Are We Testing Utility? Analysis of Usability Problem Types

Kimmo Tarkkanen, Ville Harkke, and Pekka Reijonen

Information Systems Science, University of Turku, Turku, Finland
{kimmo.tarkkanen, ville.harkke}@utu.fi

Abstract. Usability problems and related redesign recommendations are the main outcome of usability tests although both are questioned in terms of impact in the design process. Problem classifications aim to provide better feedback for designers by improving usability problem identification, analysis and reporting. However, within the classifications, quite little is discussed about the types and the contents of usability problems as well as the types of required design efforts. We address this problem by scrutinizing the findings of three empirical usability tests conducted in software development projects. As a result, 173 problems were classified into 11 categories. Specific focus was placed on the distinction between the utility and usability types of problems, in order to define the correct development phase and method to fix the problem. The number of utility problems varied from 51% to 74%, which shows that early usability testing with a think-aloud protocol and an open task structure measure both utility and usability equally well.

Keywords: Usability problem, utility problem, problem classification, usability testing

1 Introduction

Usability testing is a popular method to evaluate early designs during product development. Usability problems and related redesign recommendations are the main outcome of usability tests. Much of the critic towards usability testing pinpoints the role of usability problems both in method development and in method deployment in the design process [1,2]. Particularly, problem identification, extraction, documentation, and value for design have been challenged. Both industry and research cases report that usability problems are rarely fixed and redesign proposals are not very influential in the short term development [3,4]. Therefore, research efforts have been put to study usability practitioners' analysis practices [5], evaluation results downstream utility [6], use of usability method ingredients [7] as well as further developing problem report formats [8] and problem classifications [9]. In their CUP (Classification of Usability Problems) Scheme Hvannberg and Law identify several attributes of problems, the most important of which are, for the purpose of giving better feedback to designers and measuring the design influence of usability testing, 'failure qualifier' and 'expected phase' [10]. The former problem attribute helps designers to see the real problem whereas the latter makes them think about how to fix the problem [11]. However, quite little is known about these attributes i.e. the types and the contents of usability problems as well as types of required design efforts. We address this problem by scrutinizing the findings of three industrial usability tests. Using the grounded theory methodology, we form a categorization of problem types with the data derived from three empirical usability tests that were performed with open test tasks. All the tested systems were of professional nature and were to be used as tools in the actual work of the users. The testing was a part of systems development effort and conducted on prototypes simulating new systems.

The categories of problem types were further analyzed to reveal what the problem types are measurements of. Specific focus was placed on the distinction between utility and usability issues, the former being issues that are related to the functions/offerings of the systems i.e. whether it is possible to do with the system what the user expects to get done, and the latter being a measure of how effective, efficient and pleasant it is to use the system in doing the needed things. In their study on usability testing results, Norgaard and Hornbaek [12] found that utility problems are much less frequently explored than usability problems. By separating usability and utility types of problems, we are able to help designers to address the correct development phase and method to fix the problem.

2 Utility vs Usability

Nielsen [13] made in his early definition of usability a clear distinction between usability and utility which are the constituents of usefulness, i.e. "whether the system can be used to achieve some desired goal" (p.24) [13]. According to this definition, utility concerns the functionality of the system and usability is the question of how well users can use this functionality. In other words, usability is concerning with 'how the system is operated' and utility with 'what the system can do'. The distinction between utility and usability is not always that straight forward: The examples of the benefits of usability engineering given by Nielsen (see [13] p. 2) point more to utility than usability. Usability has no meaning without appropriate functionality and utility is not realized without good system usability [14]. The distinction can even be seen as superficial and a mere result of different disciplines focusing on different aspects. Already in 1988, Whiteside et al. stated that: "usability and functionality are linked inseparably in design and implementation" [15]. This was supported by Grudin [17]: "It is notoriously difficult to separate the function of interactive software from its form, to draw a line between software functionality and its human-computer interface." In a broader view, as in the ISO definition of usability [17], usability takes into account the context of use and usefulness aspects [18].

Despite the fact that it is difficult to distinguish between utility and usability in a real-world implementation, it is vital that both aspects of usefulness are addressed in evaluations. Furthermore, the two aspects lead to different types of problems and, ultimately, different types of possible solutions to them. As Mahmood et al. [19] describe: "It seems that end-users primarily adopt an application based on perceived benefits, and secondly on how easy or hard it is to achieve those benefits (...) no amount of use can compensate for lack of needed functionality." Johannesen and Hornbaek [20] again point out that utility is about building the right system.

For usability testing findings, this distinction would translate to that a usability problem is one that makes it difficult, cumbersome or unpleasant to achieve one's goal, but a utility problem makes it impossible. Utility problems may occur regardless of the users' knowledge of the system usage. The origin of impossibility is then in the fit between the system properties and the test users' way of work in the specific context. Although the definition of a usability problem by [8] includes such "an aspect of the system and/or a demand on the user" that makes it "impossible for the user to achieve their goals in typical usage situations", the interpretation is that usability can prevent task completion, but do not necessarily imply that the system is useless in the work tasks, if the usability problems are corrected.

3 Usability Problem Classifications

There are numerous of ways to classify usability problems. The roots of the problem classifications lay in the software defect tracking models at the beginning of the 90's [21]. For example, the Orthogonal Defect Classification (ODC) classified software defects in order to give useful feedback to developers and managers on the progress of the development project [22] and to steer development in reactive or proactive manner [23]. ODC concentrates on the problem causes on the system's side i.e. it describes what is wrong with the design and what should be fixed by the system designers. For that purpose, the defect qualifiers (the values for different types of defects) 1) missing, 2) incorrect (and later also irrelevant) were introduced [22].

One of the first holistic usability problem classifications, the User Action Framework (UAF) by [24] is "a classification scheme for usability problems based upon the type of problem in terms of its cause within the interaction cycle" ([24],p.112). UAF organizes human activity into several phases finally locating the found problem into a node in a hierarchical tree, for example under "font size and contrast" ([24] p. 127), which is then considered as the root cause for the classified problem. Thus, UAF follows a classification taxonomy, namely the UPT (the usability problem taxonomy by [25]), which defines the problem in the artifact component or in the task component at very detailed level (28 categories). In design recommendations, UAF relies on the cumulative knowledge about the problem types collected into the database rather than on describing how something is a problem in the first place, like ODC does. In other words, if we want to know how the problem appears in the system or what kind of problem it is to the developer, the classifiers that describe that some named system element is "missing" or "irrelevant" would probably be in many cases more informative and practical for designers than a statement: "The usability problem is a high-level planning issue involving the user's model of the system in order to understand the overall concept" (see [24] p.132). Nevertheless, the positive effects of such taxonomy for learning and steering purposes are undeniable within a longer time frame.

The most recently refined usability problem classification is the Classification of Usability Problems (CUP) scheme by [9,10]. It has its basis in ODC. In the CUP terminology, the attribute explaining "how the user/expert experienced" a usability problem is called a failure qualifier (defect qualifier in ODC). In the most recent version of CUP [10] a usability problem is something that is being 1) missing, 2) incongruent, 3) irrelevant, 4) wrong, 5) better way or 6) overlooked (i.e. the possible values of the qualifier attribute). As these failure qualifiers were deemed useful by designers to understand problems [11], our following analysis aims first to classify usability problems by qualifiers arising from empirical data from prototype tests, and second to separate usability and utility types of problems in order to provide better feedback for designers.

4 Research Method

4.1 Data Collection

For this study, we collected and analyzed data from three different usability tests. The tests were conducted during 2012-2013 by the authors for the responsible for the design of prototypes. Our tests were the first usability tests with future users for each of the prototypes. The prototypes were designed for professionals in the health care domain and the test participants represented the current users and customers of the company's products. Two systems were tested as paper prototypes (cases 2 and 3) one involving over 170 and the other 38 printed screens on A4 sized papers. The prototype in case 1 was implemented on a tablet computer, which allowed more feasible and effortless navigation than the paper versions in other cases. All tests applied a think-aloud method and had at least two administrators present. In the case 1, also two of the designers followed the sessions. The number of participants in cases varied from four to six professionals. Video and audio were recorded in all test sessions.

Despite the differences in the purposes and use contexts of the prototypes, all the test tasks were designed in similar manner avoiding the too detailed presumptions of the work tasks of the users: High-level and open-ended tasks were given to users in each session (see [26]). For example, in case 1 the test task was (translated into English): "You have just arrived at your workplace and you begin to prepare your work shift. This [prototype name] is a new application that you can use during your shift. You have already logged in." The open task approach was supplemented with predefined lower level tasks, for example in situations where designers had some open design questions and users did not work on that question during the open task. Otherwise, the tests followed common problem identification strategies of think aloud testing. The origins of problems lay in users' verbal and non-verbal behavior observed as well as in the evaluators' interpretations of these actions, the combinations of similar problems and system-initiated malfunctions. The problems are thus based on, for example, users negative feelings and negative expressions about the aspects of the system and their conscious or unconscious lack of understanding of the system features and objectives. Problems based on non-verbal indications were related, for example to time (e.g. slowness, delay, number of tries), errors (wrong path, randomness, slips) and task completion (giving up, wrong result, impossibilities). The found problems were documented to the final report delivered to the responsible system designers in each case. The findings and reported usability problems in the cases are based on our participatory involvement in the design and evaluation of the usability of the system, which follows loosely the tradition of action research (see [27]).

4.2 Data Analysis

The individual evaluation results, the reported usability problems of the cases were approached with a grounded theory methodology [28] retrospectively for this study. The procedure started with one researcher who reviewed usability problems and gave each problem a code, either existing from previously reviewed problems or creating a new one. Codes were abstractions of real findings and newly invented during the research process. We were not following any pre-existing problem classifications or values of failure qualifiers, in order to keep the origins of the analysis purely in our empirical data. However, we wanted to increase our understanding about the underlying characteristics of the problems in terms of how something is a problem. Therefore, the coding was done in relation to the system and users' work. The fundamental question followed in the coding was "what is wrong with the system from the viewpoint of users' work". The question is relevant particularly in the early usability evaluations, because the assumption of lean user experience studies is that the initial designs will be wrong, and what is wrong, needs to be found as soon as possible [29].

As the categories are built and formulated on the basis of the question above, the problem categories represent design faults from the users' and their work point of view, and are thus values of the failure/defect qualifier attribute. For example, users' were reported as "confusing the meaning of the symbols of isolated and inert patients" as well as "misinterpreting the meaning of the numeric value related to laboratory results", which were then encoded under the same category of (the system feature is) "misinterpreted". The categories are exclusive i.e. the same problem was located in only one category. During the coding, a criterion for each category was iteratively refined. The related design decisions (e.g. misinterpreted features need to be explicated more clearly in the next version of the design) were not considered but were attached after the whole coding process. After coding the first case, another researcher performed the same analysis and coding with the existing scheme, yet including, excluding or altering categories, criteria or both. After both researchers had coded the case, the codes were combined and each problem and coding category was discussed to achieve mutual understanding about the codes and the placement of problems in categories. The same procedure was applied to each of the three cases. The final number and description of categories and problems in categories are discussed in the next chapter.

5 Results

In our analysis, 11 different problem types were identified (i.e. failure qualifiers in CUP terms) from the total of 173 reported problems analyzed. The problem types are described below in the numbered list (1-11). Our categorization is not intended to function as a tool for usability evaluations as such, but is merely a tool for dissecting the information we have collected. The purpose is to help us define the attributes of human-system interaction that are classifiable as problems in order to highlight areas of high relevance. This categorization does not require a content/functionality dichotomy but is applied to findings regarding both the functions of the tested system prototypes and the information form and content provided by the prototypes. The categories below contain the most obvious and typical design improvement suggestions and a distinction where in the typical development cycle of the user-centered design process the problem should be addressed (see [30] for phases: *understand* context of use, *specify* requirements, *produce* design solution and *evaluate* solution [not applied]).

1. Missing information or functionality:

- An element of the system that is necessary for the users' work is not available at all. The task/work cannot be performed with the presented system. Information or functionality has not been implemented, designed or planned to be designed, yet after the test identified as a user requirement and critical for performing the work.
- Design decisions: Add new feature
- Development phase: Understand

2. Misinterpreted information or functionality:

- The terminology or symbols/functions are not correctly understood by the user. She thinks about a different meaning for the symbol, feature, function or information from the designed purpose.
- Design decisions: Clarify feature
- Development phase: Specify

3. Positive information or functionality:

- The feature or information is found good, pleasant or effective.
- Design decisions: Implement feature
- Development phase: Produce

4. Inadequate information or functionality:

- A required element of the system is present but the implementation is not sufficient for the task at hand. Information or functionality has been designed and implemented into the system, but it lacks a proper fit with the work and practices of users preventing or significantly hindering the performance.
- Design decisions: Refine feature
- Development phase: Specify

5. Unexplored design issue:

- The function or information provided by the system could be used to increase the effectiveness of work but the exact changes in requirements are unclear. A need or possibility for positive change in the current work practice is identified, which can lead to new design issues. Not necessarily critical for work performance (at this stage), yet could drastically improve UX or result in other value for the users.
- Design decisions: Consider designing a feature (i.e. invent feature)
- Development phase: Understand

6. Misplaced information or functionality:

- The needed element is available and adequate but in a cumbersome format or requires unnecessary effort to find and use. The feature or information is implemented somewhere or somehow, but not available at a required place and point in time. This covers misplacing or replicating information under certain features and representing information in unfamiliar terms and inappropriate forms.
- Design decisions: Duplicate feature (or delete and add)
- Development phase: Produce

7. Unnecessary information or functionality:

- Users do not use, notice, behaviorally or verbally ignore a function or a piece of information that has been implemented.
- Design decisions: Remove feature
- Development phase: Produce

8. Technical deficiencies or carelessness in implementation:

- The design is implemented with errors/bugs. These are mostly due to the technical development phase of the system i.e. due to unpolished prototypes.
- Design decisions: Repair feature
- Development phase: Produce

9. Problematic change of work practice:

- Using the system as planned would change the work patterns in such a way that causes problems elsewhere. This may not realize only benefits but also major drawbacks. Users point to the problematic effects and uncertain benefits of a feature. A feature may cause a change in work that is experienced as problematic and questionable.
- Design decisions: Re-consider feature
- Development phase: Understand

10. Preferenced information or functionality:

- A way of doing something in the system is preferred to an alternative way.
- Design decisions: Implement feature and create a design pattern
- Development phase: Produce

11. Misaligned information or functionality:

- The feature or the information would require a change in the work practice to be useful. The way in which information or functionality is meant to be used differs from the existing or traditional use. This may or may not be intentional, depending on whether the change in work practice is one of the purposes of the new system implementation. Features that are implemented for instance based on legislation generate this type of problems.
- Design decisions: [Out of control of the system design process]
- Development phase: Understand

The number of problems in each category is presented in Table 1. The categories Missing (no. 1), Inadequate (4), Unexplored (5), Unnecessary (7), Problematic (9) and Misaligned (11) are primarily utility problems i.e. these problems can render doing the job simply impossible (categories 1 and 4), be prone to cause unfavorable (9) and uncontrollable consequences (11), be useless (7), or worth to explore for more benefits (5). The rest of the categories (no. 2, 6, 8 and 10) are more dependent on the interface design and as such must be seen as usability problems. Problems in these categories may prevent task completion, but are not in contradiction to the goals and tasks of the users. For example, system features that are misinterpreted by the users can be redesigned without altering the purpose and goal of the feature.

Table 1. Numbers of usability and utility problems and problems in each category.

Category name:	1. Missing	2. Misinterpreted	3. Positive	4. Inadequate	5. Unexplored	6. Misplaced	7. Unnecessary	8. Technical	9. Problematic	10. Preferenced	11. Misaligned	Total	Utility problems	Usability problems
Case 1	20	4	0	7	8	5	3	6	4	0	0	57	42	15
Case 2	20	9	8	7	7	6	3	1	2	2	0	65	39	18
Case 3	12	15	10	2	1	3	3	1	0	1	3	51	21	20
Total	52	28	18	16	16	14	9	8	6	3	3	173	102	53

Problems in the Positive category (no. 3) are not counted either as utility or usability problems. Findings in the positive category implicate that users have liked system features, which do not need to be altered in the design. Moreover, positive usability findings would form a classification of their own (like problems do), if analyzed in more detail, which was not the purpose of this analysis. Definitely, positive findings would cover both usability and utility issues. Thus, the total number of problems included is 155. Table 1 shows that 34% (53) of the reported problems were usability observations, while 66 % (102) of reported problems concerned utility issues. Percentages of utility problems in individual cases vary from 51% (Case 3) to 74% (Case 1).

6 Discussion

Compared with other studies, our analysis required five more categories than the ODC-based CUP scheme [10] but 17 less than UPT-based [25] classifications. The categories of *Missing* and *Unnecessary* are commonly found in other classifications (e.g. overlooked, extraneous, irrelevant). *Misplaced* has similarities with incorrect and *Misinterpreted* with incongruent in the CUP. *Inadequate* has no direct correspondence in CUP. In our interpretation the *Unexplored*, *Misaligned* and *Problematic* categories are distinct, whereas the CUP scheme assigns only one category, better way, for these types of problems. However, we find this distinction valuable, because the subsequent design decisions are also very different.

If the results are compared with a specific evaluation method, which is designed to support substantial re-design and improvement of system utility [31], we find that our categories *Missing*, *Unnecessary* and *Inadequate* cover the possible combinations of the misfits presented by the CASSM method (see example 1 in [31]. From the study by Norgaard and Hornbaek [12] we can find at least implicit correlation in the categories *Missing*, *Unnecessary*, and *Positive* as well as in the work-based categories *Problematic*, *Misaligned* and *Unexplored*. Moreover, their exploration of utility issues is also in line with our usability and utility distinction of the categories above. For them, utility problems included the tasks the system did not support, the notions of unrealistic test tasks as well as users' actual and desired usage flows dissimilar to flows implemented into the system [12]. For example, a statement of user "I would use Phonebook [which is not implemented]" was then identified as a utility problem. However, Norgaard and Hornbaek [12] found that utility problems are much less frequently explored than usability problems in think aloud usability test sessions. Utility issues were discussed in 10 out of 14 think-aloud sessions analyzed, yet in 13 sessions, usability was favored over problems relating to the utility of the system. In contrast, our results show that utility problems can and indeed will be found in usability testing.

7 Conclusions

We set out to analyze the findings of our usability tests to identify different types of usability problems. In our data-based analysis, we were able to distinguish a set of categories, which differ from each other in terms of how they manifest themselves and how the problems can be addressed in further software development. The number of findings in each category and division of those into usability and utility issues indicate that formative usability tests conducted with open task approach measure equally well utility, "what the product should do", and usability, "how the product is operated". Despite our similar interpretation of distinction between utility and usability, our results are inconsistent with the findings by [12]. Currently, our best guess for the inconsistency is that open test tasks produce more utility related findings than predefined test tasks, which were presumably applied in their study. In addition, the early stages of the development in our tests may have brought out more utility problems.

The utility related problem types are very context-dependent and rely heavily on the procedure of testing with real users in actual usage context. This is positive in a sense that the fit of the proposed system to the actual usage patterns and needs can be verified within the tested context. However, the generalizability of the results to different usage situations and contexts is compromised. One interesting aspect of our findings was the relatively high amount of problems and categories that demand further exploration of the work patterns and contexts of the users (*Missing*, *Unexplored*, *Problematic* and *Misaligned* cover 45% of the problems). Especially interesting is the type of finding where an attribute of the tested system is potentially beneficial but works only if the work patterns and organization of work are radically changed (*Misaligned*). This leads to situations where the design team might not have the power or even right to make decisions about whether the system or organization of work should be changed. This highlights the necessity of software developers' cooperation with the users on different levels in order to realize the full potential of new information systems design.

Our findings add to the recently started development of utility evaluation methods as well as theoretical discussion of researching utility distinct from usability (see [20]). In addition, problem classifications and their potential feedback for design are again being upgraded [10], [21], [32]. One of the benefits of this research for HCI practitioners lies in the clarification of which issues can be improved by redesigning the systems and which would be better addressed by changing the work patterns or the social/organizational constructs where the systems are to be used. The distinction of usability and utility problems will help making such design decisions.

The prototypes tested were all employer-provided professional systems, i.e. their use is not voluntary for the user and the higher level goals of usage are set by the employing organizations and not the users themselves. Furthermore, the systems and their usage can be seen as complex in a sense that the order and desired outcome of system tasks can vary (see [33]). In this type of environment the majority of found problems are utility-related, suggesting that the development and requirements elicitation methods used previously in the design process do need support from this form of testing (45% of our findings denote lack of knowledge of the context of use). On the other hand, the context may also be a threat to the validity of the developed categories and the overall study as the complexity of the use context may lead to an exceptional number of utility related problems. Furthermore, the evaluator effect is present in a usability study [34], which implies that the following design recommendations have no scientifically constructible relation to observed problems. This is a validity and reliability problem of not only this research but all practical usability evaluations. The validity and reliability of the categories should be tested with more evaluators, different empirical data and formal methods. Moreover, in the future, we consider it important to study how the problem categories are valued in design, what are the practical benefits to, and how feasibly the categories can be exploited in the design process.

References

1. Wixon, D.: Evaluating usability methods: why the current literature fails the practitioner. Interactions 10 (4), 28-34 (2003)
2. Hornbæk, K.: Dogmas in the assessment of usability evaluation methods. Behaviour & Information Technology, 29(1), 97-111 (2010)
3. Molich, R., Ede, M.R., Kaasgaard, K., Karyukin, B.: Comparative usability evaluation. Behaviour & Information Technology, 23(1), 65-74 (2004)
4. Hornbæk, K., Frøkjær, E.: Comparing usability problems and redesign proposals as input to practical systems development. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 391-400. ACM (2005)
5. Følstad, A., Law, E.L.-C., Hornbæk, K.: Analysis in usability evaluations: an exploratory study. in Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 647-650. ACM (2010)
6. Law, E.L.-C.: Evaluating the downstream utility of user tests and examining the developer effect: A case study. International Journal of Human-Computer Interaction 21(2), 147-172 (2006)
7. Woolrych, A., Hornbæk, K., Frøkjær, E., Cockton, G.: Ingredients and meals rather than recipes: A proposal for research that does not treat usability evaluation methods as indivisible wholes. International Journal of Human-Computer Interaction, 27(10), 940-970 (2011)
8. Lavery, D., Cockton, G., Atkinson, M.P.: Comparison of evaluation methods using structured usability problem reports. Behaviour & Information Technology, 16, 246-266 (1997)
9. Hvannberg, E.T., Law, E.L.-C.: Classification of Usability Problems (CUP) Scheme. In Proceedings of INTERACT 2003, pp. 655-662. ACM Press (2003)
10. Vilbergsdottir, S.G., Hvannberg, E.T., Law, E.L.-C., Assessing the reliability, validity and acceptance of a classification scheme of usability problems (CUP). Journal of Systems and Software 87, 18-37 (2014)
11. Vilbergsdóttir, S.G., Hvannberg, E.T., Law, E.L.-C.: Classification of usability problems (CUP) scheme: augmentation and exploitation. In Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles., pp. 281-290. ACM (2006)
12. Nørgaard, M., Hornbæk, K.: What do usability evaluators do in practice? an explorative study of think-aloud testing. In Proceedings of the 6th conference on Designing Interactive systems, pp. 209-218, ACM (2006)
13. Nielsen, J.: Usability engineering. Elsevier (1994)
14. Goodwin N.: Functionality and usability. Comm. of the ACM 30, ACM (1987)
15. Whiteside, J., Bennett, J., Holtzblatt, K.: Usability engineering: Our experience and evolution. In M. Helander, M. (eds.) Handbook of Human–Computer Interaction, North Holland, Amsterdam (1988)
16. Grudin, J.: Utility and usability: research issues and development contexts. Interacting with computers 4(2), 209-217 (1992)
17. ISO, 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)–Part II Guidance on Usability, (1998)
18. Bevan, N.: Usability is quality of use. Advances in Human Factors/Ergonomics 20, 349-354 (1995)
19. Mahmood, M.A., Burn, J.M., Gemoets, L.A., Jacquez, C.: Variables affecting information technology end-user satisfaction: a meta-analysis of the empirical literature. International Journal of Human-Computer Studies 52(4), 751-771 (2000)

20. Johannessen, G.H.J., Hornbæk, K.: Must evaluation methods be about usability? Devising and assessing the utility inspection method. Behaviour & Information Technology 33(2), 195-206 (2014)
21. Ham, D.-H.: A model-based framework for classifying and diagnosing usability problems. Cognition, Technology & Work 16(3), 373-388 (2014)
22. Chillarege, R., Bhandari, I.S., Chaar, J.K., Halliday, M.J., Moebus, D.S., Ray, B.K., Wong, M.-Y.: Orthogonal defect classification -a concept for in-process measurements. IEEE Transactions on Software Engineering 18(11), 943-956 (1992)
23. Grady, R.B.: Software failure analysis for high-return process improvement decisions. Hewlett Packard Journal 47, 15-24 (1996)
24. Andre, T.S., Rex Hartson, H., Belz, S.M., McCreary, F.A.: The user action framework: a reliable foundation for usability engineering support tools. International Journal of Human-Computer Studies 54(1), 107-136 (2001)
25. Keenan, S.L., Hartson, H.R., Kafura, D.G., Schulman, R.S.: The usability problem taxonomy: A framework for classification and analysis. Empirical Software Engineering 4(1), 71-104 (1999)
26. Tarkkanen, K., Reijonen, P., Tétard, F., Harkke, V.: Back to User-Centered Usability Testing. In Holzinger et al. (eds.) Human Factors in Computing and Informatics, LNCS, vol. 7946, pp. 91-106. Springer Berlin Heidelberg (2013)
27. Baskerville, R.L.: Investigating information systems with action research. Comm. of the AIS 2(3es) p.4 (1999)
28. Strauss, A., Corbin, J.M.: Basics of qualitative research: Grounded theory procedures and techniques. Sage Publications (1990)
29. Gothelf, J.: Lean UX: Applying lean principles to improve user experience. O'Reilly Media (2013)
30. ISO, 9241-210: Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems (2010)
31. Blandford, A., Green, T.R., Furniss, D., Makri, S.: Evaluating system utility and conceptual fit using CASSM. International Journal of Human-Computer Studies 66(6), 393-409 (2008)
32. Geng, R., Chen, M., Tian, J.: In-process Usability Problem Classification, Analysis and Improvement. In 14th International Conference on Quality Software 2014, pp. 240-245 IEEE (2014)
33. Campbell, D.J.: Task complexity: A review and analysis. Academy of management review 13(1), 40-52 (1988)
34. Hertzum, M., Molich, R., Jacobsen, N.E.: What you get is what you see: revisiting the evaluator effect in usability tests. Behaviour & Information Technology *33*(2), 144-162 (2014)