Reporting Usability Metrics Experiences

Jeff Winter
School of Engineering
Blekinge Institute of Technology
Soft Center,
SE 372 25 Ronneby, Sweden
jeff.winter@bth.se

Kari Rönkkö
School of Engineering
Blekinge Institute of Technology
Soft Center,
SE 372 25 Ronneby, Sweden
kari.ronkko@bth.se

Mats Hellman
Product Planning User Experience
UIQ Technology AB
Soft Center VIII
SE 372 25 Ronneby, Sweden
mats@rightfromstart.se

Abstract

It is often claimed that software development is negatively affected by infrequent, incomplete and inconsistent measurements; improving with the help of metrics is an obvious solution. Software testing provides opportunities for measurement that give organizations insight in to processes. Usability testing is part of the testing area, although it is not a commonly addressed area within software engineering, perhaps because of a split between qualitative and quantitative paradigms. We compare a usability testing framework called UTUM with principles for Software Process Improvement, and find areas of close agreement as well as areas where our work illuminates new characteristics. UTUM is found to be a useful vehicle for improvement in software engineering, dealing as it does with both product and process. Our work emphasises the importance of the neglected area of usability testing. Our experience also illustrates how the metrics have been tailored to act as a boundary object between different disciplines.

1. Introduction

In this paper we discuss our experiences of improving software development by applying the language of metrics. We discuss how we tailored our metrics, based on work heavily influenced by a qualitative tradition, to fit an organization influenced by a quantitative software engineering tradition. Software engineers often believe that software development is negatively affected by infrequent, incomplete and inconsistent measurements; improving with the help of metrics is an approach that is taken for granted. Software testing permits measurements that give organizations better insight in the processes monitored. Usability testing is one part of the testing area, although it is not a commonly addressed area within software engineering. This may because usability testing is an interdisciplinary subject, often performed by people with a social sciences background that emphasises qualitative aspects. These people are thus professionals from other paradigms and backgrounds, trained for the usability test situation, rather than software development process aspects.

Iversen and Kautz [7] have studied and summarized recommendations and advice regarding Software Process Improvement (SPI) metrics implementations, and suggest that whilst guidelines and frameworks are useful, organizations must tailor methods to fit their particular situation. This is particularly true since much of the advice given disregards the fact that many companies that do not employ large numbers of software developers, and that organizational cultures differ between different countries. Advice aimed at e.g. North American companies may not fit in with organizational cultures in other countries [7:288-289].

Their study dealt with SPI, and involved collecting data on the work performed by developers on an individual level, whilst our work has been the development and implementation of usability testing. However, we have found many interesting parallels to their work and results and the work performed and findings made whilst developing and implementing UIQ Technology Usability Metrics (UTUM). UTUM

CHASE'09, May 17, 2009, Vancouver, Canada
978-1-4244-3712-2/09/$25.00 © 2009 IEEE

is a usability test framework developed in an action research fashion, in cooperation between in particular interaction designers from UIQ Technology AB and researchers (the present authors) from Blekinge Institute of Technology. In this paper we map our experiences of applying usability metrics against Iversen and Kautz's [7] analysis of SPI metrics. The comparison is made here to help the reader assess the applicability of the lessons intended for companies attempting to implement software metrics programs. The experiences presented are from a Scandinavian based company in the telecommunications branch.

First, we introduce the telecommunications branch, followed by a short presentation of the research partners. Thereafter we present the UTUM. After this we compare the areas and principles listed by Iversen and Kautz with the conclusions we have drawn from our research and implementation of usability metrics. Finally, we discuss some issues we confronted and conclude our experiences. We believe this paper is of value for organizations in search for a starting point to tailor their usability metrics.

2. The telecom area

Telecommunications, where we have cooperated with a company that developed and licensed a user interface platform for mobile phones, is a market driven software development area. In market driven areas, there are potential groups of people who fit an imagined profile of intended users rather than a distinct set of users. Requirements elicitation is mainly managed through marketing, technical support, trade publication reviewers and user groups. Recent study in this area has revealed that the constant flow of requirements caused by the variety of stakeholders with different demands on the product is an issue. In market driven companies requirements are often invented based on strategic business objectives, domain knowledge and product visions. (See [8] for overview)

It is thus a great challenge to develop a usability test framework for mass market products, where economic benefits are gained through approaching the broadest possible category with one single product. Whilst specific end-user groups must be targeted [5], there is an unwillingness to exclude other potential end-user categories [11]. In developing software designed for single organizations, end-user participation, empowerment and the development of routines to be supported by technology are easier to identify, scope, and handle. In a mass market, it is harder to identify and portray the complexity and representativeness of end-users. Social and political aspects that influence the usefulness of products might even be filtered out by the evaluation techniques used in mass markets (see [4] and [11]). The design of products is influenced by competitors launching new products and features, and by technical magazines publishing reviews and comparisons. Timing aspects give competitive advantages and influence design decisions. Telecommunications focuses on providing the market with new and improved technology rather than fulfilling end-user needs. All together these branch characteristics have so far challenged and bounded the design space for requirements elicitation and validation, and user testing. Since Apple's launching of iPhone, claimed to be the first 'real' user experience product on the market, the above branch characteristics may change.

3. Research cooperation

UIQ Technology AB, founded in 1999 and closed in January 2009, was an international company that early in 2008 had more than 320 employees in Sweden, and around 400 employees in total. The company developed and licensed a user interface platform for mobile phones using Symbian OS. The product, UIQ's user-interface platform, enabled mobile phone manufacturers to create different kinds of phones for different market segments, all based on one codeline. Through its independent position (not directly tied to a specific phone manufacturer) the company promoted the introduction of new advanced mobile phones onto the market. Its main assets were technical architecture, a unique independent product, and skilled, experienced staff. More than 20 UIQ-based phones were released.

The research group that has participated in the development of UTUM is U-ODD, Use-Oriented Design and Development [15], within the School of Engineering at Blekinge Institute of Technology (BTH), which is part of the research environment BESQ [1]. U-ODD approaches software engineering via use-orientation, influenced by the application of a social science qualitative research methodology and the end-user's perspective. The human role in software development is an area needing further research in software engineering. The task of understanding human behaviour is complex and necessitates the use of qualitative methods, since quantitative and statistical methods have been found to be insufficient [13]. The strength of qualitative methodologies is in exploring and illuminating everyday practices of software engineers, through observing, interpreting and

implementing the methods and processes of the practitioners.

The process of cooperation is Action research according to the Cooperative Method Development (CMD) methodology, see [3] for details. It combines qualitative social science fieldwork, with problem-oriented method, technique and process improvement. The starting point for CMD is existing practice in industrial settings. It is motivated by an interest in use-oriented design and development of software, but is not specific for these methods, tools and processes. The phases in CMD are:

Phase 1 – Understanding Practice. The research begins with empirical investigations, to understand and explain existing practices and designs from a practitioner's point of view, based on their historical and situational context, to identify aspects that are problematic from the practitioner's point of view.

Phase 2 – Deliberate Improvements. Results from phase 1 are used in a cooperative fashion by researchers and the practitioners involved, as an input for the design of possible improvements. The outcome of this phase is the deliberation of measures that address some of the identified problems and that are expected to improve the situation at hand.

Phase 3 – Implement and Observe Improvements. Improvements are implemented. Researchers follow these improvements as participant observers. Results are evaluated together with the practitioners. In this evaluation, the concrete results are summarized for the companies involved, and build a base for the researchers to evaluate the proposed improvements.

4. The UTUM test

UIQ Technology Usability Metrics (UTUM) is an industrial application developed and evolved through long term CMD cooperation between BTH/U-ODD and UIQ Technology. It is a test framework grounded in usability theory and guidelines, and in industrial software engineering practice and experience. It bridges a gap between the Software Engineering and the HCI communities. UTUM is "a method to generate a scale of measurement of the usability of our products on a general level, as well as on a functional level". Due to space limitations, we cannot give a closer presentation of UTUM here. It is presented in greater detail in [17]. A video demonstration of the whole test process (ca. 6 minutes) can be found on YouTube [20].

Two distinctive characteristics of UTUM deal with the relationship to users, and how user input is received and perceived, whilst two deal with software development practice, concerning organization and method development in the company.

The first characteristic is the approach to getting user input and understanding users. Here we apply an ethnographic mindset [12]. To understand the user's perspective, rather than simply observing use, the test expert interacts and works with the users, to gain insight into how they experience being a mobile phone user. The users that help with the testing are referred to as testers, because they are doing the testing. The representative of the development company is referred to as a test leader, or test expert, emphasizing the qualified role that this person assumes. The second characteristic deals with utilizing the phone users' inventiveness, and entails letting users participate in the design process. The participatory design tradition respects the expertise and skills of the users, and this, combined with the inventiveness observed when users use their phones, means that users provide important input for system development. The test expert is the advocate and representative of the user perspective. User participation gives designers, with the test expert as an intermediary between them and the users, good input throughout the development process.

The third characteristic is continuous and direct use of user input in design and decision processes. The high tempo of software development for mobile phones makes it difficult to channel meaningful testing results to the right recipient at the right time in the design process. Integrating the role of the test expert into the daily design process eases this problem. The results of testing can be directed to the most critical issues, and the continual process of testing and informal relaying of testing results to designers leads to a short time span between discovering a problem and implementing a solution. The fourth characteristic concerns presenting results in a clear and concise fashion, whilst keeping a focus on understanding the user perspective. The results of qualitative research are summarized by quantitative methods, giving decision makers results in the type of presentations they are used to. Statistical results do not supplant the qualitative methods that are based on PD and ethnography, but they capture in numbers the users' attitudes towards the product they are testing. It is in relation to the third and fourth characteristic we have the most obvious relation to organizational metrics.

Although these four characteristics are highlighted separately, they are not simply used side by side, but are part of one method that works as a whole. Even though many results are communicated informally, an advantage of a formal testing method is traceability. It is possible to see if improvements are made, and to trace the improvements to specific tests, and thereby make visible the role of the user testing in the design

and quality assurance process.

UTUM is a cost effective tool for guiding design decisions, and involves users in the testing process, since providing a good user experience relies on getting and using good user input. UTUM measures usability empirically, on the basis of metrics for satisfaction, efficiency and effectiveness, and a test leader's observations. It gives a demonstration of quality in use, from the customer and end-user point of view. So far, it is mainly used for measuring usability and quality in use, but the ambition is to adapt the tool to capturing user experience to a much larger degree than it does today. Due to the closure of UIQ Technology AB the challenge of UX is continued within a new research project within the telecommunication branch called WeBIS, with the purpose of producing industrially viable methods for user generated innovation in ICT-services [16].

5. SPI vs. Usability testing metrics

In this section, we compare the findings of Iversen and Kautz [7] with the situation that we found in our study. We point out in which way our findings agree with their principles, but also where we shed light on new aspects of the principles. Iversen and Kautz identified five areas and principles to support the tailoring of metrics to organizations: **Knowledge** (Use improvement knowledge, Use organizational knowledge), **Organisation** (Establish a project, Establish incentive structures), **Design** (Start by determining goals, Start simple), **Communication** (Publish objectives and collected data widely, Facilitate debate), **Usage** (Use the data). In the following, we demonstrate how the areas and principles listed compare with our research and our implementation of usability metrics.

Knowledge: when implementing metrics, it is important to draw on different kinds of knowledge, including knowledge about software improvement and organizational change, but also knowledge about the actual organization where the metrics program is to be implemented. To successfully implement metrics programs, members of the organization should be knowledgeable about the art of software metrics, software improvement, software engineering, and the process of organizational change.

Principle 1 concerns the use of improvement knowledge. In our case, to understand and approach the industrial practice we have applied the three action research phases from CMD [3]. With help of action research and Ethnography [12] we identified which aspects are most problematic for the practitioners, i.e. based on a historical and situational context, from their 'own point of view'. Results from the first phase were used by the researchers together with the practitioners involved as input for the design of possible improvements. The outcome was the deliberation of measures that address the identified problems expected to improve the situation at hand. Improvements were implemented, and the researchers followed these improvements as participant observers. The results were evaluated continuously together with the practitioners.

In relation to this principle we found that adequate knowledge to interpret numbers in a critical and correct way required thorough work and relevant involvement (often it was only the test leader that was in a good enough position to make such interpretations). And since numbers actually can get in the way of communicating 'the message' we decided to use graphs without the use of numbers as a presentation form; graphs that give a quick summary of the knowledge experienced by the testers.

Principle 2 is to use organizational knowledge. Many metrics programs fail because there is a gap between work procedures as they are described, and the actual work practices, and thereby a lack of understanding of the organizational context. For the program to succeed, the actors must understand why they are collecting particular data, and even understand the organizational politics. This can be done by including as many of the affected employees as possible [7:295].

In our case, besides cooperating with interaction designers through CMD, we also targeted different organizational roles in our usability test metric efforts, i.e. this was a way of including many employees. Results from the UTUM were addressed to the following roles in the company: interaction designers, system- and interaction architects, high and low level management, product planning, and marketing.

A case study was performed to enable adjusting our work to the work practice of those stakeholders, to answer questions such as: Are any presentation methods generally preferred? Does the choice of methods change during different phases of a design and development project? Can results be presented in a meaningful way without the test leader being present? Is it possible to find factors in the data that allow us to identify disparate groups, such as Designers (D), represented by e.g. interaction designers and system and interaction architects, representing the shop floor perspective, and Product Owners (PO), including management, product planning, and marketing, representing the management perspective [18] (See also [18]).

We also included reasoning about quality in the sense of finding a balance between formality and organizational needs from an agile, practical viewpoint of "day to day" work. By "formal aspects" we relate to the fact that software engineers are often trained in a hierarchic model of knowledge where abstraction, together with the production of general solutions rather than specific solutions, is the key part. Improving formal aspects is important, and software engineering research in general has successfully emphasized this focus. However, improving formal aspects may not help design the testing that most efficiently satisfies organizational needs [9] and minimizes the usability testing effort. The formal view needs to be juxtaposed with reasoning based on a 'day to day' basis of organizational needs.

Formality is necessary to gain acceptance of the test results within the organization. Companies require both rapid value and high assurance. This cannot be met by pure agility or formal/plan-driven discipline; only a mix of these is sufficient, and organizations must evolve towards the mix that suits them best. This evolution has taken place during the whole period of the research cooperation, and it is apparent that this mix is desirable and necessary.

Program organization: metrics programs need not have a formal structure, but if a metrics program is to have a significant impact, you must address the organization of the program itself, and the incentives that should support it. **Principle 3** is to establish a project. To increase visibility and validate expense, the metrics program should be given the status of a formal project, with requirements for planning and reporting progress and with success criteria. The introduction of metrics into projects should also be planned [7:296]. In our case, the metrics program was part of a research cooperation where the company financed their participation in the research environment by direct work involvement (see [2]).

Principle 4 is to establish an incentive structure, so that those who report the metrics see some advantage in the program. The results of the program will hopefully benefit the employees in their daily work, but a more indirect approach can be taken, by using bonuses and awards to facilitate adoption of the metrics program. Tools and procedures should also be developed to simplify data collection, avoiding unnecessary burdens being placed on employees [7:297]. We did establish procedures to simplify data collection, but in our case, the metrics program was manned primarily by staff whose main task was user research, so collecting metrics was part of their work tasks rather than an additional burden. The same people were also involved in developing the usability test in cooperation with researchers from academia. Applying a work practice perspective helped us to adequately understand and adequately address other stakeholders.

Program design: just as the metrics organisation needs to be designed, so does the content. The metrics need to be organised and designed. **Principle 5** is to start by determining goals. Successful implementation needs clear goals from the beginning, since without these, the effort is impossible to manage and decisions on which measures to choose will be random [7:298].

In our case, overall usability goals were set in an evolutionary fashion over a long period of time. In relation to each development project where usability was tested, prioritized use cases where decided by clients, and metrics measured based on these. Iversen and Kautz [7:303] experienced difficulties to create a baseline for long-term measurement of improvements, since measurements were made whilst the organisation was improving. We also found creating a baseline difficult, partly because the marketplace was constantly changing, but also because the test methodology itself was in a constant state of change. It was only recently, when the test was beginning to stabilise, that efforts were made to create a baseline for usability testing. Due to the closure of the company this did not mature to the extent we wished for.

Principle 6 is to start simple. Starting with a small set of metrics is to be recommended, as systematically collecting data to use as the basis for decision making is difficult and complex. Also, it may be found that if the right metric is chosen, it will be possible to fulfil the predefined goals through the use of one simple measure, if you choose the right metric from the beginning [7:299]. In our case, we started on a small scale 2001. The results of the first testing process were seen as predictable, and did not at this time measurably contribute to the development process, but showed that the test method could lead to a value for usability. (See [6] for an overview). Thereafter the metrics evolved continuously over a period of time.

Communication: Metrics must be interpreted, since measuring the performance of professionals is not an exact science. A metrics program that attempts to do this can suffer from negative feelings, and this can be countered by communicating the objectives and results of the program, and encouraging discussion about the validity and reliability of the results. **Principle 7** is to publish objectives and collected data widely. The objectives of the program must be communicated as widely as possible, and results must be published broadly for as many relevant actors as possible. It is important to ensure that individuals are protected, and that metrics are not related to

performance evaluations. Published metrics must be based on reliable and valid data, in order to support fruitful discussion. As metrics may reveal unpleasant facts about an operation, it is important to use the figures in order to improve the organisation rather than to find scapegoats for the problems that have been uncovered [7:300].

In our case, all usability test results were published on the company's intranet as a news item for all employees to read and comment on. Test results were presented to specific teams connected to the test in question. Later on the company used the UTUM test result for the Usability Key Performance Indicator (KPI). The quality team at UIQ was responsible for collecting data for each KPI and the usability person in charge sent the latest UTUM data to this team. The Quality Team then put together a monthly KPI report to the Management team. The data were not specifically sensitive in that they did not collect metrics about the performance of individuals. They were aimed at measuring the usability of a product at a particular time and were not traceable to any individual within the organization. The user researcher also presented the metrics to two different groups of stakeholders: the Designers and Product Owners mentioned earlier [18].

Principle 8 is to facilitate debate. There should be a forum for discussing the metrics program and its results, to clarify how the material that is collected will be used, in order to prevent the creation of myths about the material and its use [7:301].

In our case, there was much discussion in the company surrounding KPI's, about what they did and could communicate. These were tough discussions, and strong opinions were often expressed, e.g. that KPI's might provide management with false impressions that could lead the organisation in the wrong direction. Also that the 'form' was focused on rather than the 'content', and what the consequences of this might be in time-critical situations. There were also opposing opinions, that KPI's could give the organisation a sharper focus. Most debates were about what was measured, the importance of it, and for whom the metrics were produced. And of course questions related to consequences for the staff were ventilated, i.e. what the introduction of specific measurements might mean for them and their roles in the long term.

Data usage: the data that is collected must be used to implement improvements or increase understanding about how the organization works. If data is not applied, those who supply them are not likely to supply further data, resulting in deterioration of the data quality. **Principle 9** is therefore to use the metrics results to gain insight into software processes, and correct the problems. If this is not done, the metrics program may turn into a procedure that merely adds to development overhead. If there are no consequences of poor results, the metrics program is unlikely to succeed, but it is important to bear in mind that software development cannot be measured precisely, and that the data must not be over-interpreted. However, recognizing trends from imprecise data is still better than having no data at all [7:302].

In our case there were two main channels and target groups for implementing improvements, i.e. the previously introduced groups of stakeholders captured as Product owners and Designers. UTUM became a company standard from January 2007, and was used in all software development projects. It was also frequently requested by external clients.

In relation to the first group, the Product owners (PO), data usage is focused on comprehensive documentation consisting of spreadsheets containing the formal side of metrics and qualitative data. It is important to note that we see this formal element in the testing as an increased use of metrics that complements the qualitative testing method side. Not the other way around. Metrics back up the qualitative findings that have always been the result of testing, and open up new ways to present test results in ways that are easy to understand without having to include contextual information. They make test results accessible for new groups. The quantitative data gives statistical confirmation of the early qualitative findings, but are regarded as most useful for PO, who want figures of the findings that have been reached. There is less pressure of time to get these results compiled, as the most important work has been done, and the critical findings are already being implemented. The metrics can also be subject to stringent analysis to show comparisons and correlations between different factors.

In relation to the second group, the Designers, the high tempo of software development in the area of mobile phones makes it difficult to channel meaningful testing results to the right recipient at the right time in the design process. To alleviate this problem, we have integrated the role of the test expert into the daily design process. Directly after or during a period of testing, the test leaders meet and discuss findings with this group. This can take place even before all the data is collated in spreadsheets. In this manner the test experts are able to present the most important qualitative findings to system and interaction architects within the organisation very soon after the testing has begun. It was also shown that changes in the implementation have been requested soon after these

meetings. An advantage of doing the testing in-house is to have access to the tester leaders, who can explain and clarify what has happened and the implications of it – just in time - when that information is desired by these actors. And the results of testing that is performed in-house can be channelled to the most critical issues, and the continual process of testing and informal relaying of testing results to designers leads to a short time span between discovering a problem and implementing a solution.

This concludes our comparison of the findings of Iversen and Kautz, and the findings from our research. The comparison is made here to help the reader assess the applicability of the lessons for companies attempting to implement software metrics programs. We now proceed with a discussion of what we have found, primarily where our study gives new insights into the principles, and what we can learn that may help organizations improve testing and improvement efforts.

6. Discussion and conclusions

Beginning with a comparison of the principles listed by Iversen and Kautz, and the results of our research, there are a number of findings that are interesting in light of the situation that we have found.

They found that an organization must adapt its culture to practices and a way of thinking where decisions are based on measurable phenomena rather than intuition and personal experience. This was also found in the case of UIQ. However, having said this, our experience shows that the test leader is a primary source of important knowledge that is based on intuition and personal experience. Most importantly, we found that the metrics based on these factors are just as relevant in the metrics program as the other types of metrics.

The success of a metrics program was found to be most likely in a situation where measurement does not place particular burden on the practitioners, and where incentives for participation were offered. In our case, there was a test leader who was assigned to the collection, analysis and presentation of test results, who could request further resources when necessary, and the testing work was considered to be an integral part of the development process, and was included in estimates made when planning project activities. This meant that the metrics activities were not seen as an extra burden on top of the everyday work. This is an important fact to consider when designing and implementing both testing programs and improvement processes, emphasising the importance of ensuring that measurements are performed by specialists who are well versed in the structure and operations of the organisation.

Successful implementation of SPI needs clear goals from the beginning or the effort will be difficult to manage and decisions on which measures to choose will be random. Our experience contradicts this, and shows that we must accept that the dynamics of the marketplace, where rapid change is becoming the norm, make it difficult to achieve stability. Therefore, it is important to adopt an agile approach to the design and performance of the testing or improvement program.

It is important to continually improve and evaluate the metrics program. This has always been an integral part of the development and operation of the UIQ metrics program, and has taken place through discussions and workshops together with researchers, practitioners, management, and even outside companies. It is crucial to adapt the principles to the environment and the situation at hand. The work that has been done here, leading to the success of the metrics program, and its adoption within the everyday development activities shows that this has been a successful strategy in our case.

Concerning the use of the data, this has been one of the main focuses of our recent research. We have found it to be of the greatest importance that the data, the results of the testing, are used in ways appropriate to the needs of the recipients. This demands both agile and formal results presented in different ways for different groups of stakeholders. The formal side of testing is necessary as a support for the agile side, and ensures that testing results are accountable within the organisation, thus raising the status of the agile results. This balance was not discussed amongst the principles listed by Iversen and Kautz.

When examining the above, we see that much of it has dealt with finding the necessary balance between agility and formality, to satisfy the requirements of different stakeholders. We also find that the test leader is a key figure, since this is a person who owns the requisite knowledge to understand and explain the results and their meaning, and can work as an advocate of the user perspective.

Particularly in a situation where rapid change is the norm, we have found that the main lessons to be learned from this comparison are that in measurement and improvement processes, it is vital to strike a balance between agility and formality, in both capturing requirements and communicating them to stakeholders. It is also vital to find the right people, who can function as test leaders, analysts, communicators, and advocates of the user perspective.

SPI deals with process, whilst UTUM deals with both product and process. The paper shows the role usability testing can play in both product and process improvement. Just as SPI has been found to be a useful vehicle for improvements in software engineering, so is usability testing found to be a useful vehicle, that is well in accordance with the principles of SPI. This is a fact that emphasizes the usefulness of usability testing, which is an area that has been neglected in the area of software testing.

The study is also an illustration of how two paradigms can interact, by showing how the metrics could be tailored to allow professionals schooled in a qualitative tradition to interact together with professionals schooled in a quantitative engineering tradition, and that it is possible to balance these two traditions that are often seen as conflicting.

ACKNOWLEDGMENTS

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the software development project "Blekinge – Engineering Software Qualities" [1] and the Vinnova Foundation in Sweden under a research grant for WeBIS [16], handling method development mixing Innovation and usability.

REFERENCES

[1] BESQ. *Blekinge - Engineering Software Qualities*. (2009-01-15) www.bth.se/besq.

[2] BESQ Projects, Use orientation for innovation and quality in mobile application development (2009-01-19) http://www.bth.se/tek/besq.nsf/pages/57357f868934a36dc125740f0044a244!OpenDocument

[3] Dittrich, Y., Rönkkö, K., Lindeberg, O., Erickson, J. and Hansson, C., Cooperative method development: Combining qualitative empirical research with method, technique and process improvement, In the Journal of Empirical Software Engineering, 13(3), 2008, pp. 231-260 (Jun. 2008),

[4] Grudin, J., The West Wing: Fiction can Serve Politics. Scandinavian Journal of Information Systems, 2003. 15, pp. 73-77.

[5] Grudin, J. and J. Pruitt. Personas, Participatory Design and Product Development: An Infrastructure for Engagement. in Participatory Design Conference, 2002. Toronto, Canada.

[6] Hellman, M. and Rönkkö, K., Is User Experience supported effectively in existing software development processes?, Proceedings of the 5th COST294-MAUSE Open Workshop Meaningful Measures: Valid Useful User Experience Measurement (VUUM 2008), Reykjavik, Island. 2008.

[7] Iversen, J. and K. Kautz, Principles of Metrics Implementation, in Improving Software Organisations: From Principles to Practice, L. Mathiassen, J. Pries-Heje, and O. Ngwenyama, Eds. 2001, Addison-Wesley: New York.

[8] Karlsson, L., Dahlstedt, A.G., Regnell, B., Natt och Dag, J., Persson, A Requirements engineering challenges in market-driven software development - An interview study with practitioners, In the Journal of Information and Software Technology, 49(6), . 2007, pp. 588-604.

[9] Martin, D., Rooksby, J., Rouncefield, M., Sommerville, I., 'Good' Organisational Reasons for 'Bad' Software Testing: An Ethnographic Study of Testing in a Small Software Company in ICSE '07. 2007. Minneapolis, MN: IEEE.

[10] Rönkkö, K., Hellman, M., Kihlander, B. and Dittrich, Y., Personas is not Applicable: Local Remedies Interpreted in a Wider Context, Proceedings of the Participatory Design Conference, PDC '04, Toronto, Canada, July 27-31, 2004, pp. 112-120.

[11] Rönkkö, K., Winter, J., and Hellman, M. Reporting user experience through usability within the telecommunications industry. Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering (Leipzig, Germany, May 13 - 13, 2008). CHASE '08. ACM, New York, NY, pp. 89-92.

[12] Rönkkö, K. Ethnography accepted for publication In *Encyclopedia of Software Engineerin*g (ed.) Phil Laplante, Encyclopedia Program, Taylor and Francis Group, New York, 2009.

[13] Seaman, C.B., Qualitative methods in empirical studies of software engineering. Software Engineering, IEEE Transactions on, 1999. 25(4): p. 557.

[14] Talby, D., et al., Agile Software Testing in a Large-Scale Project. IEEE Software, 2006. 23(4): p. 30-37.

[15] Use-Oriented Design and Development (U-ODD) (2009-01-19) http://www.bth.se/tek/U-ODD,

[16] WeBIS, research project (2009-01-19) http://www.bth.se/tek/webis.nsf/sidor/hem

[17] Winter, J., Measuring usability to satisfy stakeholder's needs: balancing agility and formality, Licentiate Thesis, Blekinge Institute of Technology, 2009

[18] Winter, J., Rönkkö, K., Ahlberg, M., Hinely, M. and Hellman, M. 2007. Developing Quality through Measuring Usability – The UTUM Test Package, 5th Workshop on Software Quality, at ICSE 2007, IEEE, Minneapolis, Minnesota.

[19] YouTube, UIQ, Usability test. (2009-01-19) http://www.youtube.com/watch?v=5IjIRlVwgeo.