

# An Approach to Knowledge Transfer in Software Measurement

Jari Soini

Tampere University of Technology, Pohjoisranta 11, P.O.Box 300, FI-28101 Pori, Finland

E-mail: jari.o.soini@tut.fi, www.pori.tut.fi

**Keywords:** software engineering, software measurement, knowledge transfer

**Received:** June 14, 2007

*Being successful in knowledge management is one of the most essential factors in any organization if one wishes to succeed in business in the future. This aspect is emphasized particularly in knowledge intensive organizations, such as software companies. Existing knowledge must be easily available for all and it must be possible to bring new information to the attention of everyone. This article introduces one example of a solution implemented to deal with this challenge. The case study described in this article deals with software engineering measurement and related knowledge collection, distribution and also utilization in practice. This example is one approach to trying to solve the challenges related to both the growing needs for effective knowledge transfer and for enhancing the utilization of measurement in software engineering.*

*Povzetek: Prispevek obravnava meritve programske opreme glede prenosa znanja.*

## 1 Introduction

In a networking economy, which is typical of today's information society, the ability of organizations and their members to cooperate, interact and share their information and knowledge is a prerequisite for strategic operation [1],[2],[3],[4]. Many studies support the view that collaboration and knowledge sharing between companies are effective means to enhance profit and growth [5],[6],[7],[8]. Nowadays, the success of an organization depends, maybe more than ever, on its ability to create and share knowledge effectively and efficiently [9]. There is a strong belief that the systematic transformation of human capital into value requires structural capital as a multiplier, to realize a sustainable earnings potential for the organization [10],[11],[12],[13]. This has forced companies to build a co-operative relationship with other organizations and try to increase their own learning and knowledge by utilizing the knowledge sharing which occurs during this interaction. However, successful networking and collaboration with other companies demands great openness, which is not easy for every company [14],[15]. This article gives us an empirical example of how to implement and manage this in practice.

The research topic of the case study relates to software engineering measurement. Measurement is one of the key elements in receiving feedback and evaluating the processes used in the organization. In software engineering, as knowledge of measurement theory and experience increase, measurement has begun to be perceived as an effective method of understanding, controlling, steering, predicting and improving software development and maintenance projects [16]. Fenton and Pfleeger [17] defined software measurement as the continuous process of defining, collecting, and analysing data on the software development process and its

products to understand, control and optimize the process and its products. This definition describes quite exhaustively what software measurement is all about. Nowadays software measurement is one of the key components in an organization's ability to maintain their competitiveness in a rapidly changing business environment [18]. There is a growing need for the use of objective information in decision making, and appropriate measurement enables all levels in software organizations to obtain this kind of objective information [18]. Organizations need a deeper understanding of their own processes and to do this, they need measurement data on their current processes. Although the advantages of measurement in the software development process are indisputable, the popularity of use of measurement methods in practice is rather limited [18],[19],[20]. Implementing measurement in software engineering raises many challenges. Very often difficulties arise when trying to focus the measurement. In many cases it is unclear what should be measured and also how the measurement data obtained should be interpreted [21],[22]. Choosing the correct measurement entities and ranking the importance of measurement indicators is a challenging task [23],[24],[25].

A research project was established in autumn 2005 with the aim of examining and trying to find potential procedures and techniques to solve problems related to product and process measurement in software engineering. This research was carried out in close collaboration between two universities and Finnish software organizations. One of the most essential sources of information in this work was the current measurement practices and metrics used in software organizations. Determining a method and developing an instrument for capturing, analyzing and sharing this measurement

knowledge between individuals and organizations was one of the central parts of the research. This article describes the instrument developed, *the information system*, for enhancing and transferring knowledge, related to software engineering measurement. The main contribution of this research was to give an example of how knowledge transfer in software engineering measurement can be organized and managed in practice.

## 2 Background of the study

### 2.1 Principles and goals of knowledge management

The starting point of the research was to understand the basis of both knowledge management and software engineering measurement. With regard to knowledge transfer we must first understand the basis of knowledge management. Figure 1 describes the theoretical framework of knowledge management [26]. As we can see, knowledge management (KM) refers to the activities involved in discovering, capturing, sharing, and applying knowledge. KM processes are meant to assist operations. Furthermore, processes are supported by KM systems, which are the integration of technologies and mechanisms. KM sub-processes (such as combination, socialization, externalization, internalization, exchange, direction, and routines) facilitate the broad processes and KM systems themselves rely on a current KM infrastructure.

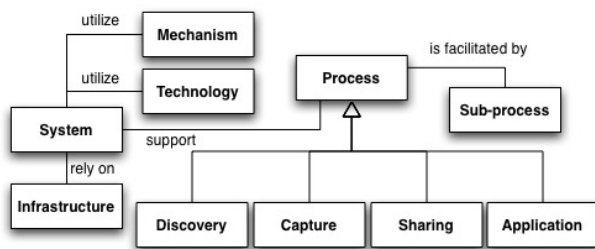


Figure 1: The concepts of knowledge management [26].

In general, knowledge management can be seen as a matter of improving conducive ways of thinking, practices and developing support systems to promote knowledge sharing. Success in knowledge transfer, which is one of the key issues in this study, is one prerequisite for creating new knowledge and organizational learning. The company must create a co-operative relationship with other organizations and try to increase their own learning and knowledge by utilizing the knowledge sharing and transfer which occurs during this intercourse [27]. The overall aim is to reduce the uncertainty of the operational environment by ensuring that the company has the opportunity to access wider knowledge of the business environment. Knowledge sharing, which is a consequence of this kind of collaboration, generates additional value for every organization, in this case by increasing its knowledge capital, thereby improving its competitiveness.

### 2.2 A case study – the SoMe project

The motivation for the research derived from issues observed in relation to the software process and product quality, at national level [28] as well as international level [29],[30],[31]. The widely accepted assumption is that software quality, in general, is caused by and dependent on the quality of the software development process. The accepted opinion is that most problems in software quality are based precisely on problems in the software development process [32],[33],[34]. In practice, it has proved difficult to define the key functional process and product measurements and many software companies have found measurement to be challenging and problematic [35],[36]. To promote a better understanding of measurement and to offer a robust and pre-selected set of metrics suitable for different kinds of business goals, FiSMA (the Finnish Software Measurement Association) [37], initiated the SoMe (Software Measurement) project in autumn 2005 together with Tampere University of Technology (TUT) [38] and the University of Joensuu (UJ) [39]. FiSMA itself is a non-profit making organization created to promote the usage and utilization of software measurement to improve the quality of processes and products. Its members, who are also the participants of this study, consist of nearly 40 Finnish software companies, plus several universities and other public organizations. In the context of the SoMe project, different instruments and practices were studied to help solve the measurement problems related to the quality of both software process development and software products. The SoMe project focused especially on experience-based measurement data. Therefore, the sample was a set of software companies (a total of 10) who perform process measurement in practice. Nine of the participant companies can be classified as small and medium size companies (SMEs) [40], from the viewpoint of organizational units involved in software engineering (table 1). The common characteristics, shared by these companies, are that their core business is the supply of software projects and they carry out software development independently.

Company	A	B	C	D	E	F	G	H	I	J
Empl., total	195	200	200	220	280	450	1,200	3,200	15,000	24,000
Empl. In SE	195	30	200	200	150	35	30	120	5000	200

Table 1: Size of participating companies by the number of employees

The main idea and the ultimate objective of the SoMe project was to develop, in cooperation with the participants, a common and open information system for Finnish software companies to help monitor and measure the quality of their software processes and products. The final outcome of the SoMe project was an information system implemented in a web environment based on a large metrics database. The final database consists of three different types of measurement information (practical experience, literature and standards). The

developed information system utilizes a web-based repository of best practices.

### 3 Developing an instrument for transferring software measurement information

#### 3.1 Method used for capturing, organizing and evaluating the collected knowledge

One of the main targets of the project was to get as much experience-based information in the database as possible. Therefore, the empirical part of the research was based on a series of interviews and questionnaires, created to collect the experiences of the companies about individual metrics and measurement practices in general. The research method used was to conduct interviews to address the research questions and the target group was quality managers. Information on the metrics used and current measurement practices were collected on a spreadsheet-style form, providing the basic data for this study (see Appendix A). The aim was to give as explicit a description as possible of all the process metrics used in the participating companies. The information on the current metrics (Appendix A) was collected before the interviews, because it provided the possibility to expand on it during the interview session. Figure 2 describes the process of how the knowledge was captured, modified, evaluated and shared.

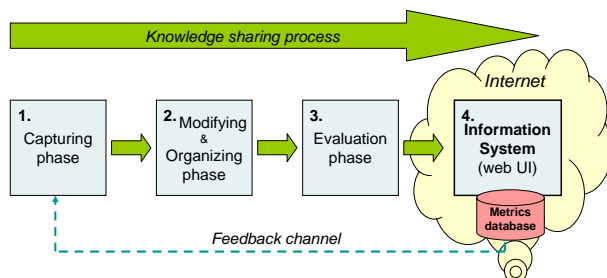


Figure 2: The phases for capturing and sharing knowledge [41].

After the capturing phase, the collected information was combined and organized by the researchers. The information captured was pre-evaluated and analyzed as to its suitability, usability and the correctness of the examined topic. The captured knowledge was also modified in the same framework. The metrics database consists of individual items of information, *knowledge items*, and the manifestations of these items are *metric documents*. The formula for the title level and the terminology used in all documents is congruent. This solution helps the end user to read, perceive the logic and make comparisons between knowledge items.

Before distributing the captured and modified knowledge via the information system, the applicability and intelligibility of the information together was evaluated with representatives of the participating companies (see ref. [41]). A support group was

established inside FiSMA. The aim of this practice was to evaluate the knowledge collected with the end users before placing it in the information system and delivering it to the organizations. All the companies involved were able to take part in regular support group meetings. Other publications in relation to the SoMe project [41],[42] describe in more detail the process used for capturing, modifying, evaluating and distributing the measurement knowledge via the information system that was developed.

#### 3.2 Overview of the information system

The final outcome of the SoMe project was a measurement knowledge base consisting of a large metrics database. The expression of this final outcome is a web-based information system for measurement knowledge transition, which was successfully implemented in April 2007. This information system is meant for individuals and organizations seeking appropriate software metrics and measurement practices for their needs. It offers a bi-directional link, from processes to metrics, and also vice versa. With the metrics offered, the information system enables organizations to utilize measurement for controlling and monitoring their software processes and products and thus enhances the quality of the software produced.

##### a. Ideology of the system

The aim of the SoMe project was to develop an information system which will enable organizations to control and improve their software development process and product quality. In relation to process quality, process assessment models like the ISO/IEC 15504 process assessment model (SPICE) [43] and Capability Maturity Model Integrated (CMMI) [44] are available, which allow evaluation of the quality of the current software development process in the organization. For process improvement work, organizations need a deeper understanding of their own processes and to do this they need measurement data [16]. With this measurement information they can reliably seek and find improvement objects in their processes [45],[46],[47]. This approach was selected as the starting point of the development work and steered the work throughout system implementation. This aspect also guides the search taxonomy design of the information system. The selected search taxonomy was created based on the CMMI and SPICE process assessment models (see Figure 3). The aim of this selection was to for the organizations to familiarize themselves with and utilize these assessment models in their operations. This aspect is crucial for the process improvement viewpoint. Assessing the actual state of the current processes, which is also the first step of process improvement actions, is one of the most typical purposes of the use of measurement in software engineering [23]. The main ideas for developing the information system were precisely to help organizations utilize measurement knowledge to control their software development process and also to use measurement for

supporting their process improvement work as well as being the source of objective information for management in their decision making.

### b. Metrics database

The system itself works on a database, which contains information about software measurement literature, standards and actual metrics and measurement practices used in software organizations. The metrics database consists of individual items of information, *knowledge items* (individual metrics). A standard form, a *metric document*, is used for presenting each knowledge item. The formula for the title level and the terminology used in all metric documents is congruent with the others. This solution helps the end user to read, perceive the logic and make a comparison between the metrics. All the process and product metrics in the database, carefully analyzed relevant metrics collected from the participants (85 metrics) as well as additional relevant metrics found in literature or standards (22 metrics), have been modified using the same standard form.

The following knowledge exists on each individual metric captured in the system: Purpose, Formula (if required), Values (with a possible threshold value), Usage, Workloads (establishing the metric, collecting the data, using the metric), Risks, Experiences plus other information and References. All this information was collected with the spreadsheet-style form (Appendix A). If required, in experience-based metrics, this information was supplemented and clarified during the personal interview sessions. In addition to this, in the standard form, the metric links to the search taxonomy can be seen (the processes that the particular metric relates to are shown). As an example, there follows a description of one metric; *Distribution of customer work*, and its information under the *Usage* title: “It is a derived metric used mainly by upper management to monitor how the maintenance work is distributed between corrective, adaptive and perfective work. All maintenance work must be classified (at least) into these categories and recorded in the time tracking system. To calculate a ratio, the formula is: work hours in one category divided by the sum of all maintenance work hours. To use the metric, a detailed time tracking system must be in use, where each employee records his/her working hours. Once a month the quality manager collects the totals of customer work hours and presents them as a table or a graph. Inserting data in the time tracking system requires manual work, but after that the results can be calculated automatically”.

Appendix B describes the experience-based measurement objects collected by the participating companies. The ISO/IEC 15504 (SPICE) standard was used as a framework to classify the metrics into suitable categories named after the SPICE processes. An evaluation is also presented of the given characteristics (see Appendix A) in relation to the metrics used. A more detailed analysis of the results of these user evaluations is presented in a previous research paper [48]. The experience-based metrics are collected from companies whose capability levels varied between 2 and

3. The experience-based metrics inside the database can be utilized mostly at a SPICE capability level of 2 (56 %), at level 3 (35 %), and 9 % at level 4 (see ref. [41]).

### c. Web-based information system

In the information system, the knowledge items (individual metrics) are linked to the process groups inside the assessment models. Every knowledge item also includes information for all the process groups to which it relates and in practice is linked. This characteristic enables the user to see the dependence between process groups from a metric viewpoint and gives important information when planning measurement activities (e.g. a measurement program). These connections are also seen from the process group viewpoint, as the proper metrics depend on the selected process group in the selected assessment model (SPICE or CMMI). This realization method enhances awareness of the relationship between process assessment and process measurement. As an example, Figure 3 presents the results of one search; SPICE assessment model / (Engineering process group) ENG 10 System testing. After making the selection, the results (individual metrics) of this search appear in the information system display (see right of the figure). The user can see directly all the related metrics and also a brief description such as: the name of the metric(s), a short summary of each metric and workload evaluation for establishing, collecting and using the metric. Depending on the given search selection, the system retrieves the particular metrics from the database that are linked to the selection (individual process group inside the assessment model). Selecting a particular metric (by clicking on the metric name field) calls up the detailed information of this metric (see example in section *b. Metric database*). Every user can examine and compare which metric or metrics are appropriate and also compatible with the needs and maturity of their organization. It is possible to make a new search with a new search criterion as many times as you want. Organizations decide and select an appropriate set of metrics according to their needs independently.

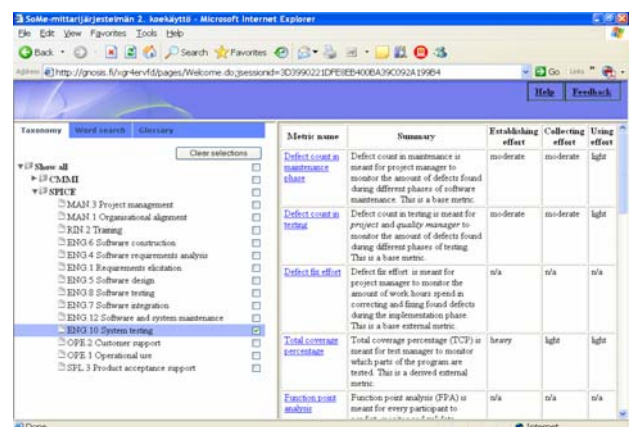


Figure 3: User interface (UI) of the web-based information system

The information system includes also a *word search* alternative (see Figure 3) for searching for a suitable metric. This feature was included because there may be organizations that are not familiar enough with the process assessment models to start using a system based on them. The starting point for planning the system web-user interface (UI) was that it must display as well as operate in such a simple way that the UI does not become an obstacle to the use of the system. A clearly and simply defined UI, both structurally and visually, is one of the most important factors when introducing new technical tools [49]. In this work, a lot of co-operation was made with the participant organizations and the people there were assumed to be the end users of the information system. This method is perceived as a functional approach when developing new applications for a certain target group [50].

Certain interactive activities are also involved in the information system. There are several reasons for this. Firstly, as stated above, the environment in software engineering is changing rapidly [51]. Therefore there could be a need for adding, modifying or even deleting some information related to the existing knowledge item (individual metric) in the system. Also, during the operation of the system, new experiences may arise and this interactivity allows new information to be added and also combined with the current item. This feature creates a line of communication between individuals and organizations, allowing them to share knowledge and learn from each other. Additionally, the information system includes a library. A glossary (see Figure 3) helps the user if the terms or concepts used in relation to measurement and metrics are not familiar. The terms and concepts used are mostly based on the terms and definitions used in software standards [52],[53],[54],[55]. This selection will guide the organization towards harmonized use of the terms related to software measurement.

## 4 Observations and evaluation of the developed system

### 4.1 General observations

The results of the research worth considering and evaluating are the research method used and the information system that was developed. The analysis and discussion presented here highlight some general observations on the research. The method used proved very useful for collecting and also evaluating information. The attitude and motivation of the participants to share their knowledge seemed very positive and the empirical data received met the requirements of the research. The method of data collection (a data form combined with interviews) and evaluating (with the support group) the data adds to its quality (see ref. [41]). The starting point and aim was to obtain more qualitative than quantitative data. The selected method supported the set goal very well. Overall, the method used during the research, as well as

the created web-based information system for executing knowledge management and sharing, seem to be workable and seem to have achieved the goals which were set. The decision to connect the participants in the study from the very outset and the close co-operation with them throughout the project proved to be a good choice. This enhanced the knowledge-sharing process itself and also ensured the appropriateness of the final outcome. This method provides important information and feedback also for researchers on how to develop the system correctly and also maintained motivation for the researchers during a long project.

### 4.2 Evaluation of the information system

Below are evaluated briefly the main strengths and also some deficiencies of the information system.

Firstly, the strengths: It contains detailed information, mostly experienced-based, on the metrics involved in the system. With this information, organizations can initiate, or confirm and improve their existing measurement system. This kind of information seems to interest organizations and is very relevant especially if the organization wants or is intending to establish a measurement system. The most significant contribution from the technical point of view is a bidirectional link between processes and metrics. This feature helps and advises the users to identify the relevant metrics for controlling the particular process. The system guide advises the user on how to use measurement as an instrument for software process improvement. The user has the opportunity to see the connection and the dependence between software metrics and the software process groups. Becoming aware of these correlations is very important when planning, implementing or controlling the software processes or their improvement. In comparison with the other existing related tools, such as the knowledge PLAN and FP workbench by Software Productivity Research (SPR) [57] or the International Software Benchmarking Standards Group (ISBSG) database [58], the advantages of the new system are a closely specified description of the metrics themselves, a guiding feature for searching for the proper metric(s) and also an emphasis on software engineering measurement. With regard to the SPR (including information on over 2000 projects) or ISBSG (over 8000 projects), one disadvantage is the amount of purely numerical data. These systems are mainly intended for effort estimation and scheduling and for performing benchmarking. The SPR and ISBSG systems are more focused on a project viewpoint, rather than the process and product. One obvious strength of the new information system is also the congruent terminology of all the documents. This helps and advises the end user on how to read, perceive the logic and make a comparison between the knowledge items.

Secondly, however there are some deficiencies: the system is only to serve the issues related to the software process and product measurement; other subjects are excluded. This factor limits the utilization of the system. Also, the heavy emphasis on the software process

assessment model which has been selected may be a factor of uncertainty. This approach could be unfamiliar to some of the users. Related to the users' evaluation of the requested characteristics of the metric used, it must be noted that the answers are subjective. This may cause a bias in the evaluation results if they are regarded purely from a scientific point of view [59],[60]. It is also noteworthy that the experience data has been collected in organizations whose capability levels vary between 2 and 3. This fact must be taken into account when planning to utilize these experienced-based metrics.

Finally, some suggestions for future work. It may be useful to add some alternatives for the current search taxonomy (SPICE and CMMI) into the information system. For example, factors related to the organization, such as size and capability level, could be pre-selection factors when starting to seek the proper metrics. More feedback is needed when decisions are made relating to the improvement actions of the system. From a software measurement viewpoint, one relevant topic for future research could be to examine how the capability or maturity level of the organization may affect the utilization of the information system and how, and if, the required measurement information varies depending on the different levels. From a knowledge transition viewpoint, and also to validate the method developed and the tool itself, it could be interesting to implement this method for some other research subject in software engineering.

## 5 Summary

This article deals with the research based on issues observed in relation to software process quality. In order to examine the solutions, the research project (SoMe) was established, with the aim of studying how measurement can be utilized for solving this issue. The goal was to collect the relevant and experience-based measurement data for this purpose, and also to create an appropriate information system for delivering the received knowledge to the software organizations. This article describes the method used and gives an overview of the ideology and the structure of the developed web-based information system as well as the measurement data included it. The generated system offers information about different metrics and their applicability to measuring different processes. It also gives an example of how issues related to information can be collected and one approach to solving its transition.

## References

- [1] P. Drucker (1999). Knowledge-Worker Productivity. The Biggest Challenge. *California Management Review*, Vol. 41, no. 2, pp. 79-94.
- [2] J. C. Jarillo and H. Stevenson (1991). Co-operative strategies – The payoff and the pitfalls. *Long Range Planning*, Vol. 24, pp. 64-70.
- [3] M. Iivonen and M. L. Huotari (2004). *Trust in knowledge management and systems in organizations*. Idea Group Publishing, London.
- [4] P. Sydänmaanlakka (2001). *Älykäs organisaatio. Tiedon, osaamisen ja suorituksen johtaminen*. Kauppakaari, Helsinki.
- [5] H. Hinterhuber and B. M. Levin (1994). Strategic networks – The organization of the future. *Long Range Planning*, Vol. 27, pp. 43-53.
- [6] C. Snow, R. Miles and H. Coleman (1992). Managing 21<sup>st</sup> century network organizations. *Organizational Dynamics*, Vol. 20, no. 3, pp. 5-20.
- [7] J. S. Whipple, J. J. Gentry (2000). A network comparison of alliance motives and achievements. *Journal of business and industrial marketing*, Vol. 15, no. 5, pp. 301-322.
- [8] I. Kulmala, A. Vahteristo and E. Uusi-Rauva (2002). *Ohjelmistoyritysten verkostoituminen – tutkimus suomalaisten ohjelmistoyritysten toimintatavoista ja tavoitteista*. Research Report 3/2002. Tampere University of Technology, Tampere, Finland.
- [9] T. H. Davenport and L. Prusak (1998). *Working knowledge: How organizations manage what they know*. Harvard Business School Press, Boston, MA.
- [10] L. Edvinsson, B. Kitts and T. Beding (2000). The next generation of IC measurement: The digital IC landscape. *Journal of Intellectual Capital*, Vol. 1, no. 3, pp. 263-272.
- [11] M. R. Testa (2002). A Model for organization-based 360 degree leadership assessment. *Leadership & Organization Development Journal*, Vol. 23, no. 5, pp. 260-268.
- [12] U. S. Bititci, V. Martinez, P. Albores and J. Parung (2004). Creating and managing value in collaborative networks. *International Journal of Physical Distribution and Logistics Management*, Vol. 40, no. 3, pp. 251-268.
- [13] U. S. Bititci, V. Martinez, P. Albores and J. Parung (2004). Creating and managing value in collaborative networks. *International Journal of Physical Distribution and Logistics Management*, Vol. 40, no. 3, pp. 251-268.
- [14] W. Seal, J. Cullen, A. Dunlop, T. Berry and M. Ahmed (1999). Enacting a European Supply Chain: A Case Study on the Role of Management Accounting. *Management Accounting Research*, Vol. 10, no. 3, pp. 303-322.
- [15] M. Beeby and C. Booth (2000). Networks and inter-organizational learning: a critical review. *The Learning Organization*, Vol. 2, pp. 75-88.
- [16] L. Briand, C. Differding and D. Rombach (1997). Practical Guidelines for Measurement-Based Process Improvement. *Software Process – Improvement and Practice*, Vol. 2, no. 4, pp. 253-280.
- [17] N. E. Fenton and S. L. Pfleeger (1997). *Software Metrics: A Rigorous & Practical Approach*, 2. Edition. PWS Publishing Company, Boston, MA.
- [18] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean and F. Hall (2002). *Practical*

- Software Measurement. Objective Information for Decision Makers.* Addison-Wesley, New York.
- [19] T. Varkoi (1999). Development of Measurement Programs to Support Process Improvement in Small Software Companies. *Proceedings of the FESMA '99 Conference.* Amsterdam, Netherlands, pp. 141-149.
- [20] Software Engineering Institute (SEI) (2006). *The State of Software Measurement Practice: Results of 2006 Survey.* Technical Report, CMU/SEI-2006-TR-009, ESC-TR-2006-009.
- [21] N. Habra, A. Abran and M. Lopez (2004). *Towards a framework for Measurement Lifecycle.* University of Namur, Technical Report, TR37/04.
- [22] P. Kulik (2000). Software Metrics: State of the Art. Retrieved 10/18/05 World Wide Web, <http://www.klci.com>
- [23] A. Neely (1998). *Measuring Business Performance. Why, What and How?.* Profile Books Ltd, London.
- [24] Software Productivity Center Inc. *SPC Metrics Resources, 8-step Metrics Program.* Retrieved 8/18/07 World Wide Web, <http://www.spc.ca/resources/metrics/index.html>
- [25] R. van Solingen and E. Berghout (1999). *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development.* McGraw-Hill Publishing Company, London, UK.
- [26] I. Becerra-Fernandez, A. Gonzalez and R. Sabherwal (2004). *Knowledge Management. Challenges, Solutions, and Technologies.* Prentice Hall, Upper Saddle River, NJ.
- [27] M. R. Testa (2002). A Model for organization-based 360 degree leadership assessment. *Leadership & Organization Development Journal*, Vol. 23, no. 5, pp. 260-268.
- [28] H. Harju and M. Koskela (2003). *Cost-effective reliability design and assessment of software. Part 2.* VTT – Research Notes 2193, Espoo, Finland.
- [29] S. H. Kan (2002). *Metrics and Models in Software Quality Engineering, Second Edition.* Addison-Wesley Professional, Boston, MA.
- [30] S. H. Kan, V. R. Basili and L. N. Shapiro (1994). An overview from the perspective of total quality management. *IBM System Journal*, Vol. 33, no. 1, pp. 4-19.
- [31] J. S. Osmundson, J. B. Michael, M. J. Machniak and M. A. Grossman (2003). Quality management metrics for software development. *Information & Management*, Vol. 40, pp. 799-812.
- [32] T. Conti (1993). *Building Total Quality: A guide for management.* Chapman & Hall, New York
- [33] S. Zahran (1998). *Software Process Improvement: Practical Guidelines for Business Success.* Addison-Wesley Longman Ltd, Essex, UK.
- [34] W. S. Humphrey (1989). *Managing the Software Process.* Addison-Wesley Longman Publishing Co, Boston, MA.
- [35] N. Fenton, “Software Measurement: A Necessary Scientific Basis”, *IEEE Transactions on Software Engineering*, Vol. 20, Issue 3, March 1994, pp. 199-206.
- [36] N. Fenton, and N. Martin, “Software metrics: successes, failures and new directions”, *Journal of Systems and Software*, Vol. 47, Issues 2-3, July 1999, pp. 149-157.
- [37] FiSMA, Finnish Software Measurement Association. Retrieved 10/16/07 World Wide Web, <http://www.fisma.fi/eng/index.htm>
- [38] TUT, Tampere University of Technology, Retrieved 9/22/07 World Wide Web, <http://www.tut.fi/public>
- [39] UJ, University of Joensuu, Retrieved 9/22/07 World Wide Web, <http://www.joensuu.fi/englishindex.html>
- [40] SME; Small and Medium size enterprise. Retrieved 5/23/07 World Wide Web, [http://ec.europa.eu/enterprise/enterprise\\_policy/sme\\_definition/index\\_en.htm](http://ec.europa.eu/enterprise/enterprise_policy/sme_definition/index_en.htm)
- [41] J. Soini, V. Tenhunen and T. Mäkinen (2007). Managing and Processing Knowledge Transfer between Software Organizations: A Case Study. *Proceedings of the PICMET'07 Conference*, Portland, USA. pp. 1108-1113.
- [42] J. Soini, T. Varkoi, V. Tenhunen, M. Tukiainen (2007). Empirical case study of software metrics using the SPICE framework. *Proceedings of the SPICE'07 Conference*, Seoul, Korea. pp. 22-28.
- [43] ISO/IEC IS 15504-5:2006 (2006) Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model (SPICE).
- [44] CMMI for Systems Engineering/Software Engineering, Version 1.02. CMU/SEI-2000-TR-029. Carnegie Mellon University, SEI.
- [45] E. Rodenbach, F. van Latum and R. van Solingen (2000). SPI – A Guarantee for Success? – A Reality Story from Industry. *Proceedings of the PROFES 2000 Conference.* Eds. F. Bomarius and M. Oivo, Oulu, Finland, pp. 216-231.
- [46] T. Dybå (2000). An Instrument for Measuring the Key Factors of Success in Software Process Improvement. *Empirical Software Engineering*, Vol. 5, no. 4. pp. 357-390.
- [47] S. L. Pfleeger and H. D. Rombach (1994). Measurement Based Process Improvement. *IEEE Software*, July, pp. 9-11.
- [48] J. Soini, V. Tenhunen and M. Tukiainen (2007). Software Metrics and Evaluation of Their Usefulness in Finnish Software Companies. *Proceedings of the IWSM Mensura 2007 Conference*, Palma de Mallorca, Spain. pp. 204-215.
- [49] S. Badker (1991). *Through the Interface. A Human Activity Approach to User Interface Design.* Lawrence Erlbaum Associates. Inc., New Jersey.
- [50] M. Huysman and D. de Wit (2002). *Knowledge sharing in practice.* Kluwer Academic Publishers, Dordrecht.
- [51] R. Petty and J. Guthrie (2000). Intellectual Capital literature review – measurement reporting and management. *Journal of Intellectual Capital.* Vol. 1, no. 2, pp. 155-176.
- [52] ISO 9000-3 (1997). *Guidelines for the application of ISO 9001:1994 to the development, supply,*

- installation and maintenance of computer software.*  
International Organization for Standardization.
- [53] ISO/IEC 9126-3:2003 (2003). *Software engineering – Product quality – Part 2: External metrics.* Technical Report.
- [54] ISO/IEC 9126-3:2003 (2003). *Software engineering – Product quality – Part 2: Internal metrics.* Technical Report.
- [55] IEEE Standard 1061-1998 (1998). *IEEE Standard for Software Quality and Metrics Methodology.* IEEE-SA Standard Board.
- [56] Software Productivity Research (SPR). Retrieved 11/12/07 World Wide Web, <http://www.spr.com/products/default.shtm>
- [57] International Software Benchmarking Standards Group (ISBSG). Retrieved 11/12/07 World Wide Web, <http://www.isbsg.org/isbsg.nsf/weben/Tools>
- [58] R. M. Cooke (1991). *Experts in Uncertainty: Opinion and Subjective Probability in Science.* Oxford University Press, NY, Oxford.
- [59] M. Bertrand and S. Mullainathan (2001). Do People Mean What They Say? Implications for Subjective Survey Data. *The American Economic Review*, Vol. 91, no. 2, pp.67-72.



## Appendix A

Spreadsheet-style interview form for collecting detailed metrics information [42].

	A	B	C	D	E	F	G	H
1	#	Name	Purpose	Type	Target	Application Domain	Formula	Values
2		<i>name of the meter</i>	<i>extensive description of the meter and its purpose</i>	<i>meter's type: Monitoring, Controlling, Predicting, Validating</i>	<i>what attributes of product, process and/or project are measured</i>	<i>in what processes, lifecycle phases etc. the meter is used</i>	<i>how the meter is calculated</i>	<i>values the meter produces and their interpretation: what's good, what's bad, what's preferred</i>
	I	J	K	L	M	N	O	P
1	Data 1	Data 2	Data 3	Data 4	Data Collection Rate	Primary Collectors	Secondary Collectors	Usage
2	<i>data used to calculate the meter</i>	<i>data used to calculate the meter</i>	<i>data used to calculate the meter</i>	<i>data used to calculate the meter (add more columns if needed)</i>	<i>how often the data is collected (e.g. X times in a day/week/month/other, please specify)</i>	<i>who are primarily responsible for measuring or for collecting the data</i>	<i>who are secondarily responsible for measuring or for collecting the data</i>	<i>how the data is collected and the metric calculated</i>
	Q	R	S	T	U			
1	Examination Rate	Primary Beneficiaries	Secondary Beneficiaries	Workload in Establishing the Meter (1)	Workload in Establishing the Meter (2)			
2	<i>how often are the meter's results looked at (e.g. X times in a day/week/month/other, please specify)</i>	<i>who are primarily using the meter's results</i>	<i>who are secondarily using the meter's results</i>	<i>how much resources are consumed when the meter is first introduced and established (e.g. person-hours, calendar time etc.)</i>	<i>estimation of the meter workload; scale: 1 = heavy, 2 = considerable, 3 = moderate, 4 = light</i>			
	V	W	X	Y	Z	AA		
1	Workload in Using the Meter (1)	Workload in Using the Meter (2)	Accuracy	Reliability	Risks	Usefulness		
2	<i>how much resources are consumed when the meter is used (e.g. person-hours, calendar time etc.)</i>	<i>estimation of the workload; scale: 1 = heavy, 2 = considerable, 3 = moderate, 4 = light</i>	<i>estimated accuracy of the meter's results; scale: 1 = inaccurate, 2 = approximate, 3 = quite accurate, 4 = very accurate</i>	<i>estimated reliability and robustness of the meter's results; scale: 1 = unreliable, 2 = moderate, 3 = quite reliable, 4 = very reliable</i>	<i>risks and problems related to the meter's usage</i>	<i>general estimation of the meter usefulness; scale: 1 = useless, 2 = of limited use, 3 = quite useful, 4 = very useful</i>		
	AB	AC	AD	AE	AF			
1	Other Information	Source	References	Author	Web Links			
2	<i>free-form notes, comments and other information (e.g. meter's relation to various process models; to different sizes and types of projects or organizations; etc.)</i>	<i>source of the meter, any of the following: Literature, Organizations, Standards</i>	<i>detailed sources of the meter: - name of the author and book/article/web site/etc. - name of the company or organization - name of the standard or model</i>	<i>name of the author who has written the information about the meter in this table</i>	<i>possible links to WWW sites with related information</i>			

## 6 Appendix B

A summary of current measurement objects (for the SPICE framework viewpoint) with user evaluation based on five separate categories (effort requirements for establishing, effort requirements for use, reliability, accuracy and usefulness) [48]. The four -point scale for evaluating each category is given in Appendix A.

