University of Turku

Abstract:

Despite being a common, established concept in wide usage, usability tests can vary greatly in their goals, techniques, and results. A usability test that one purchases and performs for a specific software product may result in either minor user interface improvements or radical U-turns in development. Researchers have discussed such variation as a problem that concerns testing method's scientific reliability and validity. In practice, what "kind of data" one can expect to obtain from the selected method has more importance than whether one always obtains the same data. This expectation about information content or "scope" has importance for those who select and conduct usability tests for a specific purpose. However, researchers rarely explicitly state or even discuss scope: too often they adopt the premise that, because a usability test involves users, it brings the (necessary) user-centeredness to the design (i.e., takes socio-technical fundamentals as inherently given). We reviewed the literature on testing practices and analytical consideration and searched for the scope of a usability test that could deliberately approach the socio-technical tradition and equally develop both the system and the user organization. A case example represents a possible realization of the extended scope of usability test.

Keywords: Usability Testing, Scope, IS Evaluation and Development, Socio-technical Approach.

Mikko Rajanen was the accepting senior editor for this paper.

1 Introduction

Technology users in a professional work setting still run into situations in which they cannot use information systems, or such systems do not suit their work tasks at hand. Each poor use experience and usability problem with a system results in a loss somewhere: users' personal productivity and efficiency at work decreases, the software company for the system misses a potential user and paying customer, or users even endanger others (e.g., in healthcare settings). Each poor user experience and usability problem at the personal, community, and organizational levels requires professional usability evaluation and technology redesign. Apparently, when a problem manifests itself, the user research, user experience (UX), and usability efforts have failed or did not even take place before the software product entered the market or an organization implemented it and it finally evoked such poor experiences in users.

Information system (IS) researchers, developers, and software designers know that efforts to design and implement new technology for a professional and complex work domain will be most successful when one builds such technology on a firm knowledge about how users actually accomplish work in their everyday practices (Suchman, 1985). The software industry widely supports this user-centered development (UCD) ideology. However, actually implementing UCD into everyday development processes is difficult and laborious and depends on developers' personal attitudes and scarce organizational resources, which impair the breadth and depth of users' focus in developing the system (e.g., Steen, 2008; Bødker, 2006; Iivari, 2006).

As a concept, usability encompasses the attributes of the artefact in use and the purpose of the use. ISO 9241-11:2018 defines usability as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO, 2018). As such, it differs from user experience (UX) in the sense that UX does not need a specific purpose for artefact use and that the users create the experience in their minds. The usability of an artefact can be characterized as an attribute of the interaction between the user and the artefact.

Professional information systems' poor usability arises equally from usability methods and the evaluation activity itself. Usability evaluation methods lack design relevance and persuasiveness among IT developers and managers (e.g., Rajanen, Iivari, & Anttila, 2011) and have persistent reliability and validity problems themselves (Hertzum, Molich, & Jacobson, 2014). It often remains unclear how one should interpret the results from usability evaluations (Hornbæk, 2008), how one should inject these results back into the development process (Bernhaupt, Palanque, Manciet, & Martinie, 2016), and to what extent these results are reliable and generalizable to other contexts, users, and products (Reijonen & Tarkkanen, 2015). Usability problems found with usability evaluation methods may only confirm earlier impressions about the system in development(Hornbæk & Frøkjær, 2005). Thus, developers neither fix nor react to usability problems (Molich & Dumas, 2008).

Of the individual usability evaluation methods, usability testing constitutes the most popular and widely used among UCD practitioners and probably the best-known method among non-professionals. The usability testing method has the following main characteristics: it involves 1) prospective users who carry out tasks with the product and 2) evaluators who observe and record users' behavior in a short session[1] during which users provide feedback about the product. According to Hertzum (2016, p. 83) "usability tests may differ in their inclusion of conversational elements but share concrete system use as their defining characteristic".

In the context of complex problem domains, the usability testing method often has too traditional and narrow a focus and does not concentrate on reviewing users' actual work in these contexts (Redish, 2007). To affect design, usability tests need to mirror the problem domain's complexity and reveal issues that bring developers closer to a solution to the wicked problem. In practice that means questioning all that one knows in the design process thus far—testing the unknowns— by focusing on acquiring user knowledge for the development with a scope that covers not only the design artefact but also the whole spheres of use contexts and beyond to the value sources (see Cockton, 2004, 2006).

In order to apply any method, evaluators need to know what type of results they can expect from the method: what problems it can and cannot find well (Blandford, Hyde, Green, & Connell, 2008b, p. 283). One can evaluate such goodness in terms of the method's scope, which refers to "the kinds of issues it does and does not address" (Blandford, Green, Furniss, & Makri, 2008a, p. 395). In this paper, we define the scope

[1] A usability test refers to a short intervention if compared with other types of observational user studies where data collection easily takes weeks or months.

as the extent to which a usability testing method uncovers problems in the development context[2]. The scope determines whether the method fits or does not fit to the evaluation case and its objectives at hand—whether the evaluator should select a certain method over another. From usability practitioners' perspective, method validity conveys the scope. It is construct validity: what "kind of data" can one expect from the selected method? And does the method measure usability as it is required and understood in this particular development project? Usability interventions focus on improving a product step by step and usually without a need to replicate, compare, or search for similar results among different evaluators. Thus, practical reliability concerns the selected method's predictable behavior in different evaluation contexts and products such that designers and developers can understand and trust the data that the method produces. The scope concept and our definition for it helps evaluators to obtain such understanding—what they include and exclude due to their methodological choices—and to select and conduct usability tests for their specific purpose.

However, usability professionals often do not well articulate and hardly even discuss the scope of usability testing in its various forms. Too often, they view scope as highly abstract and a "black box". Usability tests involve users who give their (best) contribution and bring the (necessary) user-centeredness to the design and design process. Many software development projects in industry may assume that institutionalized prescriptions for conducting the usability testing method guarantee success in the design process (see Gray, 2016) and that, when rigorously followed, will lead one to automatically identify certain types of usability problems and design flaws (see Hornbaek, 2010). Thus, the developers take socio-technical design fundamentals as inherently given in usability testing due to its institutionalized status even when the testing has a limited scope. User organizations that source and outsource evaluation activities cannot rely on a well-established understanding of the testing scope in the market. Such an understanding may lead to standardized test procedures and unsuitable development and product requirements, which turns the user-centered design toward a discontinued and fragmented direction (e.g., Eshet & Bouwman, 2015). In the scientific literature and design science studies in particular, when validating the design, usability testing often embodies a limited scope that results in, for example, terminology mismatches, structural complexities, and redesign recommendations for individual user interface (UI) elements (see, e.g., Guay, Rudin, & Reynolds, 2019). In the project management literature, scholars well recognize that poor scope definition leads to project failure, an increase in costs, and a lengthier schedule (Cho & Gibson, 2001). In the usability research literature, scholars sometimes discuss usability testing's scope (e.g., Cockton 2004, 2006), yet they have not defined or presented scope itself and, thus, seemingly underrated it. For example, Reeves (2019) refers to scope when UX practitioners dissipate a found problem "by treating it as not in the scope of the usability test". Yet, Reeves (2019) does not specifically discuss scope, although he does describe in detail how UX practitioners look for troubles (usability problems) and how they produce findings in usability testing (i.e., how they construct the method's scope through their collaborative actions).

In this paper, we raise concerns in accordance to this track's theme: how can one deliberately consider both the social and technical aspects of IS design when conducting usability tests. Specifically, we examine usability testing's scope and how and why one can extend it. We proceed as follows: in Section 2, we discuss participatory IS development fundamentals. In Section 3, we position usability tests in IS development and discuss situations in which usability test's scope can (and often must) shift. Further, based on the literature, we take a look at different method modifications that have broadened the scope of usability testing according to the values of socio-technical approaches. In Section 4, we present our empirical usability test case in which we show how the development needs for both the system and the user organization can be captured. The case study exemplifies those minor changes in the usability test protocol that can extend the method's scope to deal with, for example, physical context limitations, complex social relations, and more traditional system deficiencies. In Section 5, we discuss the findings of the case study from the perspective of clarifying the scope of our testing procedure and the benefits of extending the scope. Finally, in Section 6, we conclude the paper.

[2] We build our definition based on the Oxford Dictionary of English, which describes scope as "1) the extent of the area or subject matter that something deals with or to which it is relevant 2) the opportunity or possibility to do or deal with something" ("Scope", 2005). We could equally comply with Reeves' (2019) simpler notion "the scope of 'what is being tested' or 'what the test is really about'" (p. 19).

2 Searching for the Scope of Usability Tests in IS Development

2.1 Involving Users and Organizations in IS Development

Designing products with users' participation has a long tradition, and ideas have evolved under different names and concepts throughout the years (e.g., Bjørn-Andersen & Clemmensen, 2017). However, all these concepts focus on developing a richer understanding about users' contexts and purposes and using that in the process of designing technology. The IS literature often refers to the Scandinavian tradition when addressing participatory design's roots and first projects (e.g., Kensing & Blomberg, 1998). One can characterize the original Scandinavian approach and related methods in that era as "utopistic": in ideal participatory design, all involved groups and individuals decide themselves how to develop their work in cooperation. Along with the increased cooperation and partnership with workers in systems development, the Scandinavian school emphasized the importance of designing tools that would fit into, rather than disrupt, the skilled crafts that workers developed over the years (Spinuzzi, 2002). From its beginnings, the participatory design community has focused on practices at the workplace as a core concern (see Kuutti & Bannon, 2014). One can see a legacy from Scandinavian participatory design in user-centered design (UCD). However, in the UCD methods, the ideal of equal power for stakeholders has been replaced by the absolute power of the system developer who decides what constitutes a well-designed system and controls the user involvement. As a consequence, the cooperative and participative nature have been reduced and institutionalized under a logic of technology development (Holmlid, 2009). In other words, in user-centered design, the user constitutes an information source and a subject rather than an equal partner (Sanders & Stappers, 2008). This artifact-oriented point of view does not seriously consider what sort of activities humans participate in when they use artifacts (Kuutti, 2011). In the design field, co-design and co-creation try to maintain the original idea of equal partnership by changing and mixing users' and designers' roles. Users work together in the design and development process although they have no training in design (Sanders & Stappers, 2008). Users' role varies from being informants to consultants, equal partners, and designers themselves (Kujala, 2003). Different methods and approaches used in the development allow users to participate in design with different quantity and quality (e.g., Steen, 2008, p. 4). Methods differ in whether potential users and stakeholders can express their needs and problems and determine the design's target.

Techniques to acquire knowledge about users and user communities prior to system implementation range from market surveys to interviews and on-site observations, which have their basis in common research techniques (Hyysalo, 2009). Kujala (2003) distinguishes ethnography and contextual design as two main approaches to involve users in systems design and development. Ethnographic studies focus on achieving such a shared view on the work and provide insights into the work's unarticulated aspects by applying open-ended (contextual) interviews and participant observations (Kensing & Blomberg, 1998). However, ethnography appears too expensive and too slow in an effective requirements' capturing for design purposes that require users to directly contribute to requirement specifications and development decisions (Stewart & Williams, 2005). Moreover, even with ethnographic inquiries, one cannot collect a perfect knowledge base for IT design that addresses all intricacies in use contexts and users' work practices. Thus, ethnography more represents a resource to other methods than a primary data-gathering method (Stewart & Williams, 2005.) Much of the research has concentrated on devising formal methods for involving users that "tends to overlook the interaction and knowledge-sharing in user–producer relations" (Heiskanen, Hyysalo, Kotro, & Repo, 2010, p. 495). Heiskanen et al. (2010) describe how formalized user-involvement methods are only first "eye-openers" (i.e., they sensitize designers to users' problems), while one needs to expend significantly more effort to sustain user-inclusive innovation communities. Certainly, not all kinds of user involvement lead to a successful product, and practitioners need all means to holistically understand the user perspective (e.g., Mattelmäki, 2006, p. 26).

2.2 Selecting a Proper Usability Evaluation Method

One can involve and collaborate with users during system development and in the later deployment phase in many ways (Johnson et al., 2014). Selecting an appropriate method constitutes an emergent problem, and applying a wrong method wastes money and resources (Hyysalo, 2015). The selected method should fit the particular case, the type of the product designed, how well designers can use the method, users' availability, and developers' knowledge about users and their context. In choosing an appropriate usability testing method, Bødker and Madsen (1998) advise that one "bring test situations closer to the nature of the future situation of use". According to them, the method choice depends on several characteristics of the evaluation situation: 1) the evaluation's purpose, 2) what one knows about the context, 3) whether one can access the workplace (the intended context of use), 4) available resources, and) available prototypes or other design artefacts. One may conduct evaluation to, for example, understand the current or future practice and context, obtain alternative ideas or obtain proof of existence for a particular artefact, test a particular solution, and identify important contextual issues (Bødker & Madsen, 1998). According to Blandford et al. (2008b), in essence, method selection includes the costs and benefits of applying any particular usability evaluation method. Costs include time and effort to learn and apply the method, whereas benefits include the insights obtained from using the method (Blandford et al., 2008b). They use the scope concept to determine the potential benefits from using a method. The scope refers to "what kinds of problems a method is and is not good for finding" (Blandford et al., 2008b, p. 283). One should not confuse the scope with the scale and the extent of an individual usability problem, which usually describe its local and global appearance in the system (cf. Dumas & Redish, 1999). The scope concerns understanding what type of results one can expect from the method that constitutes an essential determinant in selecting a method for a specific evaluation task and in understanding effects of our choices as evaluators on the evaluation results.

2.3 Positioning Usability Tests within IS Development

A usability test conducted during the software development process represents one type of knowledge-elicitation intervention with future users and user organizations. In its classical form, usability testing does not focus on eliciting users' conceptual models or their activities but on evaluating the system against the set usability goals, detecting software product's usability problems, and recommending correspondent changes to the system's design (Wixon & Wilson, 1997).

Sanders (2006) positions usability testing under UCD methods that emphasize experts' mindset over participants' mindset (Figure 1, diagonal axis). In these methods, designers try their best to understand users' world and "design for people" (e.g., contextual design, applied ethnography). Thus, designers move towards users (Steen, 2008). Designers do not consider users as partners but as subjects and reactive informers who lack power in the process (Sanders, 2006). Traditional usability tests represent the expert mindset approach in many ways. Usability experts organize and coordinate a study in time and place, define and recruit an appropriate group of target users, determine the goal-oriented tasks that users will perform with a product, and investigate and interpret the results (Sullivan, 1989). Thus, the experts remain fully in control. In contrast, in methods that emphasize a participatory mindset, designers "design with people" who act as partners and active co-creators (e.g., lead user approach, co-design, Scandinavian participatory design); that is, users move towards designers, and the intimate communication between them originates from the users (Steen, 2008).

Another dimension in product design versus user research (Figure 1) describes what the method concerns (Steen, 2008) and, thus, reflects scope as well. The user research-focused methods focus on the current (i.e., "as is") situation, whereas the product design-focused methods focus on envisioning the future and alternative (i.e., "to be") situations (Steen, 2008, p. 32). Methods that emphasize the user research orientation carry interest in exploring users' current situations and the use contexts in order to first find out what design users need and why they need it. In contrast, product design-oriented methods (at their purest) consider the technological artefact as an end itself and begin searching for "the new thing" without first exploring whether any needs exist for it. Thus, product design represents "a mild form of technology push" (Steen, Kuijt-Evers, & Klok, 2007). Exploring with users (participatory mindset) and for users (expert mindset) revolves around the design artefact—refining what it should be like and how it can be improved.

Usability testing inherently focuses on designing the product rather than on its users. Usability tests rests on the rationale that evaluators perform it in order to detect and correct the usability problems with the IT artefact. The product vision—the to-be state—naturally and tangibly exists in the form of the new product design that one tests. The focus on product suppresses users' importance to only a few relevant and pre-selected aspects concerning the design. The social and organizational setting, the use context, and the activities in which users engage when using the artefact do not form usability testing's essence, but one usually studies them before conducting the usability test, which then can focus on testing only the IT artefact. This focus on the product also directs how one conducts usability testing towards the later stages of the design process where the artefact better allows one to test it. Moreover, usability evaluations mostly address the fit between the system and the individual user because usability professionals construe the usability concept at the individual level (Hertzum & Clemmensen, 2012). However, the individual view may not match with "organizational usability" (i.e., the socially acceptable and effective way to integrate the system into the work practices of employees in the organization) (Elliott & Kling, 1997). Evaluating systems at work prioritizes users and their work activities before assessing the potential support from system characteristics. By describing these links between the systemic and contextual characteristics affecting usability, not only the system redesign but also learning and change processes in the user organization could occur (Nurminen, 2006, p. 414).

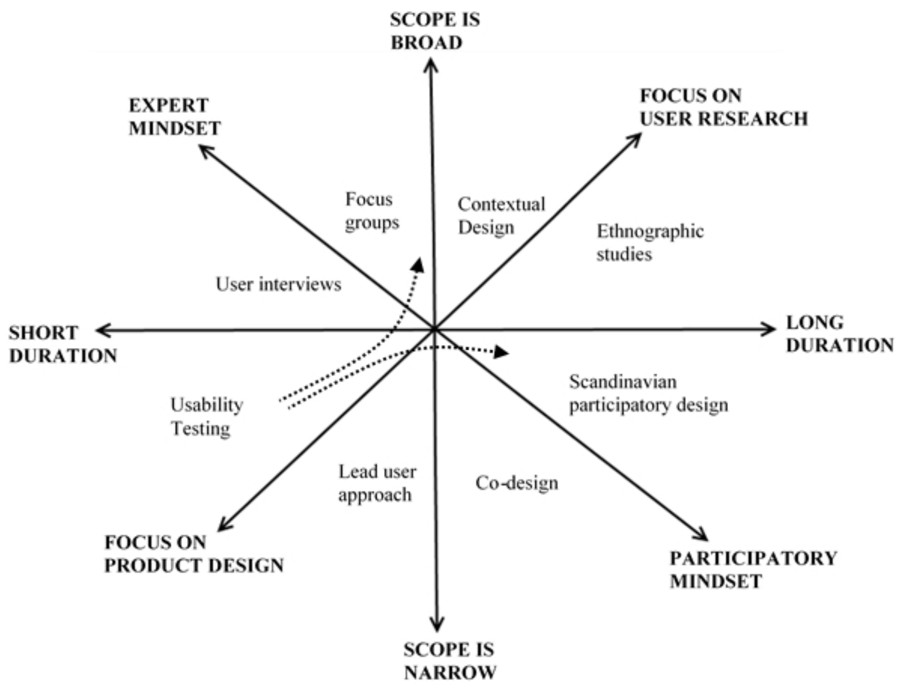

Figure 1. Scope of Usability Testing among UCD Methods (Inspired by Sanders, 2006; Steen, 2008; Blandford et al., 2008b)[3]

3 Practical Modifications of the Scope of Usability Tests

Various method collections, combinations, and modifications expand traditional usability testing's scope in order to better understand the user and the work domain beyond the user interface. One common driver has been the notion that conventional, designer-created usability test tasks, which have correct answers and clear endings, only weakly answer usability questions concerning user needs and situated work practices (Redish, 2007). Therefore, to understand and test complex work systems, Redish (2007) suggests using method collections and combinations such as conducting usability studies in users' field, exploiting multiple evaluators, building simulations, developing situation-awareness assessments, implementing ways to capture data in the long term in an unattended way, and using cued retrospective think-aloud method with users.

Similarly, in searching for indicators of complex work systems' usability, Savioja and Norros (2013) found out that a traditional usability test focuses only on measuring how well users perform activities with the tool. Thus, such a test lacks scope: it does not consider work practices (way of acting) and how the tool psychologically and communicatively functions. Based on activity theoretical foundations, they propose a contextual evaluation approach called contextual assessment of systems usability that simulates tool use according to the scenarios modeled with functional situation representations (a type of an extended task analysis). The method's scope focuses on the different perspectives on and levels of work activity and tool support when evaluating usability. Similar to Redish (2007), Savioja and Norros' (2013) approach leans on more than one data-collection technique (e.g., it employs usability questionnaires, interviews, observations, task-load measures, and expert judgments before and after the simulation sessions).

[3] Dashed arrows describe the scope expansions that we discuss in Section 3.

Følstad and Hornbæk (2010) used a method called the cooperative usability test to gain knowledge about a work domain. At its essence, the test contains several interpretation phases after each task. An interpretation phase asks why users acted in a certain way and, thus, uses their knowledge about the work-domain to identify and understand usability problems. Thus, the interpretation phase has similarities with the debriefing phase that Rubin and Chisnell (2008, p. 229) discuss. At their foundation, interpretation discussions have a task-scenario walkthrough where users can comment on the parts of the system that they did not use in performing the tasks. As a result, the method's scope includes system requirements that developers may not have responsibility for. Similarly, Spool (2006) exploited a more communicative testing approach with users in which they did not assign test tasks to users but discovered them via interviews. However, the method limits interview-based tasks to the Web shop context. While this modification expands usability testing's scope to "insights about users' domain of interest", it remains rather product design oriented. The method focuses on identifying "passionate" users, giving users the most realistic test tasks, and learning users' terminology. This information helps one in reorganizing a website's content (Spool, 2006).

Kankainen (2002) experienced that traditional usability testing with predefined tasks did not work in obtaining user feedback when evaluating a product's early design. Users evaluated only the interface and said little about the overall product concept, which made it a torture to users and designers alike. Kankainen (2002) modified the testing into co-discovery exploration and presented the new design concept with a storyboard and a blank model with accessories. In consequence, the scope and users' focus turned towards the overall product concept rather than the interface as such, which proved more useful and inspiring for later design. Similarly, Still and Morris (2010) applied a blank-page technique when testing paper prototypes' usability. They allowed users to navigate to non-existing pages and dead ends while they encouraged users to create and design the content for these empty spots. The technique expanded the scope by 1) giving insights into users' mental models and 2) how they conceptualized information encountered.

Blandford et al. (2008a) developed "a concept-based analysis of surface and structural misfits" (CASSM) method due to a finding that compromising between a fully naturalistic study and a conventional lab-based study protocol could not identify mismatches between user requirements and system representations (i.e., evaluate enough the utility). Users need to work with the concepts in the system, which, when poorly fitted, may place a high workload on them (Blandford, 2013). For example, when booking flights, test participants care more about operating with journeys between places than with flights between airports (Blandford et al., 2008a). Thus, the CASSM method helps one to identify how users conceptualize a domain prior to system implementation: it extracts and compares concepts that a user uses to the ones the system implements. One collects the user data with a think-aloud protocol or similar approach, which provides knowledge about users' procedures for completing the tasks. Thus, CASSM represents an analytical usability-evaluation tool more than an empirical-testing and data-collection method. However, it broadens the think-aloud protocol's and usability evaluation's scope to look at profound misfits in the underlying structures, which, when found, represent typically new design opportunities for the product. Similarly, Johannessen and Hornbæk (2014) expanded the analytical usability evaluation by creating an expert inspection method that focuses on finding utility issues and problems when users use a system.

Concentrating on utility issues on the empirical side, Juurmaa, Pitkänen, and Riihiaho (2013) modified a visual walkthrough method to find elements in the user interface that users consider important or useless. Among other flexible modifications of usability testing methods, Riihiaho (2015, 2009) introduced two more walkthroughs (informal and contextual). The latter evaluates a system's usefulness in a professional work setting. Thus, at its core, it evaluates the real use context with real data and lets the test tasks arise from users. Similarly, Bødker and Grønbæk (1991) added realism to the evaluation by having everyday materials and tools available to users alongside the tested product. Users could demonstrate their current role when going through a typical work task due to the work material brought into session (Bødker & Madsen, 1998). According to Riihiaho (2015), contextual and informal walkthrough methods can tackle the bias in predefined test tasks. For example, in examining a call center, Riihiaho (2009) found that other unintegrated applications affected the use situation of a new application and that the other applications' physical location became the biggest problem rather than the new application's usability. In other test cases, informal and contextual walkthroughs have revealed 1) terminology mismatches, 2) technical infrastructure problems, 3) discontinuities in task flows, 4) missing functions, 5) user misunderstandings, and 6) concerns about post-usage behavior, which all probably would have not been in scope if researchers applied only traditional usability test tasks (Riihiaho, 2009).

Similar to the contextual walkthrough above, McDonald, Monahan, and Cockton (2006) collected data with a contextual interview in which they asked participants to carry out their normal work tasks with the system while thinking aloud. In contrast to the informal walkthrough above, the moderator occasionally interrupted the participants to ask questions for clarification purposes. The authors exploited an affinity wall to extract usability problems from the overall data, which is more traditional in contextual design. As a result, about two-thirds of the problems they found concerned not the system they evaluated itself but 1) other applications in use (e.g., email), 2) lack of user training, 3) insufficient documentation, and 4) the technical environment, and 5) the physical environment. Based on the extended scope of the problems they found, McDonald et al. (2006) conclude that studies in the laboratory premises iterate our understanding about design artifacts, whereas testing in the field iterates our understanding about the use context and products' intended value. However, this generalized conclusion favors the testing environment (field/lab) and does not consider the effect of unstructured test protocol and work-originated tasks on the test results (cf. Reijonen & Tarkkanen, 2015).

In their ethnographic model of field usability testing, Rosenbaum and Kantner (2007) also borrowed from contextual inquiry practices. They applied both the same predefined high-level tasks for each participant and lookup tasks that "were of the participant's own choosing and thus were unique from session to session" (p. 5). With the latter tasks, users had more interest in the task and the outcome. Rather than usability metrics, the results from testing online banking for vision-impaired people took the form of "cases" and "scenarios". These results addressed 1) the variety of use strategies, 2) behavioral trends, and 3) utility issues. According to Rosenbaum and Kantner (2007), field testing best suits exploratory objectives where evaluators "want to learn what problems users encounter as they follow their own work processes" (p. 2). However, they suggest using contextual inquiry and ethnographical interviews when one primarily wants to understand what people really do with the products or to explore which new features to add. In this manner, Viitanen and Nieminen (2011) pre-explored users' work practices with the contextual inquiry method before combining an interaction sequence analysis to their usability test. A user research method called "guerrilla testing" involves the artifact in the pre-exploration and represents a quick way to validate design's effectiveness among its intended users and whether the design works in the way it should.

Åborg, Sandblad, Gulliksen, and Lif (2003) built an approach called ADA (Användbara datorsystem) around users' ordinary work tasks and natural test settings to address both usability and work environment aspects at the same time. At its core, the approach views work and tasks as larger units. Although the approach does not pre-define the task assignment, evaluators need familiarity with the "aspect" list, which defines in detail what they observe during the session. The list emphasizes user interface issues but also exhaustively lists user and systems-in-use factors (e.g., user's role, tasks, competence, system functionality, manuals). However, the predefined list of observable aspects limits the scope and may be irrelevant when evaluating early prototypes. Thus, the method primarily fits efforts to evaluate how users use systems each day.

When testing software prototypes in work with an open-ended nature (e.g., artistic, creative and knowledge-intensive domains of work), Sy (2006, 2007) pre-explored workflows for a future design by interviewing users on the telephone and began test sessions with contextual investigation. In open work domains, she found that "scripted usability tasks often set unrealistic constraints on user behaviour that don't match the open-ended nature of the task" (p. 18) and suggested using open-ended test tasks for more realistic results. Her technique constitutes "a way to sneak contextual inquiry into a usability testing" (p. 21) where evaluators use closed test tasks only for non-workflow-specific design goals. Open-ended test tasks, which start from a high-level activity that would cover all the tasks that one wants to validate, depend on pre-interviews. In the test session, users lead and evaluators direct then only if they need to validate a certain design goal. Tests with open tasks in open work domains include the following items in their scope: 1) contextual information about users' workflows (especially unexpected uses of the product), 2) examples of users' work in the application, 3) feature requests, 4) major usability problems, 5) bugs, and 6) successes with the design prototype (Sy, 2007).

User experience research probably applies open-ended and user-initiated tasks more widely than usability research does (cf. Bargas-Avila and Hornbæk, 2012) because exploring true experiences requires systems that one can "let loose" into people's everyday practices and lives where evaluators cannot give or control detailed instructions (Buchenau & Fulton Suri, 2000). As its core dimension, user experience research focuses on a system's hedonic qualities, such as emotions and affect, enjoyment, and aesthetics (Bargas-Avila & Hornbæk, 2012), which may not always involve goal-oriented activity in contrast to business application domains and professional work settings on which we concentrate. One objective in using open-ended test tasks in the business application domain, as Sy (2007) has demonstrated, concerns understanding users' goals and means at work and systems' applicability to them.

All method modifications that we introduce in this section have their foundations in empirical usability evaluation methods with users and in the problems and challenges that traditional evaluation practices may confront in product development. The methods we introduce all seem to have a common denominator: they require one to shift from a highly expert-minded usability evaluation towards a more user participative evaluation practice, which gives room for users to explore the system based on their needs, wants, expertise, and experience (see Section 2, Figure 1). Further, the methods seem to support that evaluators apply more user research-oriented and "ethnographic" goals than focus on designing and the artefact as evaluators' only frames of reference. Consequently, all these methods more or less end up being methods for evaluating systems quality in context due to their extended scope and focus that extends beyond usability and that provides valuable and wide-ranging results for the subsequent development process. The extensions even improve the testing methods' suitability in the design process's early stages, which decreases the risk that one will need to make dramatic (and expensive) changes later in the design process. Thus, a question may arise whether these methods concern usability and usability testing or IS quality evaluation in general. Third, these methods commonly feature modifications that evaluators create not only to expand the scope per se but also to achieve a better fit with the current design process and its challenges at hand and to increase the results' design value for the specific project.

In summary, we conclude that evaluators need to expand the scope of usability testing to areas that are outside the interaction between the system and the individual user. Further, although all scope extensions seem to complement and validate user research efforts in the development, they represent scope in a varied way and disperse these representations into a mixture of case-specific result descriptions. Similarly, the vast majority of usability tests that researchers report as part of design science activities in scientific literature do not specifically focus on developing the evaluation method or discussing its scope.

As a continuation to the methods we present above, we introduce our findings from conducting an empirical usability test case in Section 4. In the usability test, we applied an open-ended test task as a test protocol that focused on broadening usability testing's scope beyond the technical to the system value in human and social context similar to the methods we introduce in this section. Here, the open-ended test task (or open task for short) means a task assigned to the user that contained only a request to use the system with minimal explanation about the context and the system's purpose. In its shortest form, the task constitutes a short request (i.e., "please do something"), while, in the other cases, the intended purpose or the low fidelity of the system requires one to define a starting point for use. Many usability evaluations and testing methods that we discuss in this section integrate other methods in parallel or are analytical or expert evaluations. In contrast, the open test task modifies only the usability testing method's internal parts, and, unlike Sy (2006), we apply the open task in the complex professional health care domain rather than unregulated and creative work domain. The case study introduces the extended scope of usability testing with the open test task method. It also constitutes one possible way to represent usability testing's scope in general.

4 A Case Example on the Extended Scope

4.1 Method Description

Our open task test took place in an IS development project, which had the purpose to design a mobile application for nurses in hospital wards. Here, we call this application "Round". Few studies have investigated the impact that mobile EPR tablets have on clinical routines at hospital wards and the underlying mechanisms that help people who use such tablets save time (Fleischmann, Duhm, Hupperts, & Brandt, 2015). Round provided an interface to the electronic patient record (EPR) system in use at the time, and, thanks to its mobility, the application allowed nurses to instantly access the EPR system when they worked with patients in their rooms. We conducted the first usability test of the application with six nurses at one hospital ward. In the test, one operated Round, a fully functional demo only at the interface level, with a mobile tablet device. Two Round developers (a UX designer and a system architect) followed test sessions and could intervene. The participating nurses used the application for up to an hour-and-a-half in front of the table in the hospital premises. We gave the participants only the following open-ended test task: "You have just arrived at your workplace and you begin to prepare your work shift. Round is a new application that you can use during your work. (You have already logged in).". Relatively little mobile usability research has examined open and unstructured test tasks of this kind (Coursaris & Kim, 2011). Two test sessions expanded such that the nurses measured a real patient's blood pressure and heart rate. After the test, we arranged a short meeting with the developer representatives to discuss the first insights and initial results. A full report delivered one week after introduced 57 usability problems. In addition to a list of problems and recommendations, the report included a description of the common phases that nurses go through in their working day (i.e., what they do, why, when, and what results from the work at the hospital ward).

In representing and articulating the scope of usability tests with open test tasks and the results from this particular case, we use the problem-classification schemes similar to the user action framework (UAF) (Andre, Hartson, Belz, & McCreary, 2001) and classification of usability problems (CUP) scheme (Vilbergsdottir, Hvannberg, & Law, 2014). However, we do not follow any pre-existing problem classification or values of failure qualifiers in order to keep the origins of the analysis purely in our empirical data and to go deeper into the subject of scope. In analyzing the data, we reviewed and grouped the usability problems we found (i.e., gave each problem a category to abstract similar problems into groups). In determining the categories, we concentrated on analyzing problems (in the system) from users' point of view. The names of the categories in Section 4.2 reflect that view. Here, each problem category represents the scope (i.e., the extent to which our method uncovers problems in the IS development), although we note that other possible representations could exist. Both researchers coded and grouped problems independently. As a result, an interrater reliability showed 78.0 percent agreement between the two researchers and Cohen's kappa 0.742 (Cohen, 1960). The result means a moderate agreement in coding and reliable agreement percentage value since we made no categorization randomly (McHugh, 2012). In Section 4.2, we discuss these problem categories as a representation of the method's scope and provide examples of findings in each category.

4.2 Results

The usability test produced a lot of information about the work practices at the hospital ward. Most importantly, we identified the system's missing, inadequate, and the problematic functionality and highlighted previously unexplored design options that would bring value for users and induce positive changes in their current work practices. Below, we provide practical examples in each problem category that, as a whole, form the scope for our test in the case.

4.2.1 Previously Unexplored Design Option in the Context

During the test, we identified that major proportions of care actions and their documentation needs shared fundamental similarities. The EPR at the desktop PC did not support such a unified view on documentation but diversified the care documentation into separate system modules and dialog windows that each focused on different care actions. Despite the slight differences in care documentation between different care tasks, the work on the ward and Round use would become more effective if Round supported similar and consistent design patterns for all care tasks as much as possible.

4.2.2 Problematic Change of Work Practice due to System Implementation

With Round, one can assign care tasks to specific nurses and other users can see the task completion rate. Despite the possible benefits from structuring tasks, failed, neglected, delayed, or incomplete care tasks could lead to unpleasant social pressures in the community and discrimination among nurses. The nurses also considered that assigning work tasks might hinder and impair their (currently high) personal autonomy in planning different work duties, such as how they want to coordinate tasks between their co-workers, how they personally want to perform these tasks (e.g., in a certain order), and when they want to perform them (e.g., measuring blood pressure in the morning).

4.2.3 Missing Functionality

In the beginning of the test and the nurses' imaginary work shift, they would have liked to print the free form notes they entered earlier into the PC-based desktop EPR system onto paper. These notes concern patient's health condition, physical abilities, reasons for admission, and so on. Nurses were used to carrying these notes in their pocket during the day. During the patient work, they wrote new notes on the paper, which they update to the desktop EPR later. When we initially began the test, the nurses could not simulate their work with Round due to the missing annotation field, which correspond to these notes on paper. We developed Round to replace such a manual task and overlapping documentation by decentralizing the notes under specific care tasks in a structured representation form. On the one hand, nurses thought they wanted to decrease double documentation, but, on the other, they considered the centralized overview on patients' health status important as well. In the test report, we could ponder both design options in Round with and without the notes field, examine their effects on nurses' practices and organizational system implementation, and identify the types of notes missing from the current prototype.

4.2.4 Inadequate Functionality

Nurses could filter patients by ward and by nursing team with Round, but, most of all, they needed filtering "by pairs" as a third option because the hospital organized their work on the ward by pairs in the same shift. Thus, the filtering function, although implemented, lacked a proper fit with the community's needs and work practices.

4.2.5 Unfinished Physical Use Context

In addition, this case concretized some limitations in applying open tasks in a simulated environment compared to testing with real patient work and patient data. Two out of six participants took the application into clinical work partly on their own initiative (possible due to open task). Therefore, we could observe the limitations in the physical device and environment that we would not have observed when sitting at the table without a contact with patients. First, the nurses could not feasibly carry the tablet-based device with other care equipment. Inpatients at hospital wards vary in their physical health condition that restricts their ability to move, which naturally implies that nurses and physicians constantly move from room to room while caring the patients. In the hospital ward we studied, the room doors are closed and rather heavy to open. Therefore, the nurse could hardly open the door with the tablet on the one hand and a blood pressure meter on the other. She opened the door with a little finger (see Figure 2). Second, when the nurse began to measure the patient's blood pressure, she did not find a proper place to put the tablet device down (Figure 2). Since patients can use the tables next to their beds, the tables may not be free for the tablet device. Clearly, one cannot easily find usability problems related to the mobile device's physical appearance in a test at a desk even when using the open task approach. Possible solutions to these problems lie not only in software developers' hands if at all. Software developers could implement the application in a smaller device (e.g., smart phone version), yet one would need more comprehensive design and user organization involvement to provide more table space next to patient beds, sewing larger pockets for nursing jackets, purchasing carrying bags for equipment or keeping doors open at the ward, and so on. The user organization with its practices and policies constitutes an equal a key stakeholder in improving information systems' usability.

Figure 2. Opening the Door with the Little Finger (L); The Table for Patient's Personal Use (R)

4.2.6 Unexpected Situations in Service Work

Third, while measuring patients' blood pressure, one nurse found that the application did not allow her to record saturation and C-reactive protein (CRP) values. She did not notice this missing functionality when simulating her work and personal practices at the table. Although the other test participants mentioned that the application lacked these features, we understood that participants cannot always exhaustively simulate the work and personal practices in lab-like premises without real patient contact. Fourth, even when testing with the open task approach, individuals easily treat work tasks as separate entities that follow the order in the test subject's mind at the time of the session. In any service type of work, task flow depends on the client side as well and sometimes includes unanticipated turns and "jumps" that may not become visible in tests that one performs in lab-like premises. For example, due to discussions with patients, the blood pressure-measurement task expanded to include two other tasks (specifically, tasks related to patient medication). The nurse could not anticipate (or remember to ask in the first place) that the patient needed both a painkiller and a digestive medication. Both new tasks needed official medication record entries, a note for the nurse herself, and possibly a note for the nurse's colleagues sharing the care tasks with her (because somebody needs to remember to give the medicine). However, the nurse did not turn to the Round application even though it would have helped her with the tasks. Instead, she went to the office to use the desktop computer. Clearly, participants are not always ready to use a new design in a familiar situation—a natural occurrence given that they cannot know all a new software's possibilities and the open-ended task approach does not offer only doable test tasks or give hints what actions one needs to take next. Thus, participants' behavior requires careful attention from the evaluator, possible intervention, and correction after observing and recording their initial actions and aims with the new system.

In order to verify the impact that the reported findings had on the design, we conducted a retrospective interview with one developer one year after the study. The interview concerned the usability study's usefulness and realization, the report, and the state of Round's development. Together with test recordings, documentation about problems and requirements, discussions and interviews with developers, and published system descriptions, we developed a detailed picture of the system development and could investigate the possible impact that the open task usability study had on the application's design. We concluded that 15 problems (out of 57) had no impact on Round's design (requirements specification document or final implementation). Unfortunately, we could not access the finished product and observe how users actually used the developed system. From doing so, we could have obtained additional insights since the way users actually use a system may change after some time (Tarkkanen, 2009).

5 Discussion

Usability testing is a well-known concept in software product evaluation. Due to its institutionalized status as the must-be method, its scope—the extent to which it covers problems in the development—would appear to be well defined. Our literature review, however, clearly indicates that usability testing does not constitute a singular method with one well-defined scope, which evaluators could always lean on or even know in advance when applying it in practice. If we observed the UX industry practitioners at work, we would find many forms of usability testing practices and different scopes. Organizations who develop and purchase information systems and outsource evaluation activities should not accept a vaguely defined testing without questioning it, nor can these organizations afford to overlook problems that do not fall in traditional usability testing's scope.

The methods we identified from the literature review also show that many elements other than the test task or scenarios that one provides during such a test affect a method's scope (i.e., the main modification we made in the case study we present in this paper). However, as the case study example shows, diminutive changes in the test protocol, such as the open-ended task itself and the subsequent possibility to use the system in a real interaction, can radically expand the scope towards new areas. In contrast to other methods that we discuss in Section 3, we applied both our modifications *in* usability testing without additional pre- or post-phases or deliberately attached methods. The former modification, the open test task, set the basis for our findings, which not only concern human-computer interaction but also cover users' complex social relations and concerns related to possible social changes due to system implementation. The latter modification had less importance than the open task and took the case method's scope to deal with physical limitations that the device and the environment set (about opening the door and the lack of space on the table). Thus, the case study represents a scope that one cannot usually achieve with tests in lab premises. Both modifications brought findings that neither future designers nor the user organization could bypass in order to make the system effectively, efficiently, and satisfactory usable for the nurses in the hospital ward.

Especially in complex work domains such as healthcare, any system testing is inadequate if it omits the effects of the work context and factors outside the tested system itself. Usability researchers and practitioners understand this well and they have established methods that extend the scope outside the system. Most extensions rely on using multiple methods on multiple occasions and add cost and complexity to the testing phase. Even though a clear need for different knowledge-gathering methods during the design and development software development phases exists, the testing sequences could be improved and the information collected during the actual testing could reduce the need for multiple methods.

Readers may see the extension we describe here as an attempt to improve the available methods for the evaluation/development stages in an IS design and development process. As an artefact itself, we have verified the extended scope open task method only in a few instantiations; as such, it remains merely promising alternative for improving evaluations. In a broader perspective, extending testing's scope may guide general design science research towards models and methods that identify problems and objectives beyond the mechanistic single task/single user formulations without adding the time and effort that full ethnographic surveys require. That said, the extended testing scope naturally suits only some design processes and does not make other methods obsolete; rather, it represents another step in the evolution of the methods we use to understand the nature of the things we create and use.

The open task method, among the other methods we discuss, represents a step towards realizing the HCI standards that originate from participatory IS design tradition and user empowerment and try to maintain user-centeredness along the whole product development lifecycle. Open test tasks deliberately shift the usability evaluation's focus onto users' everyday practices and result in broad scope that reflects the values of the socio-technical IS development traditions. Open tasks represent a top-down approach in evaluation that tests our (and designers') representations of work, our understanding about use contexts and users, and requirements for the product.

Based on the case study, the scope of usability testing with open tasks seems to cover the system utility aspects and the more traditional usability problems equally well (see Tarkkanen, Harkke, & Reijonen, 2015). Open task method has a scope that finds problems that 1) render doing a job with the system impossible (missing and inadequate functions), 2) can often cause unfavorable and uncontrollable consequences in users' work (problematic social changes), 3) require more user research and context exploration for more benefits (unexplored design opportunities) and problems, and 4) that cause inefficiency and unsatisfied users (physical limitations and unexpected situations in the service work). The scope covers problems that comply with the classical definition of usability, whereas other problems contradict users' goals and tasks. With the former problem type, designers can produce alternative solutions without challenging their understanding about the use context and the collected requirements. Problems related to the utility scope usually require deeper user research. Practical usability tests vitally need to address both these system-usefulness aspects because possible solutions to the problems differ. One could map these findings to the ISO 9241's broader definitions so that the missing and inadequate functions fall directly under the *effectiveness* category, the uncontrolled or unwanted changes in work effect fall under the *efficiency* category, and the physical limitations and unexpected situations fall under the users' *satisfaction* category. The unearthed unexplored design options constitute the open task extension's main benefits and deepen designers' knowledge about the *specific context.*

Here, the scope is tightly intertwined with the expected outcomes—the two terms are practically synonyms (due to our data-analysis practices). However, one may want to represent the method's scope in other terms as well. For example, Reeves (2019), although not speaking with the term "scope", observed findings from a usability test from four different "relevancy devices". Some user troubles become insights, other become issues or recommendations in the final report, and others still become dissipated through the discussions in the observation room. Accordingly, we see scope not only as resulting from method prescriptiveness but also as resulting from such a collaborative work between stakeholders towards producing findings (i.e., as resulting from a positive evaluator effect).

Our case would easily stretch also to a more abstract materialization of scope. One could present scope as (the number of) usability findings targeted at the technical system, the social context, and the physical environment. However, one would need to recognize and articulate these targets regardless of the classifications and formats one used to represent and describe a method's scope. In any case, identifying a method's scope calls for more than a basic method description that contains the method, its strengths and weaknesses, and possible usage phases. Every evaluator and researcher can take similar analysis efforts after the studies they conduct, and, eventually, such work would serve the evaluator community. However, we acknowledge that one cannot thoroughly describe a method's scope—the method's outcomes and boundaries; the work will never finish, and it would rather lead to a situation that Gray (2016) describes in which "a designer would have to make decisions about the limits of the method in situations that are explicitly coded for". Despite our representing scope here as problem categories, following Gray (2016), we also consider methods as merely tools and players in a design game rather than an objective set of outcomes. Even if one considered methods prescriptive but situated, the discussion about the scope still has relevance because each test outcome begins a new design iteration and represents an opportunity to learn.

6 Conclusions

In this paper, we discuss the usability testing method's scope in IS development. One needs to understand the method's scope to understand its validity and effectiveness and, subsequently, to select the right method for the evaluation case at hand. In this paper, we discuss differences in usability tests' scope and how the elements and applied usability-testing protocols affect it. From a theoretical perspective, this study contributes to shifting usability testing's scope towards a more participatory- and user research-centric direction, which IS evaluation and IS development practices fundamentally require but that many easily dismiss. The literature has shown for a long time that usability testing does not naturally implement users' and organizations' view, which includes wider socio-technical design dimensions. Therefore, one should not take users' and organizations' views for granted in usability testing but deliberately attach them to the method performance requirements when needed. With this paper, we contribute to practice by introducing the literature's scope-broadening method modifications, the usage and value of which the case study we present further exemplifies. Although one cannot find only one scope for usability testing based on this study, our study does imply that the discovered and experimented shift in the usability testing method's scope is both possible and valuable in practice. By cutting the link between the design and the evaluation process, the methods we present (and the case method specifically) serve both technology developers and end-user organizations equally and rather cost-effectively. As the case study shows, only a moderate change in the test task towards openness allows user control and freedom in the test session, which further reveals, for example, unexplored design options and problematic future changes at organizational and community levels of work. Thus, the open-ended test task introduces one possible realization of usability testing, which considers organizational and social factors beyond individual users.

References

Andre, T. S., Hartson, H. R., Belz, S. M., & McCreary, F. A. (2001). The user action framework: A reliable foundation for usability engineering support tools. *International Journal of Human-Computer Studies, 54(1)*, 107-136.

Bargas-Avila, J., & Hornbæk, K. (2012). Foci and blind spots in user experience research. *Interactions, 19(6),* 24-27.

Bernhaupt, R. Palanque, P. Manciet, F., & Martinie, C. (2016). User-test results injection into task-based design process for the assessment and improvement of both usability and user experience. In C. Bogdan, J. Gulliksen, S. Sauer, P. Forbrig, M. Winckler, C. Johnson, P. Palanque, R. Bernhaupt, & F. Kis (Eds.), *Human-Centered and Error-Resilient Systems Development* (LNCS vol. 9856, pp. 56-72). Berlin: Springer.

Bjørn-Andersen, N., & Clemmensen, T. (2017). The shaping of the Scandinavian socio-technical IS research tradition: Confessions of an accomplice. *Scandinavian Journal of Information Systems, 29*(1), 79-118.

Blandford, A. (2013). Eliciting people's conceptual models of activities and systems*. International Journal of Conceptual Structures and Smart Applications*, *1*(1), 1-17.

Blandford, A., Green, T. R., Furniss, D., & Makri, S. (2008a). Evaluating system utility and conceptual fit using CASSM. *International Journal of Human-Computer Studies*, *66*(6), 393-409.

Blandford, A. E., Hyde, J. K., Green, T. R., & Connell, I. (2008b). Scoping analytical usability evaluation methods: A case study. *Human-Computer Interaction*, *23*(3), 278-327.

Bødker, S. (2006). When second wave HCI meets third wave challenges. In *Proceedings of the 4th Nordic conference on Human-computer interaction: Changing roles* (pp. 1-8).

Bødker, S., & Grønbæk, K. (1991). Cooperative prototyping: Users and designers in mutual activity. *International Journal of Man-Machine Studies, 34,* 453-478.

Bødker, S., & Madsen, K. H. (1998). Context: An active choice in usability work. *Interactions*, *5*(4), 17-25.

Buchenau, M., & Fulton Suri, J. (2000). Experience prototyping. In *Proceedings of the 3rd conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (pp. 424-433).

Cho, C.-S., & Gibson, G. E., Jr. (2001). Building project scope definition using project definition rating index. *Journal of Architectural Engineering*, *7*(4), 115-125.

Cockton G. (2006). Focus, fit, and fervor: Future factors beyond play with the interplay. *International Journal of Human-Computer Interaction*, *21*(2), 239-250.

Cockton, G. (2004). From quality in use to value in the world. In *Proceedings of the International Conference on Human Factors in Computing Systems* (pp. 1287-1290).

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37-46.

Coursaris, C. K., & Kim, D. J. (2011). A meta-analytical review of empirical mobile usability studies*. Journal of Usability Studies, 6*(3), 117-171.

Dumas, J., & Redish, J. (1999). *A practical guide to usability testing*. Exeter, UK: Intellect Books.

Elliott, M., & Kling, R. (1997). Organizational usability of digital libraries: Case study of legal research in civil and criminal courts. *Journal of the American Society for Information Science*, *48*(11), 1023-1035.

Eshet, E., & Bouwman, H. (2015). Approaching users and context of use in the design and development of mobile systems. In A. Marcus (Ed.), *Design, user experience, and usability: Users and interactions (*LNCS vol. 9187, pp. 508-519). Berlin: Springer.

Fleischmann, R., Duhm, J., Hupperts, H., & Brandt S.A. (2015). Tablet computers with mobile electronic medical records enhance clinical routine and promote bedside time: A controlled prospective crossover study. *Journal of Neurology, 262*(3), 532-540.

Følstad, A., & Hornbæk, K. (2010). Work-domain knowledge in usability evaluation: Experiences with cooperative usability testing. *Journal of Systems and Software*, 83(11), 2019-2030.

Gray, C. M. (2016). What is the nature and intended use of design methods. In *Proceedings of the Design Research Society.*

Guay, S., Rudin, L., & Reynolds, S. (2019). Testing, testing: A usability case study at University of Toronto Scarborough Library. *Library Management, 40*(1), 88-97.

Heiskanen, E., Hyysalo, S., Kotro, T., & Repo, P. (2010). Constructing innovative users and user-inclusive innovation communities. *Technology Analysis & Strategic Management, 22*(4), 495-511.

Hertzum, M. (2016). A usability test is not an interview. *Interactions, 23*(2), 82-84.

Hertzum, M., & Clemmensen, T. (2012). How do usability professionals construe usability? *International Journal of Human-Computer Studies, 70*(1), 26-42.

Hertzum, M., Molich, R., & Jacobsen, N. E. (2014). What you get is what you see: Revisiting the evaluator effect in usability tests. *Behaviour & Information Technology, 33*(2), 144-162.

Holmlid, S. (2009). Participative; co-operative; emancipatory: From participatory design to service design. In *Proceedings of ServDes: DeThinking Service ReThinking Design*.

Hornbæk, K. (2008) Usability evaluation as idea generation. In G. G. Cockton, E. T. Hvannberg, E. Law (Eds.), *Maturing usability: Quality in software, interaction and value* (pp. 267-286). Berlin: Springer.

Hornbæk, K. (2010). Dogmas in the assessment of usability evaluation methods. *Behaviour & Information Technology, 29(1)*, 97-111.

Hornbæk, K., & Frøkjær, E. (2005). Comparing usability problems and redesign proposals as input to practical systems development. In *Proceedings of ACM Conference on Human Factors in Computing Systems* (pp. 391-400).

Hyysalo, S. (2009*). Käyttäjä tuotekehityksessä: Tieto, Tutkimus ja Menetelmät*. Helsinki: Aalto University.

Hyysalo, S. (2015**).** Redrawing the landscape of designing for, with and by users. In *Proceedings of the Scandinavian Conference on Information Systems.*

Iivari, N. (2006). *Discourses on "culture" and "usability work" in software product development* (doctoral dissertation). University of Oulu, Finland.

International Organization for Standardization. (2018). *Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts (ISO 9241-11:2018).* Retrieved from https://www.iso.org/standard/63500.html

Johannessen, G. H. J., & Hornbæk, K. (2014). Must evaluation methods be about usability? Devising and assessing the utility inspection method. *Behaviour & Information Technology, 33*(2), 195-206.

Johnson, M., Mozaffar, H., Campagnolo, G. M., Hyysalo, S., Pollock, N., & Williams, R. (2014). The managed prosumer: Evolving knowledge strategies in the design of information infrastructures. *Information, Communication & Society*, *17*(7), 795-813.

Juurmaa, K., Pitkänen, J., & Riihiaho, S. (2013). Visual walkthrough as a tool for utility assessment in a usability test. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*.

Kankainen, A. (2002). *Thinking model and tools for understanding user experience related to information appliance product concepts* (doctoral dissertation). Helsinki University of Technology, Espoo, Finland.

Kensing, F., & Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work*, *7*(3-4), 167-185.

Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour & information technology*, *22*(1), 1-16.

Kuutti, K. (2011). Out of the shadow of Simon: Artifacts, practices, and history in design research. *In Proceedings of the Doctoral Education in Design Conference.*

Kuutti, K., & Bannon, L. J. (2014). The turn to practice in HCI: Towards a research agenda. In *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 3543-3552).

Mattelmäki, T. (2006). *Design probes* (doctoral dissertation). Aalto University, Finland.

McDonald, S., Monahan, K., & Cockton, G. (2006). Modified contextual design as a field evaluation method. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction* (pp. 437-440).

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica, 22(3)*, 276-282.

Molich, R., & Dumas, J. S. (2008). Comparative usability evaluation (CUE-4). *Behaviour & Information Technology, 27*(3), 263-281.

Nurminen, M. I. (2006). Work informatics—an operationalisation of social informatics. In *Proceedings of the IFIP International Conference on Human Choice and Computers* (pp. 407-416).

Rajanen, M., Iivari, N., & Anttila, K. (2011). Introducing usability activities into open source software development projects—searching for a suitable approach. *Journal of Information Technology Theory and Application, 12*(4), 5-26.

Redish, J. (2007) Expanding usability testing to evaluate complex systems. *Journal of Usability Studies, 2*, 102-111.

Reeves, S. (2019). How UX practitioners produce findings in usability testing. *ACM Transactions on Computer-Human Interaction,* 26(1), 1-38.

Reijonen, P., & Tarkkanen, K. (2015). Artifacts, tools and generalizing usability test results. In *Proceedings of the Scandinavian Conference on Information Systems* (pp. 121-134).

Riihiaho, S. (2009) User testing when test tasks are not appropriate. In *Proceedings of the European Conference on Cognitive Ergonomics* (pp. 228-235).

Riihiaho, S. (2015). *Experiences with usability testing: Effects of thinking aloud and moderator presence* (doctoral dissertation). Aalto University, Finland.

Rosenbaum, S., & Kantner, L. (2007). Field usability testing: Method, not compromise. *In Proceedings of the IEEE International Professional Communication Conference*.

Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing: How to plan, design and conduct effective tests* (2nd ed.). Indianapolis, IN: Wiley.

Sanders, L. (2006). Design research 2006. *Design Research Quarterly, 1*(1), 1-8.

Sanders, E. B. N., & Stappers, P. J. (2008). Co-creation and the new landscapes of design. *Co-design, 4*(1), 5-18.

Savioja, P., & Norros, L. (2008). Systems usability—promoting core-task oriented work practices. In In E. Law, E. T. Hvannberg, & C. Cocton (Eds.), *Maturing usability: Quality in software, interaction and value* (pp. 123-143). London: Springer.

Scope. (2005). In *The Oxford Dictionary of English*. Oxford, UK: Oxford University Press.

Spinuzzi, C. (2002). A Scandinavian challenge, a US response: Methodological assumptions in Scandinavian and US prototyping approaches. *In Proceedings of the 20th Annual International Conference on Computer Documentation* (pp. 208-215).

Spool, J. (2006). Interview-based tasks: Learning from Leonardo DiCaprio. *UIE*. Retrieved from https://articles.uie.com/interview_based_tasks/

Steen, M., Kuijt-Evers, L., & Klok, J. (2007). Early user involvement in research and design projects—a review of methods and practices. In *Proceedings of the 23rd EGOS Colloquium*.

Steen, M. (2008). *The fragility of human-centred design* (doctoral thesis). Delft University of Technology, Delft, Netherlands.

Stewart, J., & Williams, R. (2005). The wrong trousers? Beyond the design fallacy: Social learning and the user. In In D. Howcroft & E. Trauth (Eds.), *Handbook of critical information systems research: Theory and application (pp.* 195-221). Cheltenham, UK: Edward Elgar.

Still, B., & Morris, J. (2010). The blank-page technique: Reinvigorating paper prototyping in usability testing. *IEEE Transactions on Professional Communication, 53*(2), 144-157.

Suchman, L. (1985). *Plans and situated actions. The problem of human-machine communication*. New York, NY: Cambridge University Press.

Sullivan, P. (1989). Beyond a narrow conception of usability testing. *IEEE Transactions on Professional Communication*, *32*(4), 256-264.

Sy, D. (2006). Formative usability investigations for open-ended tasks. In *Proceedings of UPA.*

Sy, D. (2007). Adapting usability investigations for agile user-centered design. *Journal of Usability Studies, 2(3),* 112-132.

Tarkkanen, K., Harkke, V., & Reijonen, P. (2015). Are we testing utility? Analysis of usability problem types. In *Proceedings of the International Conference of Design, User Experience, and Usability* (pp. 269-280).

Tarkkanen K. (2009). Business process modeling for non-uniform work. In J. Filipe & J. Cordeiro (Eds.), *Enterprise information systems (*LNBIP vol. 19, pp. 188-200). Berlin: Springer.

Viitanen, J., & Nieminen, M. (2011). Usability evaluation of digital dictation procedure—an interaction analysis approach. In *Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group* (pp. 133-149).

Vilbergsdottir, S. G., Hvannberg, E. T., & Law, E. L. C. (2014). Assessing the reliability, validity and acceptance of a classification scheme of usability problems (CUP). *Journal of Systems and Software*, *87*, 18-37.

Wixon, D., & Wilson, C. (1997). The usability engineering framework for product design and evaluation. In M. G. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (pp. 653-668). Amsterdam: North Holland.

Åborg, C., Sandblad, B., Gulliksen, J., & Lif, M. (2003). Integrating work environment considerations into usability evaluation methods—the ADA approach. *Interacting with Computers, 15*(3), 453-471.

About the Authors

Kimmo Tarkkanen is a lecturer in the Faculty of Engineering and Business and a member in the Future Interactive Technologies research group at Turku University of Applied Sciences, Finland. He is a passionate usability practitioner, who has performed early usability evaluations for commercial software products and games over ten years and published in many HCI and IS conferences. The topic of this paper is a snapshot of his PhD thesis. His current research interests include also serious games, virtual reality and related user studies.

Ville Harkke is a senior researcher at the Work Informatics User Experience Lab in the University of Turku Business School. His main research interests are the sources and mechanisms of value creation in interaction with complex systems in professional settings.