SPI success factors within product usability evaluation

Jeff Winter*, Kari Rönkkö

School of Computing, Blekinge Institute of Technology, Soft Center, SE 372 25 Ronneby, Sweden

ARTICLE INFO

Article history:
Received 14 June 2009
Received in revised form 20 April 2010
Accepted 27 April 2010
Available online xxx

Keywords:
Software process improvement
Usability product metrics
Process metrics
Management
Software development
Software engineering
Human–computer interaction
User experience

ABSTRACT

This article presents an experience report where we compare 8 years of experience of product related usability testing and evaluation with principles for software process improvement (SPI). In theory the product and the process views are often seen to be complementary, but studies of industry have demonstrated the opposite. Therefore, more empirical studies are needed to understand and improve the present situation. We find areas of close agreement as well as areas where our work illuminates new characteristics. It has been identified that successful SPI is dependent upon being successfully combined with a business orientation. Usability and business orientation also have strong connections although this has not been extensively addressed in SPI publications. Reasons for this could be that usability focuses on product metrics whilst today’s SPI mainly focuses on process metrics. Also because today’s SPI is dominated by striving towards a standardized, controllable, and predictable software engineering process; whilst successful usability efforts in organisations are more about creating a creative organisational culture advocating a useful product throughout the development and product life cycle. We provide a study and discussion that supports future development when combining usability and product focus with SPI, in particular if these efforts are related to usability process improvement efforts.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

This article presents an experience report that is based on 8 years of our own experiences of implementing usability product metrics as a quality assurance approach. The material that it is grounded in has been collected in close cooperation with an industrial actor, in an action research based approach, using ethnographic methods and a case study approach.

Software engineers often believe that software development is negatively affected by measurements with no quantitative description of where we are and where we would like to go (Pfleeger and Atlee, 2006). Measurement is seen as being indispensable to introducing engineering discipline to software development, maintenance and use (Basili et al., 1994). Software testing permits measurements that give organisations better insight into the processes being monitored. Measurement can help us to understand what is happening, to control what is happening, and to improve processes, and products (Fenton and Pfleeger, 1998). Usability testing is one part of the testing area, although it is not a commonly addressed area within software engineering. This may be 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 product test situation, rather than for software development process aspects. To be a professional in usability requires extensive knowledge of interaction theory, user-research methodologies, and especially the principles of user testing. It can often take an interaction designer several years to build up this knowledge. Specialization might also be important here, as an individual’s orientation towards design or usability seems to require different personality types. This is visible in usability and interaction design education targeting industrial actors (Nielsen, 2002).

Another explanation might be that usability focuses on product metrics whilst software engineering mainly focuses on process metrics. It is recognised in software engineering that usability does not only affect the design of user interfaces, but the development as a whole, especially related to the requirement process. It is also recognised that both areas require own professional knowledge and approaches. Still, the interplay between these two fields and activities they advocate to be undertaken in software development has proved to be difficult (Hellman and Rönkkö, 2008a).

Today’s prevailing focus on process metrics has not always been the major trend. It has also been questioned how successful the process focused approach actually is. Based on an extensive review of quality management, organisational learning and software process improvement (SPI) literature regarding key factors for success, Dybå (2000a) identified three trends: The first trend is that current developments in quality management are heavily influenced by recent attention towards the improvement of production pro-

* Corresponding author. Tel.: +46 455385562.
E-mail addresses: jeff.winter@bth.se (J. Winter), kari.ronkkoo@bth.se (K. Rönkkö).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.04.066

Please cite this article in press as: Winter, J., Rönkkö, K., SPI success factors within product usability evaluation. J. Syst. Software (2010),
doi:10.1016/j.jss.2010.04.066
cesses. Hence, SPI is viewed mainly from one point of view. In
continuation Dybå claims that such a one-sided view causes confu-
sion and is a misunderstanding “... partly related to quasi-religious
and sectarian controversies and partly related to the most impor-
tant features of modern quality management: it draws attention
to the improvement of production processes and not simply to the
characteristic of the product.” Whereby, the original quality con-
cept principle, assuming that the characteristics of the final product
fall within decided tolerance, is opposed. The second trend is that a
recent revision of the ISO 9000 family of standards also follows the
same tendency, by distancing itself from the term “quality assur-
ance” and instead emphasising the adoption of a process approach
to “quality management”. The third trend is that no single model
has yet established itself as a basis for total quality management
theory.

In relation to the above, it is worth noting that aligning SPI goals
and action with explicit and implicit business goals and strate-
gies is one of the factors that has the strongest influence on SPI
success (Dybå, 2005). Success is dependent upon successfully com-
bining SPI and business orientation. The question is how this can
be achieved. Related to our usability focus it is well known that
usability and business orientation have strong connections; good
usability is part of the public’s perception of a company, affect-
ing the value of a brand and the share of the market. Products are
often publicly “talked about” in magazines and newspapers in terms of:
usability and quality of documentation, i.e. common usability aspects
(Bias and Mayhew, 2005, p. 18). It is also well known that usability efforts lead to increased
user productivity, decreased user errors, decreased training costs,
early changes in the development lifecycle, and decreased user sup-
port; which lead to business benefits in the form of increased sales,
decreased customer support, reduced costs of providing training
support, increased savings from making changes early in design
lifecycle, and finally increased perception of the value of company
by stakeholders (Bias and Mayhew, 2005). Here usability produ-
uct metrics provide one potential area for increased and improved
business, which can be related to SPI. Unfortunately the one-sided
focus on process that currently prevails in SPI, together with a lack
of publications discussing the combination of product and pro-
cess, is a hinder for positive development in this area. Even though
our study does not explicitly address SPI in agreement with the
product line approach that currently dominates (the rationalistic
paradigm), with standardized, controllable, and predictable soft-
ware engineering processes, we still believe that our study provides
knowledge here. On the other hand there are other views that relate
closer to our case, i.e. creating an organisational culture.

In software engineering in general it is believed that the correct
path follows imitating technical procedures identified by others.
Based on a quantitative study of 120 software organisations Dybå
suggests an approach that differs from most of existing SPI litera-
ture, i.e. to focus the SPI efforts instead on creating an organisational
culture within which technical procedures can thrive (Dybå, 2005).
An earlier study by Iversen and Kautz (2001) supports Dybå’s idea of
“thiving” by means of advocating the need to tailor meth-
ods to organisational needs. Iversen and Kautz have studied and
summarised recommendations and advice regarding SPI metrics
implementations, and suggest that whilst guidelines and frame-
works are useful, organisations must tailor methods to fit their
particular situation. This is particularly true since much of the
advice given disregards the fact that many companies do not
employ large numbers of software developers, and that organisa-
tional cultures differ between different countries. Advice aimed at,
e.g., North American companies may not fit in with organisational
Iversen and Kautz’s study is of particular relevance to us, since our
metrics efforts took place in Sweden, in a Scandinavian context
(see also Dybå’s critique of applying SPI results from North America
in European contexts in general, and particularly in Scandinavian
cultures (Dybå, 2005)).

Another topic of relevance for our study is exploitation and
exploration. Software organisations can engage in exploitation,
the adoption and use of existing knowledge and experience; or
exploration, the search for new knowledge either through imita-
tion or innovation. According to Dybå (2000b), which of these two
two becomes dominant depends on the size of the company. Smaller
companies often keep the same level of exploitation, independently
of whether they operate in a stable or turbulent environment,
whilst at the same time they increase their exploration in turbulent
environments. Larger companies (more than 200 developers) had
similar results for exploitation, but in contrast to smaller compa-

dies, they do not increase their levels of exploration when the level
of turbulence increases. Dybå explains that management groups
within large organisations still rely on their previous experience to
prepare for the future rather than exploring new possibilities. They
generate the same response even though stimuli change, and in a
sense keep doing what they do well, rather than risk failure (Dybå,
2000b). One suggested explanation for this is that the dominant
perspective in today’s software development is the rationalistic
paradigm emphasising standardized, controllable, and predictable
software engineering process. A large part of the SPI community has
also promoted this rationalistic view, e.g. through explicit support
of the use of rigorous statistical techniques. Without suggesting the
abandonment of the discipline altogether, it is still claimed that
software development practice does not conform well to such a
rationalistic view (Dybå, 2000b).

Of course, both creativity and discipline are needed. Dybå
observed that small successful organisations tend to put more
emphasis on exploring new possibilities, embracing diversity and
creativity, whilst larger successful organisations tend to emphasise
“best practices” and formal routines (Dybå, 2003). This is a balance
that is challenging. In our study we refer to these two sides by using
the terms agile and formal (Winter and Rönkkö, 2009). This does not
imply that we see agile processes as being informal or unstructured,
but because the word “formal” is more representative than e.g.
“plan driven” to characterise the results of testing and how these
results are presented to certain stakeholders. When we started the
cooperation with the studied company it was a small company with
approximately 50 employees. When we concluded our cooperation
it was a large company with approximately 400 employees (more
than 200 developers; Dybå’s definition in (Dybå, 2000b)). In rela-
tion to the usability evaluation framework developed by us, we kept
an explorative mindset throughout its implementation with help of
a Cooperative Method Development (CMD) (Dittrich et al., 2007)
research approach. We also adjusted our test evaluation framework
to satisfy management needs for standardized, controllable,
and predictable software engineering process.

The present article will use our achieved knowledge and rec-
mendations found within the SPI literature to discuss 8 years
of our own experiences of implementing usability product metrics
as a quality assurance approach, including growth in an industrial
organisation and strategies to support requests from management
and business. The experiences presented of introducing usability
metrics in an industrial organisation are also well in line with
Dybå’s identified need of re-introducing product quality assurance
in SPI. We provide an example of what it might mean to cre-
ate an organisational culture within which technical procedures
can thrive. Our discussion is framed within the area of usability
and organisational improvement, which as we mentioned earlier
seems to be an unusual topic in software engineering, despite its
close relation to business and the recognised importance of com-

bining SPI and business orientation. In this article we discuss our
experiences of influencing and improving a software development

Please cite this article in press as: Winter, J., Rönkkö, K., SPI success factors within product usability evaluation. J. Syst. Software (2010),
doi:10.1016/j.jss.2010.04.066
organisation by applying the language of usability metrics. We discuss how we tailored our metrics, based on work heavily influenced by a creative qualitative tradition, to fit an organisation influenced by a quantitative rationalistic paradigm and software engineering tradition.

Iversen and Kautz’s 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. Dyba’s study provided extended knowledge of success criteria for SPI based on a quantitative survey where 120 software organisations helped verify previously known key factors for success in SPI. We have found many interesting parallels to their work and results and the work performed and findings made whilst developing and implementing our usability framework called UIQ Technology Usability Metrics (UTUM). UTUM is a usability test framework developed in an action research (Avison et al., 1999) fashion, based on the principles of CMD (Dittrich et al., 2007). The research was a cooperation between, in particular, interaction designers from UIQ Technology AB and researchers (the present authors) from Blekinge Institute of Technology. This cooperation is presented in more detail later in this article. The comparison made, besides discussing usability product metrics experiences and SPI, also aims to help the reader assess the applicability of the lessons intended for companies attempting to implement software metrics programs. We believe this paper is of value for organisations in search for a starting point to tailor their usability metrics or introducing SPI. 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 and the research partners. Thereafter we present the UTUM. After this we compare the areas and principles listed by Iversen and Kautz and Dybå with the conclusions we have drawn from our research and implementation of usability metrics. Finally, we discuss some issues we confronted and conclude our main experiences.

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 Karlsson et al., 2007 for an 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 may be targeted (Grudin and Pruitt, 2002), there is an unwillingness to exclude other potential end-user categories (Rönkkö et al., 2004). When developing software designed for single organisations, 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. A study demonstrating how socio-political circumstances made a usability and participatory design technique a too precise technique to be applied is Rönkkö et al. (2004). 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. Taken 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 in 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 code line. 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 (U-ODD, 2008) which was also part of the research environment BESQ (2008). U-ODD approaches software engineering via user-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 (Seaman, 1999). 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.

Although UIQ as a whole worked as a traditional development organisation, influenced by a software engineering tradition, the particular work that we have studied at UIQ has been performed according to ISO 13407:1999 “Human-centred design processes for interactive systems” (International Organisation for Standardization, 1999), an international standard concerning usability and human-centred design. ISO 13407:1999 “provides guidance on human-centred design activities throughout the life cycle of computer-based interactive systems. It is aimed at those managing design processes and provides guidance on sources of information and standards relevant to the human-centred approach” (International Organisation for Standardization, 1999, p. 1). It defines human-centred design (HCD) as “an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity which incorporates human factors and ergonomics knowledge and techniques” (International Organisation for Standardization, 1999, p. iv).

Bevan (2009a) states that international standards for usability should be more widely used, and that if you are going to read only one standard, it should be ISO 9241-210 or ISO 13407. These provide a high level framework for usability work. The activities that are given in ISO 13407 are present in many design methods, and therefore the standard offers limited practical guidance; however,
ISO 18529 contains a detailed list of activities and base practices, and including these activities in a design process ensures a continuous focus on the users of the system (Kesseler and Knapen, 2006). It can be used in the specification, assessment and improvement of human-centred processes in system development and operation, and is intended to make the contents of ISO 13407 available to those familiar with or involved in process modelling (Bevan and Bogomolni, 2000). Bevan (2009a) also states that ISO 18529 or ISO 18152 should be used as a tool to improve usability capability within an organisation. This is supported by Earthy et al. (2001) who claim that the implication of ISO 13407 and ISO 18529 is that there is a professional responsibility for software engineers, system engineers and usability professionals to adopt good practice as defined in ISO 13407 and ISO 18529 as their baseline. This was not implemented in this company, so we cannot provide any details of it in this case, but we return to the reasoning behind it in our conclusions and discussion.

HCD has four main principles and four main activities (see Fig. 1 for an illustration of the activities), which are closely connected. In the following, we look at these principles and activities, and how they are connected to the work performed at UIQ. The inclusion of UTUM in an HCD-centred process was important, since this has allowed the results of usability testing to have an impact on design and development processes within the company.

The first principle in HCD is active involvement of users. This needs a clear understanding of user and task requirements. User involvement gives knowledge about the use context and how users are likely to work with a future system or product. Even when developing generic or consumer products, which by their nature have a dispersed user population, it is still important to involve users or “appropriate representatives” in development, in order to identify relevant requirements and to provide feedback through testing of proposed design solutions. The first activity in HCD is therefore understanding and specifying the use context. The context should be identified in terms of the characteristics of the intended users, the tasks the users are to perform and the environment where the users are to use the system.

In the context of this company, this was not a trivial task. The company worked in a rapidly changing mass market situation, with a product aimed at satisfying several market segments simultaneously. This is a situation that demands knowledge of end users and their current and expected future use of mobile phones, in a situation that can change over a short period of time; for example, if a competitor releases a new model with new and exciting features. Fortunately, there was a great deal of branch knowledge within the company, and within the different companies that the company worked together with. This meant that the people who were involved were familiar with the market, and were aware of what was happening in the marketplace, even though details of coming models are shrouded in secrecy. There is a great deal of information to be gleaned from many sources, and the company gained information about trends through participation in trade fairs, attending conferences and workshops, and kept abreast of publications within the field. They also gained input from mobile phone manufacturers, regarding future models that were designed to use the UIQ user interface. This information together with other branch knowledge meant that it was possible to understand and specify the use context.

The second principle is an appropriate allocation of functions between users and technology, which means the specification of which functions should be carried out by the users and which by the technology. The second activity is therefore specifying the user and organisational requirements. A major step in most design processes is specifying functional and other requirements for products or systems, but in HCD this must be extended to create statements of user and organisational requirements in relation to the use context. This includes identifying relevant users, formulating clear design goals, and providing measurable criteria for testing the emergent design, that can be confirmed by users or their representatives. Once again, in a mass market situation, where the product is aimed at many potential customers, this is a potentially difficult task. In our study, the use context, and therefore the requirements, is dependent on the type of user and their use of the phone. Once again, there was an extensive knowledge base regarding phone use, much of which was gained through having access to panels of users, and people who were recurrent phone testers. The company had access to user representatives who were long-term users, and focus group meetings and workshops were organised together with users. In this field, since the use of mobile phone technology is primarily individual and is not explicitly affected by organisational issues, we did not look in depth at organisational requirements. Therefore what follows deals explicitly with user requirements.

The third principle is the iteration of design solutions. Iterative design allows the results of testing “real world” design solutions...
to be fed into the design process. Combining user involvement with iterative design minimizes risks that systems do not meet requirements, and feedback from users is an important source of information. The third activity is therefore to produce design solutions. Potential solutions are produced by drawing on the state of the art, the knowledge of the participants, and the context of use. The process involves using existing knowledge to develop design inputs with multi-disciplinary input, concretizing the design through simulations, mock-ups, etc., presenting the design to users and allowing them to perform tasks — real or simulated — and altering the design in response to user feedback. This process should be iterated until design objectives are met. The interaction design department at UIQ has been a good example of the type of actor involved in this process. They had a mix of many different competencies, from graphical designers, interaction designers, and usability experts, working closely not only with customers and end users, but also together with development teams within their own organisation. This method of working allows the design and development of features that are acceptable in the marketplace, and are feasible to develop. The way in which UTUM has been used in this phase will be detailed more clearly in Section 4.

The fourth principle concerns multi-disciplinary design, and states that HCD requires a range of personnel with a variety of skills in order to address human aspects of design. The multi-disciplinary design teams need not be large, but should be capable of making design trade-off decisions, and should reflect the relationship between the customer and the development organisation. The fourth activity is to evaluate the design against requirements. Evaluation is an essential step in HCD. It can be used to provide feedback that can improve design, can assess whether design objectives have been met, and can monitor long term use of the system. It should take place at all stages of the system life cycle (International Organisation for Standardization, 1999). This is one of the activities where UTUM is the central factor in our study, allowing the evaluation and validation of solutions, from a feature level to the level of the complete product. For more information regarding this, see Section 4.

Our cooperation with UIQ technology has focused on usability issues at the company (Rönkkö et al., 2009). It began with a 12-month ethnographic (Rönkkö, 2010) study that led to an attempt to implement the participatory design method (PD) Personas (see e.g. Grudin and Pruitt, 2002; Pruitt, 2003) in 2001. The Personas attempt was abandoned in 2004, as it was found not to be a suitable method within the company for branch related reasons that can be found in (Rönkkö et al., 2004). To our surprise, the original reason for introducing the PD method had disappeared during our attempt at method implementation, and internal socio-political developments had solved the power struggle that the PD method was aimed to mediate (Rönkkö et al., 2008).

In parallel with the attempt to implement Personas in 2001, we also started to approach the new company goals focusing on metrics for aspects of the system development process. In our case these were focused on usability. Therefore, an evaluation tool was developed by the head of the interaction design group at Symbian (at that time the parent company for UIQ), Patrick W. Jordan, together with Mats Hellman from the Product Planning User Experience team at UIQ Technology, and Kari Rönkkö from Blekinge Institute of Technology. This evaluation tool was the first prototype of UIQ Technology Usability Metrics (see Winter, 2009, chapter 4 and Appendix A). The first testing took place during the last quarter of 2001. The goal was to see a clear improvement in product usability, and the tests were repeated three times at important junctures in the development process. The results of the testing process were seen as rather predictable, and did not at that time contribute in a measurable fashion to the development process, but showed that the test method could lead to a value for usability.

The test method was further refined during 2004–2005 leading to the development of UTUM v 1.0 which consisted of three steps. The users involved in the testing could be chosen from the group of long-term users that were mentioned previously, or could be chosen from a list of people who had shown interest in participating in testing, or who had been recommended by other users. In the first step, a questionnaire about the user’s usage of a device was used to collect data that could be used to prioritize use cases and for analysis and statistical purposes. The use cases could then be decided by the questionnaire that the user had completed (in this choice Jordan’s level of functionality is visible) or decided in advance by the company, if specific areas were to be evaluated. After each use case the user filled in a short satisfaction evaluation questionnaire explaining how that specific use case supported their intentions and expectations. Each use case was carefully monitored, videotaped if this was found necessary and timed by the test expert. The second step was a performance metric, based on completion of specified use cases, resulting in a value between 0 and 1. The third step was a component metric based on the SUS (Brooke, 1986), also resulting in a value between 0 and 1. These values were used as parameters in order to calculate a Total Usability Metric with a value between 0 and 100. Besides these summative usability results, the test leader could also, through his/her observation during the test, directly feedback formative usability results to designers in the teams within the organisation, giving them early user feedback to consider in their improvement and redesign work.

In 2005, Jeff Winter joined the cooperative method cooperation as a new Ph.D. student, and began to study the field of usability, observed the testing process, and began interviewing staff at UIQ. At the same time, a usability engineer was engaged in developing the test. In an iterative process during 2005 and 2006, a new version of the test was produced, UTUM 2.0, which included more metrics, and new ways of interpreting and calculating the metrics. At this time, we also started to consider how the testing is related to the agile movement and the principles of agile software (Agile, 2001), and the balance between agility and formality, and to the differentiation of usability results for different stakeholders needs (Winter and Rönkkö, 2009).

The process of cooperation has been action research (Avison et al., 1999) according to the CMD methodology (Dittrich et al., 2007). Action research is a “vague concept but has been defined as research that involves practical problem solving which has theoretical relevance” (Mumford, 2001, p. 12). Action research is a combination of practice and theory where researchers and practitioners cooperate through change and reflection in a problematic situation within a mutually acceptable ethical framework (Avison et al., 1999). It is thus a merger of research and praxis, and on this basis it produces findings that are exceedingly relevant (Baskerville and Wood-Harper, 1996). It involves not only gaining an understanding of a problem, and the generating and spreading of practical improvement ideas, but also the application of the ideas in a real world situation and the spreading of theoretical conclusions within academia (Mumford, 2001). The researcher expects to generate knowledge that will enhance the development of models and theories (Baskerville and Wood-Harper, 1996). The purpose of action research is to influence or change some aspect of whatever the research has as its focus, and improvement and involvement are central to it (Robson, 2002, p. 215). A central aspect of action research is collaboration between researchers and those who are the focus of the research, and their participation in the process, and the terms participatory research and participatory action research are sometimes used as synonyms for action research (Robson, 2002, p. 216). Action research often takes the form of a spiral process, involving observation, planning a change, acting and observing what happens following the change, reflecting on processes and consequences, then plan-
CMD is based on the assumption that software engineering is heavily influenced by its social contingencies. The development of CMD was motivated by a discontent with the way in which existing research approaches to software engineering and information systems addressed use-oriented software development, since they did not address the questions of how practitioners tackle their everyday work, from a shop floor point of view, or how methods, processes and tools can be improved to address practitioners’ problems, from their own point of view (Dittrich et al., 2007).

CMD is a domain-specific adaptation of action research. 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. Although it is motivated by an interest in use-oriented design and development of software, it is not specific for these methods, tools and processes. It is a three-phase framework that can be further detailed in relation to each specific project, but it can even be limited to a single phase if no improvement is decided upon. In phase 1, Understanding Practice, the research begins with empirical investigations, in order to understand and explain existing practices and designs from a practitioner’s point of view, based on their historical and situational context. The intention is to identify aspects that are problematic from the practitioner’s point of view. In phase 2, Deliberate Improvements, the results from the first phase are used in a cooperative fashion by the researchers together with the practitioners involved, as an input for the design of possible improvements. This phase results in the deliberation of measures to address some of the identified problems. These measures are expected to improve the situation at hand. In phase 3, Implement and Observe Improvements, improvements are implemented. The researchers follow these improvements as participant observers. The results are evaluated together with the practitioners, and the concrete results are summarised for the companies involved. They build a base for the researchers to evaluate the proposed improvements (Dittrich et al., 2007).

In addition to the three-phase research cycle, CMD is built upon a number of guidelines. These are:

- Ethnomethodological and ethnographically inspired empirical research, combined with other methods if suitable.
- Focusing on shop floor software development practices.
- Taking the practitioners’ perspective when evaluating the empirical research and deliberating improvements.
- Involving the practitioners in the improvements (Dittrich et al., 2007).

This CMD approach has been used in cooperation with UIQ Technology AB in a large number of cycles during the period between 2001 and 2009, which have also had a profound impact on the development of CMD. For a more detailed explanation of CMD, and examples of its use in practice, see Dittrich et al. (2007, 2005). The following presents an example of one of the CMD cycles that was applied in the research cooperation between UIQ technology and U-ODD.

In discussions during late 2005, we introduced the concept of the Kano model (see e.g. Sauerwein et al., 1996; CQM, 1993) into discussions together with the participants who were involved in developing the UTUM test. The following can also be read as an illustration of one of the CMD cycles applied in this study.

The first phase began in 2005 when the researcher was introduced to the company, and began his studies. He was given a workplace at the company, and spent time to become acquainted with the company and its organisation, whilst doing background studies of the area of interest. During this period, he interviewed members of staff, observed working procedures, and studied the development methodologies and project models in use at the company. At this stage, more than ten interviews were performed, and the roles included interaction designers and architects, technical writer, graphic designer, usability test leader, the head of the interaction design department, and the head of one of the development departments, responsible for Personal Information Management (PIM) applications. Concurrently, he performed literature studies in order to become acquainted with the theoretical background to the field.

The second phase was based on the work performed in phase one. One concrete example of this is that the researcher identified methods for presenting the results of usability testing as an area of interest, and had studied the theory and use of the Kano model (Sauerwein et al., 1996). Based on this, research articles were sent to practitioners at the company, and he organised discussion and seminars to examine how this could be of use in the usability evaluation framework that was being developed. As a result of the discussions and seminars, a new way of presenting the results of the usability testing was formulated, presenting the metrics as plots in a 4-field diagram (see Fig. 2).

In phase three the researcher observed how the changes in presentation had taken place, took part in meetings where the results were presented to and discussed with senior managers, and discussed these changes with the practitioners, to gain an understanding of what the improvements meant in the everyday work of testing and presenting the results of usability testing. These cycles conformed to the principles given above, by using a combination of ethnographically inspired fieldwork, combined with document studies, in order to grasp and understand the practices of the workers who were involved in the everyday work of testing and designing the evaluation framework. The work was performed together with the practitioners, and the improvements were designed, tested and evaluated together with these practitioners.

4. The UTUM test

UTUM is one concrete result of the research cooperation mentioned above. It is an industrial application developed and evolved through long-term CMD cooperation between BTH/U-ODD and UIQ Technology. It is a usability 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”. Hornbæk (2006) states that amongst the challenges when measuring usability are to distinguish and compare subjective and objective measures of usability, to study correlations between usability measures as a means for validation, and to use both micro and macro tasks and corresponding measures of usability. Emphasis is also placed on the need to represent the entire construct of usability as a single metric, in order to increase the meaningfulness and strategic importance of usability data (Sauro and Kindlund, 2005). UTUM is an attempt to address some of these challenges in the industrial environment. Due to space limitations, we cannot give an extensive presentation of UTUM here. It is presented in greater detail in (Winter, 2009, chapter 4 and Appendix A). A video demonstration of the test process (ca. 6 min) can be found on YouTube (BTH, 2008).

Briefly, the test is performed as follows: the test is performed in an environment that feels familiar to the tester, in order that he or she feels at ease. In this case, we call the user a tester, as it is he or she who is testing the device, under the guidance of a usability expert, who we call the test leader. The tester is welcomed by the test leader, who explains the purpose and performance of the test. In the first step, the tester fills in a form that records their personal details and usual phone usage patterns. This information can be used to choose the use cases that are included in the test, or these use cases can be predetermined. The next step involves introducing the tester to the phone or phones to be tested. If more than one phone is tested, then all of the use cases are performed on one phone, before proceeding to the next phone. For each device, the tester is given a few minutes to get acquainted with the phone, to get a feeling for the look and feel of the product. The tester is then asked to fill in a Hardware Evaluation form, a questionnaire based on the System Usability Scale (Brooke, 1986), which records their immediate impression of the phone. Then the tester is asked to perform the chosen use cases on the phone, whilst the test leader records details of the use case performance, and makes a judgement of the course of events. After each use case, the tester fills in another questionnaire, concerning task effectiveness, regarding the phone in relation to the use case. Between use cases the test leader can discuss with the tester what happened during the performance of the use case. In the final step, the tester fills in a System Usability Scale (Brooke, 1986) questionnaire that expresses the tester’s opinion of the phone as a whole.

This process is repeated until all of the phones are tested. The data collected in the test, both qualitative and quantitative, are used to calculate a number of metrics for effectiveness, efficiency and satisfaction, which are then used to produce usability KPIs. As mentioned previously, the initial results of UTUM testing are useful in the fourth phase of an HCD process (International Organisation for Standardization, 1999), when evaluating designs against requirements. The results that cannot be used in the immediate stage of development serve as input in the third phase of the next cycle in the process, when producing design solutions.

UTUM has a number of distinctive characteristics. The first two of these concern the relationship to users, and how user input is received and perceived, whilst a further two deal with software development practice, concerning organisation 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 (Rönkkö, 2010). 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 who help with the testing are referred to as testers, because they are the ones who are actually performing the test. The representative of the development company is referred to as a test leader, or test expert, emphasising 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 (Rönkkö, 2010) 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 summarised by quantitative methods, giving decision makers results in a 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 organisational 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 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. We have collected metrics for all of these aspects, based mainly on the industrial partner’s reliance on the ISO 9241-11:1998 (International Organisation for Standardization, 1998), where usability is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. This is probably the best-known definition of usability (Bevan, 2009b). Effectiveness is the accuracy and completeness with which users achieve specified goals. Efficiency concerns the resources expended in relation to the accuracy and completeness with which users achieve goals. Efficiency in this context is related to productivity rather than to its meaning in the context of software efficiency. Satisfaction concerns freedom from discomfort, and positive attitudes towards the use of the product. ISO 9241-11:1998 states that it is necessary to identify goals and decompose effectiveness and satisfaction and components of the use context into sub-components with measurable and verifiable attributes when specifying or measuring usability. These metrics for effectiveness, efficiency, and satisfaction were used as key performance indicators (KPI) of product usability.

The inclusion of all of these metrics is based on the fact that it has been seen as important to consider effectiveness, efficiency and satisfaction as independent aspects of usability, and that all three should be included in usability testing (Frokjær et al., 2000). Accord-
ing to (Bevan, 2006), when the purpose of a test is deciding whether a product has adequate usability, then measures should focus on the end result as defined in ISO 9241-11, and take into account the three aspects mentioned above. In a review of experimental studies, Frøkjær et al. (2000) found that many experimental studies of complex tasks account for only one or two aspects of usability, relying on assumptions of correlations between aspects of usability. However, there is some debate about whether there are correlations between these aspects, and, e.g., Sauro and Kindlund (2005) suggest that there are correlations, and propose a single score for usability metrics based on their findings. In work in progress, we are currently examining the metrics collected in a series of tests to determine whether we can find correlations between the different aspects of usability measured in the test.

If we return again to the principles and activities of HCD (International Organisation for Standardization, 1999), it is in the fourth phase that the initial results of UTUM testing are used. This is the phase where designs are evaluated against requirements. The results of testing are both in the form of metrics and as knowledge possessed by the test leader. These results can be communicated in different fashions to different stakeholders, on both a design and development level, and also on a management level. They affect features of the product that are soon to be included in a version for release to the market. However, some of the results dealt with aspects of the product that are no longer possible to influence or change at this particular stage of development, and these results are seen as candidates for future use when designing and developing design solutions. Thus, it is in the third phase, which allows the results of testing “real world” design solutions to be fed into the design process, that the results of previous cycles of UTUM testing can be used, as a form of existing knowledge, allowing the use of results that were impossible to include in the development that took place when the testing was performed.

Thus, we see that UTUM is part of a HCD based development process, and gives a demonstration of quality of use, defined as the extent to which a product satisfied stated and implied needs when used under stated conditions (Bevan, 1995), from the customer and end-user point of view (Winter et al., 2007). UTUM became a company standard from January 2007, and was used in all software development projects. It was also frequently requested by external clients. So far, it has mainly been 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 telecommunications branch called WeBIS, with the purpose of producing industrially viable methods for user generated innovation in ICT services (WeBIS, 2008).

5. SPI vs. usability testing metrics

In this section, we compare the findings of Iversen and Kautz (2001) (see Table 1) 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. We also set our work in relation to the key organisational factors for success in SPI as listed by Dybå (2005) (see Table 2).

In the following, we discuss how Iversen and Kautz’ five areas and nine principles for successful SPI compare with our research and our implementation of usability metrics. The areas and principles are dealt with in the same order as they are listed by Iversen and Kautz; interleaved in this, we also discuss our findings in relation to Dybå’s key factors for success in SPI.

### Table 1

<table>
<thead>
<tr>
<th>Area</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>1. Use improvement knowledge</td>
</tr>
<tr>
<td></td>
<td>2. Use organisational knowledge</td>
</tr>
<tr>
<td>Organisation</td>
<td>3. Establish a project</td>
</tr>
<tr>
<td></td>
<td>4. Establish incentive structures</td>
</tr>
<tr>
<td>Design</td>
<td>5. Start by determining goals</td>
</tr>
<tr>
<td></td>
<td>6. Start simple</td>
</tr>
<tr>
<td>Communication</td>
<td>7. Publish objectives and collect data widely</td>
</tr>
<tr>
<td></td>
<td>8. Facilitate debate</td>
</tr>
<tr>
<td>Usage</td>
<td>9. Use the data</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Key success factors for SPI (Dybå, 2005).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key factors</td>
</tr>
<tr>
<td>Business orientation</td>
</tr>
<tr>
<td>Involved leadership</td>
</tr>
<tr>
<td>Employee participation</td>
</tr>
<tr>
<td>Concern for measurement</td>
</tr>
<tr>
<td>Exploitation of existing knowledge</td>
</tr>
<tr>
<td>Exploration of new knowledge</td>
</tr>
</tbody>
</table>

5.1. Area 1: knowledge

When implementing metrics, it is important to draw on different kinds of knowledge, including knowledge about software improvement and organisational change, but also knowledge about the actual organisation where the metrics program is to be implemented. To successfully implement metrics programs, members of the organisation should be knowledgeable about the art of software metrics, software improvement, software engineering, and the process of organisational 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 (Dittrich et al., 2007). With the help of action research (Avison et al., 1999) and ethnography (Rönkkö, 2010) 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.

**Principle 2** is to use organisational knowledge. Many metrics programs fail because there is a gap between work procedures as they are described, and the actual work practices, and thereby there is a lack of understanding of the organisational context. For the program to succeed, the actors must understand why they are collecting particular data, and even understand the organisational politics. This can be done by including as many of the affected employees as possible (Iversen and Kautz, 2001, p. 295). This principle is connected to several of Dybå’s key factors. The first is employee participation, which is the extent to which employees use their knowledge and experience to act and take responsibility for SPI (Dybå, 2005, p. 412). This was found to be the factor that had the strongest influence in predicting SPI success, perhaps because people tend to support what they have participated in creating (Dybå, 2005, p. 419). The use of organisational knowledge is also connected to the factor of business orientation, which concerns the extent to which goals and actions are connected to explicit and implicit business goals and strategies (Dybå, 2005, p. 411).
Organisational knowledge is created through the experience and experimentation of members of the organisation, and it is important that both practitioners and researchers direct their efforts towards understanding the knowledge that is shared between groups within the organisation. This demands respect for the expertise of others, an understanding of the relevance of this expertise, and sufficient knowledge of the other group’s problems to be able to communicate about them. In its turn, this demands that the members of the different groups have had actual experience of the activities of the other groups (Dybå, 2005).

This supports the focus that our research has placed on participatory design (Schuler and Namioka, 1993). In our case, this has meant that, besides cooperating with interaction designers through CMD, we also targeted different organisational roles in our usability test metric efforts, i.e., this was a way of including many employees at many levels. 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.

We performed a case study 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 (Winter, 2009) (see also Winter et al., 2008 for more background to this). In our cooperative method implementation, we have included reasoning about finding a balance between formality, as we have defined it previously, and organisational needs from an agile, practical viewpoint of “day to day” work.

When referring to “formal aspects” we relate to the fact that software engineers are generally trained in a hierarchic model of knowledge where abstraction, together with the production of general solutions rather than specific solutions, is the key part. The software engineer’s systematic work often aims to produce an overview as a basis for simplifying, framing and explaining suggested findings/solutions from an objective and repeatable outside point of view (see also Rönkkö, 2010: Sections 3 and 4). Improving formal aspects is important, and software engineering research in general has successfully emphasised this focus. However, improving formal aspects may not help design the testing that most efficiently satisfies organisational needs (Martin et al., 2007) and minimizes the usability testing effort. A consequence of using “best practice” models, which is closely connected to a concentration upon “formal aspects”, is the fact that routines can become institutionalised. Thereby, they become a hinder to the exploration of alternative routines, and can become a barrier to improvement (Dybå, 2003). The emphasis we have placed on agility is a result of the concentration that we have had on examining the practical viewpoint of the study participants.

The conflict between formality and agility is connected to the ideas regarding learning strategies found in (Dybå, 2005, p. 412) where Dybå states that a learning strategy deals with the way in which an organisation is engaged in the exploitation of existing knowledge or the exploration of new knowledge. This deals with finding a balance between discipline and creativity, where relying too much on existing knowledge can limit the behaviour that is necessary for improvisation, whilst too much risk taking can lead to failure through fruitless experimentation (Dybå, 2000b). Dybå argues that both exploitation and exploration are necessary, involving balancing the improvement of an existing skill base with experimentation around new ideas. Both of these modes of learning were found to be connected to SPI success (Dybå, 2005).

The concept of exploitation is related to the formal aspects of our testing results, where the group we have identified as Product Owners has a long-term need for structured information allowing them a clear view of the product and how it is developing over time. However, the formal view needs to be juxtaposed with reasoning based on a ‘day to day’ basis of organisational needs. The latter viewpoint has been recognised in software engineering in the form of: (a) “agility”, by practitioners in the field; and (b) “work practice”, by sociologists and software engineers studying the influence of human aspects in software engineering work (for ‘agility’ and usability testing Winter et al., 2007; Talby et al., 2006). For examples of ‘work practice’ and software engineering see (Rönkkö, 2010, Section 8; Martin et al., 2007; Rönkkö et al., 2005; Rönkkö, 2002). This is connected with the exploration of knowledge, which includes such factors as innovation, creativity, flexibility, and diversity (Dybå, 2005). The concept of exploration also appears to be connected to the concept of employee participation.

A clarification is in order so that our intention is not misunderstood. It would be difficult to gain acceptance of the test results within the whole organisation without the element of formalism. In sectors with large customer bases, companies require both rapid value and high assurance. This cannot be met by pure agility or through formal/plan-driven discipline; only a mix of these is sufficient, and organisations must evolve towards the mix that suits them best. In our case this evolution has taken place during the whole period of the research cooperation, and reached a phase where it became apparent that this mix is desirable and even necessary.

Based on studies of work practice we find that we have managed to strike a successful balance between agility and formalism that works in industry and that exhibits qualities that can be of interest to both the agile and the software engineering community (see also Winter et al., 2007).

5.2. Area 2: program organisation

If a metrics program is to have a significant impact, you must address the organisation of the program itself, and the incentives that should support it. This is dependent on what Dybå calls involved leadership, which is the extent to which leaders at all levels in an organisation are committed to and participate in SPI (Dybå, 2005, p. 411). Dybå states, however, that management commitment need not go beyond the level of allocating the necessary resources. In our case, the commitment of the leadership is shown in several ways. One of these is that the company provided resources for participation in the BESQ research cooperation (BESQ, 2008). In this particular case, the company contracted to co-finance their participation through the work performed by the interaction design team, and the staff that had usability testing and the development of the usability test framework as part of their everyday work activities.

**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 (Iversen and Kautz, 2001, p. 296). This is also connected to the key factor of employee participation, which is the extent to which employees use their knowledge and experience to act and take responsibility for SPI (Dybå, 2005, p. 412).

In our case, the metrics program was, as previously mentioned, part of a research cooperation where the company financed their participation in the research environment by direct work involvement (see BESQ, 2008).
Principle 4 is to establish an incentive structure, so that those who report the metrics see some advantage in their participation 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 (Iversen and Kautz, 2001, p. 297).

In our case, it was not necessary to provide an incentive structure, as the work performed by the participants was a part of their everyday planned work activities and they were positive to the research cooperation together with the researchers. We did establish procedures to simplify data collection, but as mentioned, 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. These were the same people who were 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.

5.3. Area 3: program design

Just as the metrics organisation needs to be designed, so does the content. The metrics need to be organised and designed. This is also connected to Dybå’s key factor of business orientation. It is also dependent on management commitment, where the role of managers and leaders is to create a situation that is supportive of learning and SPI (Dybå, 2005).

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 (Iversen and Kautz, 2001, p. 298). This is closely connected to Dybå’s key factor of business orientation, and it concerns the need to focus on the alignment of SPI goals with business goals in both the long and the short term. If the SPI strategy is not kept in line with the business strategy, there is a risk that the processes simply become a burden (Dybå, 2005).

In our case, we have always concentrated on adapting the metrics to the business needs and strategy of the organisation. In relation to each development project where usability was tested, prioritized use cases were decided by clients, and metrics measured based on these. The overall usability goals were set in an evolutionary fashion over a long period of time. Iversen and Kautz (2001, p. 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 fulfill the predefined goals through the use of one simple measure, if you choose the right metric from the beginning (Iversen and Kautz, 2001, p. 299). This is connected to the key factor of a concern for measurement, where Dybå (2005) find that it is better to collect a few measures that are directly related to the process, rather than many measures that lead to a lack of focus and confusion about what is important. Translated to our situation, where we collect product metrics rather than process metrics, this means that we began by choosing a limited number of suitable measures for the product. These are the metrics for efficiency, effectiveness and satisfaction that are collected in the UTUM test. We started on a small scale in the year 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 Hellman and Rönkkö, 2008b for an overview). Thereafter the metrics evolved continuously over a period of time.

5.4. Area 4: 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. This is related to Dybå’s key factor of a concern for measurement, the extent to which the organisation collects and utilises quality data to guide and assess the effects of SPI activities (Dybå, 2005, p. 412). Emphasis is placed on using the metrics to improve the improvement process itself, the ability to measure aspects of the process, and make judgements of the process based on the measurements.

In our case, we have not measured the success of the metrics process itself. Given the focus that usability testing places on collecting metrics for the product rather than the process, this factor must be translated to our situation. Dybå found that it is necessary to feed the data back to the members of the organisation, and this has been performed in our case by presenting the results of usability testing to different stakeholders at many levels within the organisation, and by encouraging discussion of the results and their meaning. For more details regarding this, see below, regarding principles 7 and 8, on how the results were published and the debate that took place. In improving the usability testing process itself, improvements have not been based on measurement of the process, but have been made through the iterative research and development process that has taken place over a period of years.

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 (Iversen and Kautz, 2001, p. 300). This also has connections to Dybå’s factor of concern for measurement, where it is stated that the data must be perceived as accurate and valid, and must be used in a non-threatening way to identify and solve problems (Dybå, 2005).

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 results, based on the metrics for efficiency, effectiveness and satisfaction, to calculate a Usability 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 organisation. The user researcher also presented the metrics to two different groups of stakeholders: the Designers and Product Owners mentioned ear-

Please cite this article in press as: Winter, J., Rönkkö, K., SPI success factors within product usability evaluation. J. Syst. Software (2010), doi:10.1016/j.jss.2010.04.066
lier (Winter et al., 2008). For more information about these KPI’s, see Section 4 in this article, where we describe the UTUM test.

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 (Iversen and Kautz, 2001, p. 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.

5.5. Area 5: data usage

The data that is collected must be used to implement improvements or increase understanding about how the organisation 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 to use the metrics 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, recognising trends from imprecise data is still better than having no data at all (Iversen and Kautz, 2001, p. 302).

In relation to this principle we found for example 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 who 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.

In our case there were two main channels and target groups for implementing improvements, i.e. the previously introduced groups of stakeholders designated as Product owners and Designers. As previously defined, PO included management, product planning, and marketing, representing the management perspective, whilst Designers, were represented by e.g. interaction designers and system and interaction architects, representing the shop floor perspective. The distinction between these groups was connected with finding a balance between formality, and organisational needs from an agile, practical viewpoint of “day to day” work. Using the term formal in contrast to agile does not imply that we see agile processes as being informal or unstructured, but because the word “formal” is more representative than e.g. “plan driven” to characterise the results of testing and how these results are presented to certain stakeholders.

In relation to the first group, the 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.

The main findings detailed above are summarised in Table 3, showing in which areas our study confirms the findings of Iversen and Kautz, and where our material extends their findings with new

<table>
<thead>
<tr>
<th>Area</th>
<th>Confirmation or new knowledge</th>
<th>Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>C</td>
<td>We confirmed that it is important to use the knowledge within the organisation to improve and evaluate the program, to adapt to the environment and the situation at hand</td>
</tr>
<tr>
<td>Organisation</td>
<td>C</td>
<td>Our study confirms the importance of allocating resources, and in particular we found that it was an advantage that resources were dedicated to the program, minimizing the burden of &quot;extra&quot; work</td>
</tr>
<tr>
<td>Design</td>
<td>N</td>
<td>SPI needs clear goals from the beginning, but we also found the importance of adapting to a rapidly changing world by adopting a flexible approach. This means that there must be a readiness to change the goals during the ongoing process</td>
</tr>
<tr>
<td>Communication</td>
<td>N</td>
<td>It is not only important to base decisions on measurable phenomena – it is also important to include the knowledge that is based on intuition and experience, and that it was the test leader who was most suited to communicating this knowledge</td>
</tr>
<tr>
<td>Usage</td>
<td>N</td>
<td>We also found that it is important to have both agile and formal results, to satisfy the needs of different stakeholders. The formal results support the agile results and ensure that testing results are accountable in the organisation</td>
</tr>
</tbody>
</table>
knowledge. For each area, a confirmation is marked with a C in the second column, whilst new knowledge is marked with an N. This is discussed in more detail in the next section.

This concludes our comparison of the findings of Iversen and Kautz, the discussion of Dybä’s key factors, and the findings from our research. The comparison is made here to help the reader assess the applicability of the principles for companies attempting to implement software metrics programmes and usability programmes. We 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 organisations 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 presented above.

Iversen and Kautz found that an organisation 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 a 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. The focus on measuring the product rather than the process means that others within the organisation are not burdened by collecting information or through the use of the information. This is important 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 is said to need 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 illustrates another aspect of 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. There is a collision here, where those who deal with processes apply their lessons to the product, implying that improvements in the process automatically lead to improvements in the product. We find that improving the product requires a degree of flexibility, and suggest that this may be applicable within the field of SPI. We have found that it is important to follow and adjust to a rapidly changing world by adopting 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 results, for example in the form of direct feedback from the test leader to a designer or developer, 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, although it is connected to Dybä’s discussion regarding learning strategies and the exploitation or exploration of knowledge.

We 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. It is vital to find the right people, who can function as test leaders, analysts, communicators, and advocates of the user perspective. Within usability and user experience testing it becomes apparent that the test leader needs to possess traits other than those demanded in the engineering and technical areas. Here, the focus is directed towards understanding and discovering faults in software; it is instead directed towards understanding human beings, and how to best treat them to get them to reveal their attitudes and feelings towards and understanding of our products – things that they themselves may sometimes be unaware of – and even discover why they feel and act the way they do. We have for example found that some of the successful test leaders have a background in healthcare or the social services.

To summarise the concrete lessons for product focused usability work that can be learnt from the discussion above, which we suggest may also be applicable to SPI, we find that it is important to:

- ensure the presence and participation of specialists who are well versed in the structure and operations of the organisation,
- adopt a flexible approach to the design and performance of the programme, to adapt to changes in the environment,
- adapt the way results are presented in ways appropriate to different groups of stakeholders, to support the emphasis that different groups place on agile or formal results,
- find suitable people to work as key figures when collecting metrics, when analysing and communicating results, and who can act as advocates of the user perspective.

To conclude, this work 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 and is simultaneously in line with a focus on business orientation. This is a fact that emphasises the usefulness of usability testing, which is an area that has been neglected in the area of software engineering. As such, we can make a contribution to the field of software engineering.

However, there are still areas that need to be addressed, and one of these is the need to create a structured approach to user-centred design processes. Despite the similarities between usability testing and SPI, usability testing is not equivalent to SPI. In the same way as there is a need for SPI, there is also a need for usability process improvement. As previously mentioned in Section 3, it is important that international standards for usability should be more widely used, and that they should be used to improve usability capability.
within an organisation. The standards that exist provide a framework for how user-centred design should be practiced, and can have a serious impact on HCI and usability practices (Bevan, 2001). Earthy et al. (2001) claim that the implication of ISO 13407 and ISO 18529 is that there is a professional responsibility for software engineers, system engineers and usability professionals to adopt good practice as defined in ISO 13407 and ISO 18529 as their baseline. Taking the lessons learned from our study, and relating them to usability process improvement based on these international standards may lead to new insight in how to include these lessons in the detailed practices specified in the standards.

Finally, the study presented above is 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

The authors would like to thank the anonymous reviewers and the editor for their helpful advice and suggestions. This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the software development project ‘Blekning – Engineering Software Qualities’ (BESQ, 2008) and the Vinnova Foundation in Sweden under a research grant for WeBiS (WEBIS, 2008), dealing with method development, mixing innovation and usability.

References