UITQ
2005

UITQ 2005

User-driven IT Design and Quality Assurance

Pre-Proceeding print-out of workshop papers

CID

CID

Centre for User Oriented IT Design

TABLE OF CONTENTS

Welcome to the UITQ 2005 Workshop!

FOREWORD

This year's theme takes the form of a question "Who can judge IT Quality better than professional end-users?" Here we will give you our motivation for the question and an overview of how all the presenters of this year's workshop have approached it.

Eight years of UsersAward network activities have shown that professional end-users, when given the chance to "judge IT quality", select and promote remarkably innovative and sustainable IT projects. The diversity and creativity of all the thirty Users' IT Prize Finalists between 2000 and 2005 give ample proof of this. We think this is an excellent collection of ongoing, influential, "good examples" of effective, efficient, and pleasing IT software.

But the question of "who can judge" is also directed against the mainstream of IT quality interpretations that tend to disregard the judgement of end-users when discussing IT quality issues and formulating policies for technical and social research and development. Therefore, the often tacit professional knowledge and work habits of those who are to use the new IT tools needs to be made visible through more exact *and* more expressive validation instruments. With the help of such instruments, the possible gains from end-user participation in IT design can become tangible for industry, services, customers, and employees.

How can the workshop help to make the potential gains of user and customer influence tangible and clear? Here is an overview of the urgent research questions that the presenters bring to the table:

Claes Fornell shows how consumers, taken as an aggregate and given the chance to judge the quality and price of products and services through the American Customer Satisfaction Index (ACSI), can predict stock value growth in a way that, in the long run, outperforms the three US stock value indexes.

Can the role of IT user satisfaction – as a traceble part of customer satisfaction – be demonstrated in the same way through a systematic judgement by end-user?

Jochen Prümper, Thomas Vöhringer-Kuhnt & Jörn Hurtienne have made the first test of the validity and reliability of an international version of the User Certified 2002 instrument through a comparison to the German ISONORM instrument.

Can an international trade-union and university network for certification of workplace software for specified usage become a complement to ISO standards and other quality measures?

Timo Jokela presents and discusses the feasibility of certifying software supplier organisations through the use of Usability Maturity Models (UMM).

Can the certification of software suppliers and deployment consultants become a complementary instrument for a quality assurance network to influence specified workplace IT usage?

Peter Brödner concludes in his paper that the UsersAward procedures adopt "an adequate approach to (...) intensify communication between suppliers and users".

Can the UsersAward procedures complement instruments such as the ACSI, ISONORM, and UMM in visualizing possible social benefits from IT usage within enterprises and organisations?

Takehiro Eto reports on the development of a broad commitment in Japanese industry to better match business and IT strategy, understand end-users' needs and provide good training.

Can an international research exchange focused on the needs of professional end-users increase the impact of national research groups?

Alan Borning demonstrates how the concept of Value Sensitive Design (VSD), i.e. clearly articulating the value considerations of all involved stakeholders, can make value judgements more understandable and traceable in IT projects with broad public scope.

Can Value Sensitive Design and similar methods for establishing "track-records of user involvement" become one of the key contributions from universities to a quality assurance network?

Karl Heinz Rödiger presents interesting results from introducing problem oriented hands-on education for systems engineers.

Can the carrying out of quality assessments in real workplaces become an obligatory elements in future university courses for systems engineers?

Bengt Sandblad discusses the consequences of bad design and bad use of IT in workplaces in terms of health costs.

Can local unions and health care units cooperately carry out yearly screenings of health aspects of IT tools, as an ongoing follow-up of health-related IT research carried out within the international quality network exchange?

Lars Taxén presents results from an IT design project at Ericsson in which the main target of design was not an information system but the entire work practice in which an information system was developed. This meant that all actors performing coordination acts in their work became contributors to the IT design.

Can knowledge from designing the IT intensive work environments of software and communications companies be transferred to the less IT intensive branches of industry and services, with local unions and healthcare units serving as a networking bridge?

Clas Thorén from the Swedish Agency for Public Management discusses the need for clear, well-defined, understandable criteria for accessibility and usability of various products and services. He argues that suppliers' declarations offer a possible way of introducing usability requirements in calls-for-tender.

Can the national public procurement agencies, and the public purchasers they represent, become key users of future certification instruments, as they have been discussed in the earlier questions?

Magnus Lif, Bengt Göransson & Torsten Sandbäck reports work with writing a guide for procuring according to a User-Centered Procurement Process (UCPP), an assignment from the Swedish Agency for Public Management. The authors hope the guide will become a tool for public procurers to enhance the usability work of their own organisations.

Can these kinds of guides, based on IT quality research, become yet another tool for the union-health care network envisioned in the earlier questions?

Stefanie Floegel, Reinhard Linz & Jochen Prümper show how ergonomic customizing as an integrated measure in deployment methods can enhance the effectiveness, efficiency and satisfaction of existing standard ERP software.

Can this kind of integrated ergonomic customization measures work as a model for the yearly screenings done by local unions and health care units?

Christian Koch discusses development of ERP and deployment methods and their impact on work routines and user roles, specifically the opportunities for enterprise implemention actors to make room for local requests and adaptations of standardized ERP software during a prolonged deployment phase.

Can validated methods for deployment and follow-up, through e.g. certifications, be another tool for local unions and health care units to make an impact with their yearly screenings?

With these questions, some of which we hope to bring up in the final summary session, we warmly welcome you to a productive and enjoyable workshop in Stockholm!

Yngve Sundblad Åke Walldius, Co-chairs of UITQ 2005

UsersAward – first Results of a Pilot Study in Germany

Jochen Prümper
FHTW – Berlin
University of Applied Sciences
Treskowallee 8
D-10313 Berlin
j.pruemper@fhtw-berlin.de

Thomas Vöhringer-Kuhnt & Jörn Hurtienne
bao GmbH
Büro für Arbeits- und Organisationspsychologie
Kösterstraße 1b
D-14165 Berlin
t.kuhnt@bao.de, j.hurtienne@bao.de

ABSTRACT
Within the framework of a project on the transfer of established methodologies for measuring and enhancing user satisfaction with workplace software from Sweden to other European countries, the Swedish UsersAward questionnaire has been adapted to German demands. Together with the questionnaire "ISONORM 9241/10", it was used as part of a pilot study examining customer satisfaction with eight different software products. 90 participants from eight different companies took part in the survey. The data has been analysed to gain detailed insight into the usefulness and quality of the translated instrument with respect to the German context of software application and certification. In addition to the outcome of the software certification process itself, the results of a reliability and validity analysis of the whole instrument and the proposed subscales are reportted. Cronbach's Alpha and correlations with the ISONORM 9241/10 are calculated. Recommendations for revisions of upcoming versions of the UsersAward questionnaire are specified.

Author Keywords
UsersAward, ISO 9241-10, user certification, software quality

INTRODUCTION
Collaboration between users, software developers and scientists has a tremendous impact on the quality of the design of interactive software systems. Especially with regard to satisfaction with a software product, the participation of end users during the development and implementation of interactive systems is essential. A quality assurance process for software systems was established in Sweden by UsersAward in 1998, according to the certification of displays and other business products by TCO (Tjänstemännens Central-Organisation).

The project was launched by the Swedish Trade Union Confederation (LO) in cooperation with the Swedish Agency for Innovation Systems (VINNOVA) and four universities [4]. Within this network, union and consumer organizations, researchers, user companies and software providers work together to develop strategies for better workplace software and to create a de-facto standard for a *user-driven* software certification process. To accomplish this task, different measures have been taken: The "IT-Kartan, Användare och IT-System i svensk näringsliv" [2] describes the status quo in the Swedish industry concerning the difficulties users have to deal with when using the IT-tools at their workplace. Corresponding "IT-Maps" for other industrial sectors of the Swedish economy are being developed. Additionally, first steps have been taken to establish a quality label for existing user-friendly software products. Since 2000, UsersAward has annually awarded an IT-prize to a software product, which has been nominated by end-users and fulfils certain quality criteria [5]. These quality criteria also build the basis for the quality-label "User-Certified", for which software providers can put forward their software packages. Recently, some endeavours have been made to transfer this process to other European countries, such as Finland, Austria and Germany[1]. In this paper, first results of a UsersAward pilot study in Germany are presented.

USERSAWARD'S QUALITY CRITERIA
The quality and success of a software product at the work place are determined both by the context of use within the organisation and the characteristics of the software itself.

Based on preliminary research, UsersAward devised six success factors and developed a questionnaire for measuring their fulfilment. The questionnaire quantifies users` satisfaction with a software product on six dimensions: *total benefit* (4 items), *deployment process* (5 items), *technical design* (10 items), *support for work tasks* (6 items), *support for communication and cooperation* (5

[1] In Germany, the feasibility of "UsersAward" is examined in a project financially supported by the "Hans Böckler Stiftung" and carried out by a cooperation between three institutions, namely "bao – Büro für Arbeits- und Organisationspsychologie GmbH" (under J. Hurtienne and T. Vöhringer-Kuhnt), "BIT – Berufsforschungs- und Beratungsinstitut für interdisziplinäre Technikgestaltung e.V" (under S. Floegel and R. Linz) and "TIBAY - Technologie- und Innovationsberatungsagentur in Bayern e.V. beim DGB" (under S. Heegner).

items), and *quality assurance* (2 items). Each item is measured on a 6-point Likert scale from 1 (statement doesn't apply at all) to 6 (statement totally applies). Additionally, a "don't know" and a "this question is not important" answer category are provided. The software fulfils the conditions to achieve the IT-prize and the quality label, respectively, if a mean value of at least 4.0 is attained for a software product on 80% of all questionnaire statements by all interviewed persons. Furthermore, a mean value of at least 4.0 has to be attained on 67% of all questionnaire statements by both men and women, and a mean value of at least 4.0 has to be attained on 67% of all questionnaire statements for each user category [5].

METHOD

The German Pilot Study

The German pilot study took place within the framework of a university class concerning the measurement of customer satisfaction with software. A German version of the Swedish UsersAward questionnaire was applied to eight software products[2]. End users and their managers from eight different companies took part in the survey.

To guarantee the feasibility of the study in the context of a university class, a slightly different approach concerning the sample in the study was preferred. According to the Swedish UsersAward procedure 10 *percent* of the users of a software package in a company are asked to answer the questionnaire and three of them are interviewed with regard to the motivation for their judgments given in the questionnaire. In the German pilot study 10 *users* per software package and company took part in the written survey, and one of them was interviewed. Additionally, one manager per company was invited to fill in the questionnaire and to take part in the oral interview.

Participants

The participants of the study included 82 end users (36 female) and eight managers (one female) from different companies. Age groups ranged from 19 to 60 for the end users (mean = 33, std = 9.1) and 29 to 59 for the managers (mean = 39, std = 8.6). Computer experience ranged from 2 to 25 years for the end users (mean = 12, std = 5.5) and from 12 to 26 years for the managers (mean = 19, std = 5.2). The end users' experience with the examined software ranged from 2 to 216 months (mean = 54, std = 46) and managers had worked with the software under inspection between 12 and 219 months (mean = 79, std = 75).

Material

Software packages

The software examined in the pilot study included:

a. A novel software package for preparing credit agreements used by the majority of employees of a big German financial service provider.

b. An individual call-centre software package used in a financial customer service centre of a big German bank.

c. A standard software package for accounting transaction and tourist traffic management used in a small traffic agency.

d. An information system used at a compulsory health insurance company. The software contains several modules; two of them were evaluated.

e. A standard software package for enterprise resource planning (ERP) in small and medium-sized enterprises used by a software service provider. The module for order processing was inspected.

f. The sales module of an ERP software package used in a small IT-warehouse.

g. A stock data management system of a life-insurance agency.

h. An established integrated data handling and information system of a big health insurance company.

Questionnaires

Users and managers were asked to fill in two questionnaires with respect to the software examined in their company. The first questionnaire was the German translation of the UsersAward instrument. In addition to the translation of the statements, the scale was adapted to the German school grading system ranging from 1 (here: "statement fully applies") to 6 (here: "statement doesn't apply at all"). Hence, in comparison to the Swedish questionnaire the scale polarity is reversed. However, to make the results comparable to Swedish findings, data were recoded in accordance with the Swedish scale polarity.

The second questionnaire used in the pilot study was the ISONORM 9241/10 [3], measuring the conformance with the requirements of ISO 9241 ("Ergonomic requirements for office work with visual display terminals"), Part 10 ("Dialog principles"). The latter questionnaire operationalises the seven ergonomic principles "suitability for the task", "self-descriptiveness", "controllability", "conformity with user expectations", "error tolerance", "suitability for individualization" and "suitability for learning". For each principle, five bi-polar statements are to be rated on a 7-point scale ranging from "- - -" (1) to "+++" (7). The cut off criterion for a software package to fulfil the ISO

[2] The pilot study was carried out at the "FHTW-Berlin – University of Applied Sciences" by J. Prümper and J. Hurtienne during the autumn term 2004/2005. Thanks are due to the students who participated in this project and helped us to collect the empirical data.

requirements is a mean value of ≥ 5 (which refers to the "+" on the rating scale).

Procedure

After the companies and the software packages for the case studies had been selected, the questionnaires were sent to a contact person in each company, who organized the distribution of the questionnaires in house. One of ten end users who had filled out the UsersAward questionnaire was interviewed and asked for his/her motivation and reasons for his/her answers in detail. One manager from each company took part in the study in written and oral form. The statements during the interviews were written down and stored in electronic form, the questionnaire data was transferred into a statistical software package. Data from all samples (end users and managers) are included in the analysis.

RESULTS

In this section, the overall results with respect to the quality demands for the software certification and scale characteristics are reported. Furthermore, reliability and validity measures for the German instrument are provided.

Certification Criteria

One result of the pilot study is that *none* of the inspected software products achieved the necessary amount of confirmed statements to fulfil the certification quality criteria, relating to the overall results of the UsersAward questionnaire for all participants (Figure 1).

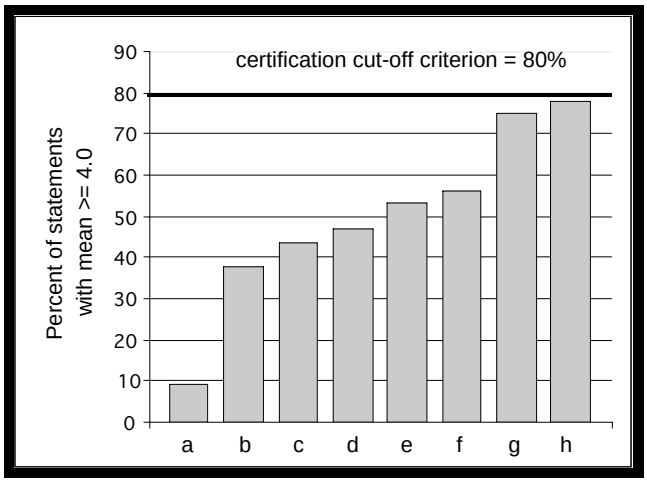

Figure 1: Certification result for all software products a … h

Regarding the overall mean value, a t-test shows no significant differences between mean ratings of men (mean = 3.9) and women (mean = 3.7), $t = 1.49$, $p \geq .14$, $N = 87$, and between users (mean = 3.8) and managers (mean = 4.2), $t = -1.38$, $p \geq .17$, $N = 90$, respectively.

Descriptive Statistics

A descriptive analysis has been performed on the data set of the pilot study. We show means and standard deviations for the six dimensions collapsing over all 8 software packages (Table 1).

UsersAward Subscale	min	max	mean	std
Total benefit	2.28	5.62	4.43	0.99
Deployment process	2.45	4.40	3.18	0.75
Technical design	3.35	4.85	4.01	0.48
Support for work tasks	2.90	4.46	3.89	0.59
Support for communication and cooperation	3.50	4.60	4.10	0.44
Quality assurance	2.29	4.25	2.96	0.66

Table 1: Results of the UsersAward evaluation of eight software products

Overall, the software products received the worst evaluation with respect to "quality assurance" (mean = 2.96), while the best results were achieved in "total benefit" (mean = 4.43), followed by "support for communication and cooperation" (mean = 4.10) and "technical design" (mean = 4.01).

Reliability

To estimate the reliability of the German UsersAward questionnaire, Cronbach's Alpha as an index for internal consistency was calculated for each of the proposed subscales (Table 2).

UsersAward Subscale	Cronbach's Alpha
Total benefit	0.95
Deployment process	0.77
Technical design	0.79
Support for work tasks	0.79
Support for communication and cooperation	0.56
Quality assurance	0.31

Table 2: Cronbach's Alpha of subscales

According to DeVellis [1], Cronbach's Alpha – values over 0.80 are very satisfactory, over 0.70 acceptable and below 0.65 undesirable. Thus, four of the six subscales show satisfactory internal consistency.

Despite the illustrated constraints, the overall Cronbach's Alpha of the questionnaire is remarkably high (= 0.94 for

all 32 items), which suggests that the questionnaire could be shortened by removing certain items. On the contrary, internal consistency could improve by removing items with undesirable item characteristics.

Validity

For the UsersAward questionnaire, mean scale values for the six dimensions were determined in two different ways: across participants (part) and across software products (sw). Then, the correlations with the matching subscales in the ISONORM 9241/10 were calculated.

UsersAward Subscale	r_{sw} [1]	r_{part} [2]
Total benefit	.07	.18
Deployment process	.38	.43**
Technical design	.67*	.65***
Support for work tasks	.42	.46***
Support for communication and cooperation	.53	.33*
Quality assurance	.14	.39**

Note: * p<.05, ** p<.01, *** p < .001 (Spearman-Rho; one-tailed); [1] N = 8 software products; [2] N = 37 - 60 participants

Table 3: Correlations with ISONORM 9241/10

The subscale "technical design" shows the highest correlation, as could have been expected considering the scope and internal structure of the ISONORM 9241/10 questionnaire.

Similar to the results for the UsersAward questionnaire, the certification criterion of the ISONORM 9241/10 questionnaire (i.e. mean score above 5) is not achieved by any of the eight software products (min = 4.04, max = 4.68, mean = 4.33, std = 0.27).The highly significant correlation on the subscale "technical design" in the product perspective ($r = .67$, $p < .05$, $N = 8$) verifies the validity of this subscale.

CONCLUSION

The fact that none of the software products reached the quality criteria recommended by UsersAward could be confirmed with the results of the ISONORM 9241/10. The descriptive statistics of the subscales of the UsersAward instrument point to some inconsistencies, probably as a result of ambiguous item verbalisations (e.g. two questions in one statement) or statements, which might not be eligible in the German context of use and customisation of work place software, both because the attitude towards a satisfying software product and towards the (software supported) cooperation in every day work life itself may be different between Sweden and Germany. This might also explain the differences in internal consistency between the more technical orientated subsections on the one hand, and the more organisational orientated subsections on the other hand. Due to the undesirable low values for Cronbach's Alpha, the items of the sections "support for communication and cooperation" and "quality assurance" need further investigation by conducting a detailed item analysis. Items with unfavourable item characteristics should be removed or reworded. The unequal number of statements for each of the subscales has to be reconsidered. If a different weighting of the subscales is not intended, each subscale should be made up of the same number of statements, or a scale mean value across all statements of each scale has to be calculated to balance their relevance. Additionally, the polarity of the scales should be harmonized between the Swedish and the German instrument to simplify comparison of the results. As a matter of course, it has to be examined whether the intended dimensions of the questionnaire can indeed be found in the data. Hence, the dimensionality of the instrument has to be verified with a factor analysis and the existence of subscales has to be proven empirically.

All things considered, the German adaptation of the Users-Award questionnaire shows acceptable quality criteria. Some items and subscales might need reviewing, but the reliability of the overall instrument is satisfying. The validation with an external questionnaire points in the right direction, especially under the constraints described. Additional validation measures could be calculated to lend more support to the validity of dimensions other than the technical part. Whether the overall instrument (and not only the subscale on "technical design") measures what it is intended to measure (i.e. user satisfaction with a software product) requires further research.

REFERENCES

DeVellis, R. F. *Scale Development: Theory and Applications*. Sage, Newbury Park, 1993.

Lind, T. *IT-Kartan, Användare och IT-System i svensk näringsliv*. LO, Stockholm, 2002.

Prümper, J.: Test IT: ISONORM 9241/10. In H.J. Bullinger and J. Ziegler (Eds.), *Human-Computer Interaction – Communication, Cooperation, and Application Design*. Lawrence Erlbaum Associates, Mahwah, New Jersey (1999), 1028-1032.

Sundblad, Y., Lind, T. and Rudling, J. IT Product Requirements and Certification from the Users' Perspective. In: H. Luczak, A.E. Çakir and G. Çakir, (Eds.), *Work With Display Units WWDU 2002 - World Wide Work*. ERGONOMIC, Berlin. (2002), 280-282.

Walldius, Å., Sundblad, Y., Bengtsson, L., Lind, T. and Sandblad, B. User-Driven Quality Certification of Workplace Software, the UsersAward Experience. In: H. M. Khalid, M. G. Helander and A. W. Yeo (Eds.): *Work with Computing Systems*. Damai Sciences, Kuala Lumpur, (2004), 150-155.

Certification of the User-Centredness of Development Organisation – A Way for Ensuring User Acceptance even before the Development of Software?

Timo Jokela
P.O. Box 3000
90014 Oulu University, Finland
timo.jokela@oulu.fi

ABSTRACT
Usability maturity models are methods for evaluating the capability of software development organizations in developing usable software. Usability maturity assessment could, in principle, be used to certify the usability and user acceptance of software even prior the software development project is started.

Author Keywords
Usability, usability maturity model, UMM, user-centred design

INTRODUCTION
Could the usability and user acceptance of a software system be ensured during development, before the system is delivered and used?

Analogous questions have been a key topic in discussions among software engineering for years. Is one able to assess prior the start of a development project whether a software development organization capable to develop fault-fee software in a planned schedule? Probably many usability practitioners know – at least by the name - CMM, Capability Maturity Model [14], and its more recent version CMMI [15]. CMM and CMMI as well as the international standard ISO 15504 [10] are systematic methods – more precisely, *process assessment* models - for evaluating the capability of companies in software development. The idea behind these models is that the higher ratings a company gets in an assessment, the higher probability is that it is capable to deliver fault-free software in schedule.

It may be a surprise to many usability practitioners that there also exist analogous models which are aimed at assessing the performance of user-centred design. The models are often called *usability maturity models*, *UMM's*[3]. Considerable efforts have been used to develop such models from early 90's.

A usability maturity model, UMM, is a method for evaluating the level of user-centredness of a software or product development organization. Usability maturity assessment can be conducted in order to know whether a supplier is capable of designing usable software. Usability maturity assessment, in principle, could be a useful thing for ensuring the usability of the system-to-be-developed in a contract-based system development setting. The idea behind usability maturity thinking is that *a company that gets high ratings in a usability maturity assessment is able to deliver usable (user-driven) software.*

In this paper, an introduction to UMM's is given, and the feasibility of certification of the user-centredness of software development is briefly discussed.

WHAT IS A USABILITY MATURITY MODEL?
A usability maturity model includes three main elements: (1) a *UCD reference model*, (2) *a performance scale*, and (3) *assessment guidelines*.

(1) A *UCD (user-centred design) reference model* defines the organizational elements of UCD that can be included in an assessment. One can examine the user-centredness of individual development projects, the management of development projects, or wider organizational elements such as UCD in quality system, usability skills, awareness on usability and usability in business strategy.

The basic scope of an assessment is typically to examine the UCD practices that are carried out in individual development projects. This is reasonable because if there is to be any effective UCD in an organization, it should be

[3] Other terms such as *usability capability maturity model* are also used.

visible in development projects. In practice, this means the examination of the extent to which usability engineering activities (user analysis, task analysis, usability requirements determination, usability evaluations, etc.) are effectively carried out in the project.

The management of UCD in a development project is often the next, wider step. Issues such as inclusion of UCD activities in a project plan, follow-up of the implementation of the plan during the project, and configuration management of the documents produced can be included in an assessment.

Another organizational element which could be in the scope of an assessment is UCD infrastructure, i.e. those UCD resources that any development project can utilize when planning and carrying out UCD activities. Typically these include the procedures, templates, tools and their support, staff training programs, etc, which are documented as a part of the quality process system in the company. Further, other foci of an assessment could be usability skills, management support, awareness of and attitude of personnel towards usability, and the organizational position of usability persons.

(2) A *performance scale* is used to rate how well an organization performs in the elements that are included in an assessment. The assessment results are a set of ratings, for example on a 1 to 5 scale. The closer the organization meets the requirements of the UCD reference model, the higher its ratings. Figure 2 illustrates an example where seven organizational UCD elements are rated with a 5-level performance scale.

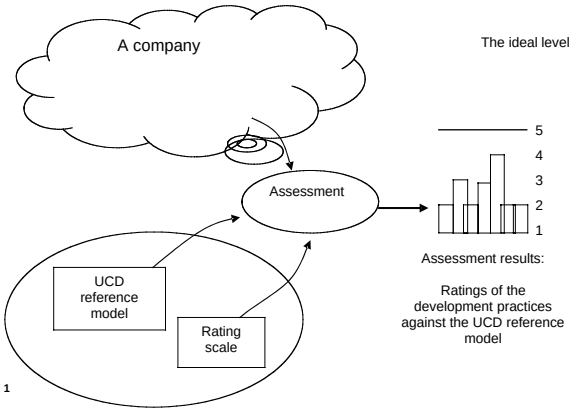

Figure 2. Development practices are rated against a UCD reference model.

(3) *Assessment guidelines* provide the practical guidance for how to carry out an assessment. An assessment is typically carried out by a team consisting of a lead assessor and assistants from the assessing and assessed originations. The assessors gather data by examining available documentation and interviewing the different stakeholders of the project or the company. The assessment team interprets the findings, generates the results and presents the results to the customer. Advice on improvement may or may not be given depending on the purpose of the assessment.

DIFFERENT USABILITY MATURITY MODELS

An overview of various UMM's is given in Table 4. The first UMM models were developed in North America: Trillium [1] by Bell Canada (a general assessment model including a specific part for usability engineering) and Usability Leadership Maturity Model, UMML by IBM [6].

Model	**Developer**
Trillium	Bell Canada
ULMM	IBM (US)
HPA	Philips (UK)
UCDM	HUSAT Research Institute (now the Ergonomics and Safety Research Institute, ESRI), Loughborough University (UK)
UMM-HCS	European INUSE project (further refined in the European TRUMP project)
UMM-P	European INUSE project (further refined in the European TRUMP project)
ISO 18529	International Organization for Standardization (ground work: UMM-P)
KESSU	Oulu University (Finland)
DATech-UEPA	DATech (Germany)
HCD-PCM-D	Mitsubishi Research Institute, NTT advanced technology and Otaru University of Commerce (Japan)
HCD-PCM-V	Mitsubishi Research Institute, NTT advanced technology and Otaru University of Commerce (Japan)
HFIPRA	UK Government HMIS and HFICMM projects (work performed by Lloyd's Register and Process Contracting Ltd.
ISO 18152	International Organization for Standardization (ground HFIPRA)

Table 4. Various UMM models in a rough chronological order. The exact years of publication are not known; the order is based on the years of publications.

In Europe, HumanWare Process Assessment, HPA, [16] was introduced by Philips, and User Centred Design Maturity, UCDM, [5] by Loughborough University. In the late 1990s, Usability Maturity Model: Processes, UMM-P, [3] – which follows the format of software process assessment (ISO 15504) - and Usability Maturity Model: Human-Centredness Scale, UMM-HCS, [4] were developed in a European research project. Later, an ISO Technical Report ISO TR 18529 [9] was produced based on UMM-P.

The latest developments are ISO 18152 [9] – which has the broadest scope of any model -, DATech [2] in Germany, SDOS [13] in Japan, and KESSU [12] in Finland.

The level of user-centredness of company can also be evaluated against ISO 13407, 'Human-centred design processes for interactive systems', [8]. The outcome of an ISO 13407 evaluation is 'pass or fail': a development organisation either conforms to the requirements of the standard or not. Such an evaluation is, however, not an 'assessment' in the sense of usability maturity models. An elementary feature of usability maturity assessment is that the result is a rating (or more typically: a set of ratings): the higher ratings an organisation gets, the better is its performance.

The various UMM's have diverged features:

(1) *Paradigm*. UMM-P, HFIPRA and the design process part of HCD-PCM represent a standard process capability assessment approach – they are models that strictly follow the format and requirements of the process assessment standard ISO 15504. The use of these models is similar with each other; an assessor needs only to learn the definitions of the processes to be assessed.

The scope Trillium, HPA, UCDM, KESSU and the visioning process part of HCD-PCM is user-centred design processes but they do not follow the standard ISO 15504.

The ULMM, UMM-HCS, DATech-UEPA and the visioning part of HCD-PCM are different from the other models in the sense that they include also other foci than processes. For example, ULMM has foci of assessment such as 'Organizational Awareness' and 'HCI Resources' that are not processes. DATech-UEPA uses the position of usability engineer as one assessment criterion.

(2) *Scope*. All the UCM models cover the examination of the performance of usability engineering. One can basically use any model for examining to which extent usability engineering activities are carried out in development projects. There are, however, differences in the details. For example, in standard process assessment one typically examines the *extent* to which a defined set of *base practices* are performed while KESSU examines the *extent, quality* and *impact* of the *outcomes* of the UCD activities.

Most models can be used to examine the management of UCD activities. The standard process assessment methods are specifically systematic in this viewpoint, addressed at the capability level 2[4]. Most models also address the position of UCD in quality management systems. All the standard process assessment models explicitly examine this issue at the capability level 3. They provide comprehensive and detailed results: each process is examined separately, up to the level of systematic process improvement. Quality systems are also addressed by Trillium, HPA, UMM, and DATech-UEPA.

The role of usability and UCD in strategic level processes is explicitly addressed by HFIPRA, ULMM, and HPA. Systematic improvement of UCD is addressed inherently by standard process assessment approaches, and by ULMM and by UMM-HCS. Further, the ULMM model examines issues such as organizational awareness, skills, impact and resources of usability specialists. UCM-HCS includes viewpoints such as awareness and training of usability and UCD. DATech-UEPA additionally addresses the organizational role of usability persons: the highest rating is given if "usability engineer is part of the design team and is responsible for design decisions".

(3) *Documentation*. The level of detail of guidance for carrying out an assessment varies significantly. Probably the best basis is provided in standard process assessments. The approach is documented in detail in standards, and there exist established schemes and training for how to become an assessor. On the other hand, such training is necessarily needed since the terminology and format are likely to be confusing for those without a background in process assessment.

Trillium also provides clear documentation. The documentation of DATech-UEPA seems to be a comprehensive one. However, it is documented in German only. UMM-HCS is clearly documented, too. The KESSU UCD reference model is recently updated [12]. The available documentation of ULMM, HPA, and UCDM is very limited. Their practical implementation requires a lot of interpretation, and thereby may not lead to results that the creators of the models had in mind.

FEASIBILITY OF CERTIFICATION OF THE USER-CENTREDNESS OF DEVELOPMENT PROCESSES

In practice, UMM's have mainly been used when a company which wishes to improve its performance of user-centred design. A company may also be interested to compare the performance of user-centred design with that of another company. Is one's company competitive in usability design within the industry? Does a company rate better or worse than the others, in which respects, and does it matter?

One basically also could certify the user-centredness of a software development company with usability maturity assessments: how seriously users are considered when systems are developed, and how professionally user data is gathered and integrated in designs. One could think that if a development company is certified to be user-centred, it is willing and capable of developing systems that users will accept.

In the author's knowledge, however, assessments for certifying the user-centredness of a development organisation have not been carried out. There are some

[4] The space of this paper does not allow discussion on the capability levels of standard processes assessment; more information of can be found e.g. from [7].

challenges that need to be answered before such certifications:

- Which UMM model to choose? There are standard models (ISO 18529 and 18152) but, for example, the author did not find them applicable in his assessment trials [11][5].
- About the organizational aspects to be covered in an assessment: should one, for example, examine development processes only, or should one also assess issues such as usability skills or usability in quality systems?
- A fundamental question is about the appropriate capability level for achieving a certification. If the capability scale spans, for example from 0 (lowest) to 5 (highest), which level should be reached for issuing a certification. Is level 3, for example, adequate or should level 4 be required for achieving a certification?

The author's belief is that the development of a usability maturity certification procedure requires empirical 'user-centred' research. In this context, 'users' would be the different stakeholders of an assessment, i.e. software suppliers and purchasers, and the assessor organizations. The certification procedure to-be-developed should be perceived sensible and useful.

CONCLUSION

There exist many usability maturity models, UMM's, that provide a technical basis for the certification of the user-centredness of development organizations. However, certifications based on UMM's are not today's practice, and some fundamental questions need to be answered if such certifications are wished to be carried out.

REFERENCES

1. Coallier, F., R. McKenzie, J. Wilson, and J. Hatz, *Trillium Model for Telecom Product Development & Support Process Capability, Release 3.0. Internet edition*. 1994, Bell Canada.
2. DATech, *DATech-Prüfbaustein. Usability-Engineering-Prozess, Version 1.2*. 2002, Deutsche Akkreditierungsstelle Technik e.V.: Frankfurt/Main. p. 70.
3. Earthy, J., *Usability Maturity Model: Processes. INUSE/D5.1.4(p), EC INUSE (IE 2016) final deliverable (version 0.2)*. 1997, Lloyd's Register: London.
4. Earthy, J., *Usability Maturity Model: Human Centredness Scale. INUSE Project deliverable D5.1.4(s). Version 1.2.*. 1998, Lloyd's Register of Shipping: London.
5. Eason, K. and S.D. Harker, *User Centred Design Maturity. Internal working document*. 1997, Department of Human Sciences, Loughborough University: Loughborough.
6. Flanagan, G.A., *Usability Leadership Maturity Model (Self-assessment Version). Delivered in a Special Interest Group session*, in *CHI '95: Human Factors in Computing Systems*, I. Katz, et al., Editors. 1995: Denver, USA.
7. ISO/IEC, *15504-2 Software Process Assessment - Part 2: A reference model for processes and process capability*. 1998: ISO/IEC TR 15504-2: 1998 (E).
8. ISO/IEC, *13407 Human-Centred Design Processes for Interactive Systems*. 1999: ISO/IEC 13407: 1999 (E).
9. ISO/IEC, *18529 Human-centred Lifecycle Process Descriptions*. 2000: ISO/IEC TR 18529: 2000 (E).
10. ISO/IEC, *15504-2 Process Assessment - Part 2: Performing an assessment*. 2003: ISO/IEC 15504-2: 2003 (E).
11. Jokela, T., *Evaluating the user-centredness of development organisations: conclusions and implications from empirical usability capability maturity assessments.* Interacting with Computers, 2004. **16**(6): p. 1095-1132.
12. Jokela, T., *The KESSU Usability Design Process Model. Version 2.1*. 2004, Oulu University. p. 22.
13. Kurosu, M., M. Ito, Y. Horibe, and N. Hirasawa. *Diagnosis of Human-Centeredness of the Design Process by the SDOS*. in *Proceedings of UPA 2000, Usability Professionals Association*. 2000. Asheville, North Carolina.
14. Paulk, M., C. Weber, S. Carcia, M. Chrissis, and M. Bush, *The Capability Maturity Model: Guidelines for Improving the Software Process*. 1995: Addison-Wesley.
15. SEI, *Capability Maturity Model Integration (CMMI), Version 1.1. Staged Representation*. 2002, Carnegie Mellon University, Software Engineering Institute: Pittsburgh. p. 729.
16. Taylor, B., A. Gupta, W. Hefley, I. McLelland, and T. Van Gelderen. *HumanWare Process Improvement - institutionalising the principles of user-centred design. In Tutorial PM14 H Human-centred processes and their impact.* in *Human-Computer Interaction Conference on People and Computers XIII*. 1998. Sheffield Hallam University.

[5] One reason was that the companies perceived the results too complex and too difficult to understand.

Profitability and ICT literacy
- Japanese Experience -

Takehiro Eto
Japan Productivity Center for Socio-Economic Development
3-1-1, Shibuya, Shibuya-ku, Tokyo, Japan
t.etou@jpc-sed.or.jp

ABSTRACT
In this paper, We introduce the outline of the Self Assessment Guidebook that Japan Productivity Center for Socio-Economic Development(JPC=SED) developed in 2002. This guidebook is a Tool for organizations to diagnose the strength and weakness of ICT usage. In addition, we have done the proof experiment of the self assessment in four Japanese companies. We

Author Keywords
Guides, instructions, author's kit, conference publications.

INSTODUCTION OF SELF ASSESSMENT GUIDEBOOK
The Self Assessment Guidebook based on Criteria for Information Value Development is a tool to identify the criteria factors in an organization's information systems for enabling it to stay in touch with it's market and customer, which are changing faster than ever.

The Perspective of the Guidebook
This Guidebook has an assessment approach that is in line with such assessment schemata as the Malcom Baldrige Award and the Japan Quality Award(JQA). It also contains idea derived from various empirical studies of Japanese and International organizations and prevailing studies on the subject of assessing the information systems of organizations. As is the case with the JQA, the Guidebook has Criteria to assess the existence of systems that ensure the functioning of the management cycle.

The Criteria is designed to be used in three steps. The first step is to understanding the current status. Among other things, the existence of a definition for the organization's core value and how the organization is managing and utilizing information are reviewed. The second step is to assess the organizational structure for promoting the use of an information system within the organization. The readiness of the organizational structure, how literacy is defined, and whether the management cycle is continuously improving the effort to deploy the information system are all assessed. The third step is an assessment of the result.

THE SELF-ASSESSMENT CRITERIA
The Criteria of this Guidebook is based on the business processes of corporations. The use of the Criteria means that all of these processes are assessed by the various concepts that support the Guidebook and the applications of the management cycle.

The Structure of the Criteria
All six Criteria are designed to have the management cycle applied within them. These six Criteria are

1) Clarification of objectives and strategies
2) Assessment of the current situation
3) Establishment of a goal
4) Establishment of an organization and method to achive the goal
5) Modification of the plan
6) Assessment of the results and the monitoring of problems

By applying the cycle in actual business activities, the creation of information value and innovation are realized. That in turn increases the organization's level of maturity, leading to the next improvement in the information system's effectiveness and information value creation

The Outline of the Criteria
Organization Profile: The Organization Profile is a summary description of the organization that is going to be assessed. The production of the Profile is the first step of the assessment. The Profile usually describes the organization's past, present, and future success factors in the utilization of IT. It contains an internal view of the organization itself and the status quo and is to be used together within the Criteria in making the self-assessment.

Criterion 1 “Definition of information value”: Gain an understanding of the basic concepts and definitions of information and communication. Define what kind of data is critical for respective organizations and examine whether the definition procedure is appropriate.

Criterion 2 “Development, utilization, and management of information (knowledge)”: Evaluate the current system for developing, utilizing, and managing information, assess its level and see if there is a method and organization to further optimize the system.

Criterion 3 “Organization for information value development”: Verify whether the information system and the structure to manage the system is aligned with the overall strategy, core strength, and alliance of the organization. Leadership, allocation of duties, and the deployment of plans are also assessed.

Criterion 4 “ICT(information and communication technology) literacy of the organization and its members”: Assess the current level of ICT literacy, the system to develop that literacy, and how effective it is. Verify whether the system and organization to define the necessary literacy is adequate.

Criterion 5 “Creativity and continuous improvement”: The introduction of ICT is sometimes fraught with unexpected constraints, risks, and burdens. As such, the ability of (internal)end-users to apply creativity and improvements in optimizing the usability of the system are assessed.

Criterion 6 “Results”: This criterion evaluates whether the activities verified through Criteria 1 to 5 are indeed yielding expected results. The results encompass the extent of goal achievement, as customer satisfaction, increased decision making speed, increased synergy between business partners, increased business knowledge through the application of creativity, etc.

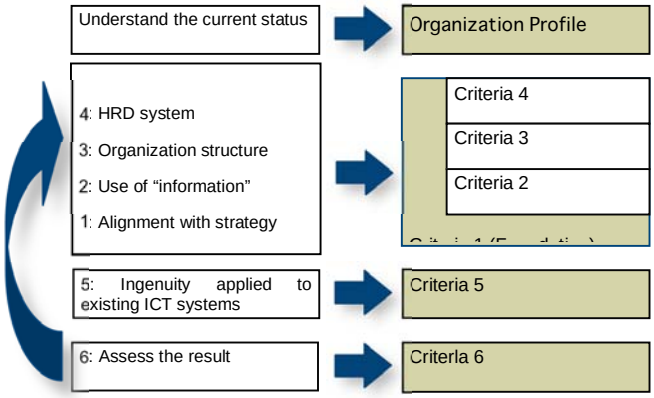

Figure 1. Self Assessment Criteria Structure

CRITERION 4: ICT LITERACY OF THE ORGANIZATION AND ITS MEMBERS

In this paper, we focus on the criterion 4 “ICT literacy of the organization and its members”. Criterion 4 are composed of six sub-criteria.

4.1. Alignment of corporate strategy and development of ICT capability

4.2. Recognize the need for literacy and training

4.3. Organization to develop and implement ICT capability

4.4. Programs for ICT capability development

4.5. Programs to improve usability

4.6. Publication and recognition of the ICT capabilities of organization members

CASE STUDY OF JAPANESE ENTERPRISE

We have done the proof experiment in 4 enterprises.

1. Musashino Ltd.

   Service of rental of cleaning articles, Environmental cleaning service

2. Textile Company

   Uniform manufacturer

3. Parts Supplier

   Spring manufacturer (l.e. the mobile phone “hinge”)

4. Confectionary Company

   Rice cracker manufacturer (sold at the Tokyo Disneyland)

The performance of all of four companies is very good. However, the big difference occurred in these companies’ descriptive content.

Alignment of corporate strategy and development of ICT capability

As for the company who mastered IT and has attained high performance, it became clear that the business strategy and the information strategy are related closely. In the case of Musashino Inc. of Fig.2, the ICT investment is performed very being conscious of realization of a business strategy. So, Criterion 4.1 are described clearly. In the case of the other company, the business strategy is clear. However the situation where the business strategy and the ICT strategy are not in cooperation is also seen.

Furthermore, in case of the Musashino Inc, it is defined as required skill for joining a company that a mobile phone with internet function can be mastered. In this investigation, in the other company, the ICT capability for business strategy realization was not defined as required skill for joining a company.

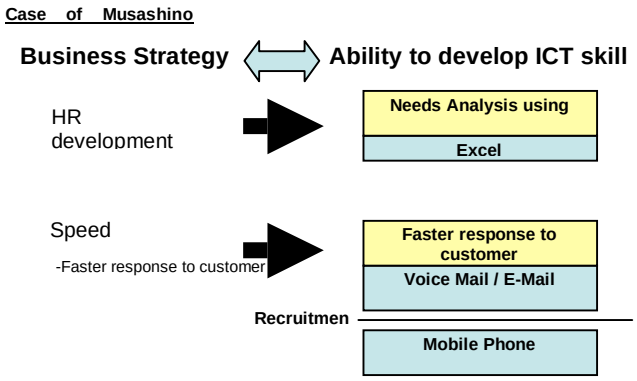

Figure 2. Busines Strategy and ICT strategy in case of Musashino

Organization to develop and implement ICT capability

Fig.3 shows the gradual ICT skill development image in Musashino Inc.

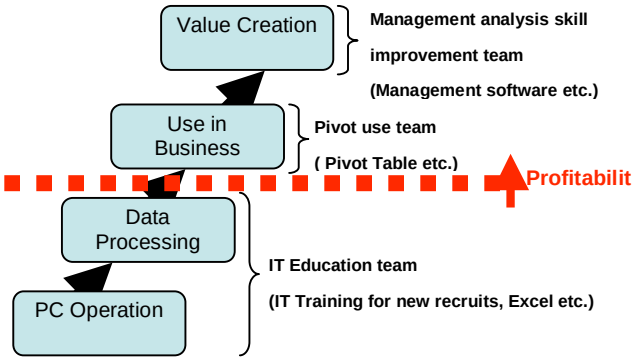

Figure 3. ICT skill development image in Musashino

Musashino Inc. defines gradual ICT skill development, in order to create value. First of all, to a new employee, it begins from the general PC skill , and even the method of data processing is trained. So far, other companies in this investigate are trained.

However, In the Musashino case, high training of added value is defined for management, and it has clarified also about the hierarchy which training should be given. In case of other companies, training exists. However, the training is not divided by class, moreover those who should take a class are not defined. Therefore, required training is not performed to a required person. The level which creates value using ICT is not reached.

Programs for ICT capability development

This proof experiment was conducted for the company who continues from the time of ICT hardly existing. It is about 2000 that ICT spread thorough the end-users. In such a company, the object which must be educated is considered by deviding into three about ICT.

- Existing employee
- Collage grads
- New employee

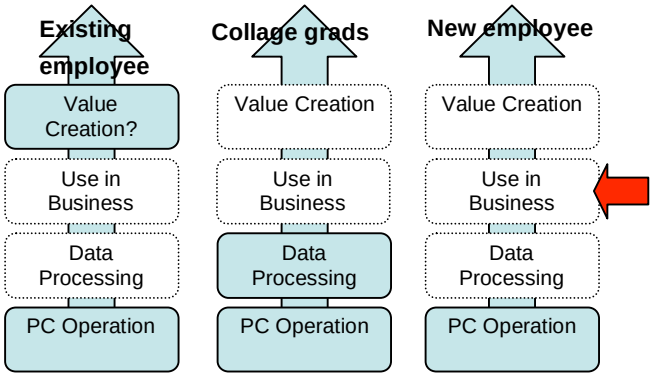

Figure 4. Three objects of ICT Training

In the company, when the time of PC being introduced and when joined the company, the education of general PC skill is performed. However, the educational system after PC skill training did not exist in one of the companies this time. Consequently, the end-users ware not using the system and the situation where existence of systems ware not known had occurred.

Common to the company who cannot achieve high performance with ICT, the education of the general PC skill is performed, however the education of how to use for skill is deterministically insufficient. Consequently, users cannot achieve high performance.

Programs to improve usability

In this case, there was no company with the program for improving usability.

The end-user has various dissatisfaction. However, they did not know what and where to say. Furthermore, IT section does not grasp IT practical use situation of end-user, either. Consequently, the system it is hard for end-users to use is built.

CONCLUSION

This time, in the actual proof experiment of the self assessment carried out to four companies, we introduced the result part of ICT and usability.

The company which has attained the high performance with ICT had three features

1. The business strategy and the ICT skill development program ware related closely.

2. The required capability of ICT was defined according to the hierarchy, and the ICT skill development program was defined similarly.
3. Top management had clear intension of ICT.

Other companies have attained the high performance. However, since end-users cannot master ICT, the situation that end-users was forced the remarkable burden had occurred.

In order for the end-users to attain high performance with ICT, Top management, end-users, IT section need to consider the three features together.

REFERENCES.

1. Information Value Assessment Criteria Preparation Committee. *The Self Assessment Guidebook based on Criteria for Information Value Development*, Japan Productivity Center for Socio-Economic Development, TOKYO, JAPAN, 2003.
2. Musashino, Inc. Self Assessment Project Team. *Record of the self assessment in Musashino, Inc.*, Japan Productivity Center for Socio-Economic Development, TOKYO, JAPAN, 2004
3. The Self Assessment introduction example in Musashino, Inc. http://www.infovalue.jp/criteria/musashino/index.htm

How to Teach User-Driven IT Design to Computer Science Students?

Karl-Heinz Rödiger
Universität Bremen, Dept. of Mathematics/Computer Science
P.O. Box 33 04 40
D-28334 Bremen, Germany
roediger@informatik.uni-bremen.de

ABSTRACT
Teaching user-driven IT design to computer science students the traditional way (by lectures and exercises) is a rather unsuccessful undertaking as long as there are no real users available; therefore other types of teaching are asked to prepare students for the cooperation with users. An attempt is presented, the main objective of which is directed at the alteration of attitudes towards users and their problems. Everyday interactions, which nonetheless are not very familiar to students, form the centre of the course. The experiences with these interactions should cause a better understanding of the users' situations.

Author Keywords
Courses in human factors, everyday interactions, attitude towards users.

INTRODUCTION
The concept of user-driven IT design is not a very new one; as it is said in the scope of this workshop "the 1990s saw an increased awareness of the need of user participation". Although it was already known since the 1980s, I never liked it, because the term verbalises that a stronger one is allowing some sort of participation to a weaker one. Real cooperation might have taken place only in a few Scandinavian projects, e.g. [3]. Software development has been an activity between unequal partners in most of the German projects during the 1980s. The disparity has many different reasons: from the investors' point of view participation is time consuming, expensive and not crowned with quick success; from a software engineer's point of view users are disturbing with comments, questions, and always new requirements; or even worse: they do not know most of the time what they want. And the users: neither trained in IT design nor familiar with the dialect spoken by computer scientists they could not oppose.

Most of the users are in the meantime experts not only concerning their work, but also in the use of software systems. Many of them are fit for cooperation in software development. But user experts alone cannot develop new and better systems. Software engineers are needed as adequate partners for that. In the course of studies in computer science we have the task to prepare students for this partnership. How could this task be fulfilled?

LOOK BACK IN ANGER
In the beginning of the 1980s German researchers discovered the field of human factors in computing. It was the time when the first personal computers have been available in department stores and at home; it was the time of Apple's Lisa, which everybody called a nice machine, but nobody knew what to do with it. And it was the time, when German researchers (the author included) started to solve the usability problem in a very typical German way: a standard (DIN 66234 part 8, the forerunner of ISO 9241 part 10 [4]) has been developed. A relevant part of those researchers has spent the following twenty years with the transfer of this standard into the international context, in developing instruments for the conformity test of software with this standard, e.g. [2], with founding test organisations, with the examination of software against those test batteries, and with the development of company style guides which accomplished the standard. The next highlight in the life of a German human factors researcher was the council directive 90/270/EEC on the minimum safety and health requirements for work with display screen equipment. The main interest in this directive has been to identify software offending against it, just as in DIN 66234 part 8. Tests have again been developed. The same happened in 1996 when the EC directive was implemented as a German by-law.

Laws and standards are helpful in conflict situations if one argues about the correctness of a solution. But the abstract dialogue principles of ISO 9241 part 10 suitability for the task, self descriptiveness, controllability, conformity with user expectations, error tolerance, suitability for individualisation, suitability for learning, are only restricted suitable for the development of new software. Therefore many

software engineers have problems translating them into action.

Special courses for computer science students in human factors or in 'software ergonomics', as it was called in Germany, have only been presented at few universities. Subjects of those courses have mainly been laws, by-laws and standards or how to develop software in conformity with this legal stuff. The objectives of those courses: computer scientists as experts of work, technology, and organisational environment, able to develop systems in conformity with all these juridical hurdles. Another subject of these courses has been participation: the users have been trained about their rights accruing from those laws.

Germany has more laws than any country in the world as it is told (I cannot proof it). This legal system is very well functioning, because (nearly) all Germans are believing in these laws. This might be an explanation, why Germans are tending to a legal treatment of things, which are primarily not juridical. On the other hand laws and by-laws give one the possibility to pursue their rights. This is – in a good working legal system – a fine starting point for the improvement of e.g. working conditions as long as fantasy is not dropped out. Therefore it is not argued here against legal regulations in its entirety. But as long as most activities in human factors exhaust in this, it must be criticised. In the meantime a lot of software might be compliant with the ISO standard; nevertheless most of it is inefficient, bleak, and not aesthetic.

Also in a good working legal system there is a big difference between having a right and getting one's right. To ascertain that SAP's[6] software R/3 violates the ISO standard is very easy; to accomplish the consequences is nearly impossible.

USER-DRIVEN DESIGN – WHO IS IMPLEMENTING IT?

Believing that this kind of university teaching – as described before – is not very helpful to create the right partners for user-driven IT design, one has to look for a different way. Two courses are mainly relevant for this at the University of Bremen: software engineering and human factors. In the software engineering course students usualy learn, how to develop big software systems. Because these students have solved problems like towers of Hanoi, n!, search and sort in their undergraduate courses, they have no real understanding for the subjects of software engineering.

To achieve a more devotional attitude towards the human ability handling complex situations and towards software problems in particular the software engineering every lecture has been started with Peter Neumann's Computer Related Risks [5]. In the beginning the students remained in disbelieving astonishment; in a while the regular drop has washed out the stone and they became thoughtful.

Software systems are very often changing human work; it is 'orgware'. Therefore work design and organizational development have been introduced in the software engineering course, just as much as enabling the participants to discuss these matters with organizers and users.

If one teaches human factors to computer science students you have to fight against two problems. One is the opinion that this subject is a quantité négligeable compared to the algorithmic problems. Students are easily to convince that the integration of differently structured data bases via XML[7] technology is important for their working life. Lecturing about project management, life cycle models and the right time of user cooperation, about human factors and user-driven design on the other hand is a comparable hard job.

Because computer science students are accustomed to poorly designed software and – in the meantime – have developed their own mechanisms to manage it, imagining the situation of a normal user is difficult for them to. They have learned to work with software without spending a short look into a manual, they have learned to explore software and to fix bugs, they have learned to work around software as long as it is malfunctioning, and they have learned to throw software away and to download a better one when the problems are arising too high. They never have learned a real user's situation sitting in front of a display, being forced to use a new, crude software system. Imagining such a situation is very complicated for a prospective software engineer, who never has experienced like this.

On the other hand students are swamped with the role of an expert in real working situations. They would manage unexpected situations as they do it in normal life: as computer scientists, not as users; they implement exactly this, their own mental model. That must not – as we know – fit to the users' ones.

The mismatch of mental models in error situations is often discussed during scientific workshops. This is indeed a severe problem; but before getting to this point, one has to think about attitudes. Attitudes of young software engineers just graduated from a university, being able to solve every software problem, thinking about faster and more efficient algorithms to definitely solve the travelling salesman's problem.

This young man or woman now shall understand the problem of a user searching for a letter in the Windows file system that he or she just has closed. This requests a change

[6] Systems Analysis and Program Development, world's third-largest independent software vendor

[7] Extensible Markup Language

in attitude towards users as well as towards their problems and it requests a change in teaching.

ONE MORE ATTEMPT IN TEACHING

It is impossible to lecture changes in attitude; at best you can tell students how they should behave when talking with users about their problems, hoping they will not do the opposite. Changes of attitude require own experiences. Therefore the idea arose last semester to put students into situations, in which they are real users, neither familiar with the new task nor with the system, which they should use to solve their task.

At the beginning of the last winter semester the students have been asked to look for a computerized piece of equipment, which is used in everyday life, but at the same time not very familiar for them. They should use these devices and afterwards make a presentation about their experience with the possibilities and malfunctions of these objects. The aims of the presentation have been an objective description of the embedded system and its use and a critical discussion of the possible design errors; if possible, some ideas about the potential reasons, and a redesign of all problems found in use. They were encouraged to bring the devices into the lecture hall for their presentations and demonstrate its use or to show films or photos. The presentations of everyday interactions have been discussed with all students. This was the content of the exercise part of the course; the lecturing was done 'on demand', that means, the problems and the discussions of a presentation have stimulated the next lecture. The whole course consisted of two hours lecture and two hours exercises weekly.

The students have chosen a wide variety of everyday interactions: camcorder, digital camera, digital clock, digital TV receiver, microwave, Chinese rice boiler, coffee machine, ticket machine with a chip card, synthesizer, photo robot and BMW's I-drive. All these things are equipped with chips; the functions are realized either in hard- or in software. The overall results: 1. great astonishment about the amount of mistakes, 2. sullenness about the poor design, 3. confirmation of Alan Cooper's thesis [1]: if you cross something with a computer, you will always get a computer.

Some outstanding mistakes ensured a lot of fun in the course: with the camcorder one could surf in the internet, the recording of films was some sort of a additional gimmick. With the digital camera one can choose between five languages, nevertheless all explanations are in English. The microwave allowed the choice between beef, chicken, lamb1 and lamb 2; the time setting was a matter trial and error. The rice boiler has been designated only for the Chinese market, the explanations are not understandable; the result therefore a cake instead of boiled rice. The photo robot is of the type 'one-armed bandit': the correct sitting position is always false; before you have chosen the type of photo that you want, you will loose your money. Nobody can or will tell you, which data are stored when chipping a ticket. The use of a mobile phone without a headset during driving a car will be punished in Germany; BMW's I-drive is needing more attention: circa 700 settings are to be controlled by one knob, a display in the middle of the console, and a female voice.

The more sophisticated problems of interaction design like context, suitability for the task, perception. affordance, mental models, metaphors etc. have been discussed subsequent to the presentations. The subject of the next lecture was chosen by the students out of the central problems of these discussions. By that a strong connection between knowledge and its application has been achieved. Preece/Rogers/Sharp [6] and Shneiderman/Plaisant [7] were used as textbooks.

... AND THE EVALUATION?

At this point every reader will ask for the results of this course. As everybody knows, the success of teaching can be checked at its best in practice. Not even an examination can verify the benefit of a course like this. Therefore the gentle readers must be patient with a possible evaluation.

Nevertheless something may be said about the success. Normally the students miss the lectures; they only take part in the exercises, because they achieve their credit points (ECTS) by that, not by hearing the lectures. It was the first time for years that the same amount of students took part in both parts of the course. Evidently they had a lot of fun and interest in this new attempt of teaching. The written working outs of their presentations have also mostly been excellent.

REFERENCES

4. Cooper, A. The Inmates are Running the Asylum. SAMS, Indianapolis, IN 1999.
5. Dzida, W. et al. Gebrauchstauglichkeit von Software – ErgoNorm: Ein Verfahren zur Konformitätsprüfung von Software auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11, Wirtschaftsverlag NW, Bremerhaven, Germany 2001.
6. Ehn, P. Work-Oriented Design of Computer Artifacts. Arbetslivscentrum, Stockholm, Sweden 1988.
7. ISO 9241-10 Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 10: Dialogue Principles, 1996.
8. Neumann, P.G. Computer-Related Risks. Addison-Wesley Publ. Co., Reading, MA 1995.
9. Preece, J., Rogers, Y., and Sharp, H. Interaction Design – Beyond Human-Computer Interaction. Wiley, New York, NY 2002.
10. Shneiderman, B. and Plaisant, C. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 4th ed., Pearson, Boston, MA 2004

IT-supported work.

Work environment problems and health costs

Bengt Sandblad
Uppsala University
Dept. of Information Technology
Box 337, 751 05 Uppsala, Sweden
Bengt.Sandblad@it.uu.se

ABSTRACT
Work environment and health problems of different nature are increasing as a consequence of computer supported work. There are different reasons for work environment problems. We will here distinguish between physical, psychosocial and cognitive work environment problems. Most efforts to improve the work environment are focused on physical aspects, and to some extent to psychosocial aspects. Mental workload and cognitive problems are of a more complex nature, more difficult to measure and provide efficient solutions to, and are more seldom studied or solved. The direct costs related to work environment problems are difficult to estimate. We will here present some rather accurate calculations together with some other more vague estimations. There also exist some other important costs related to development of IT work support systems. These are caused by inefficient development processes, and we will refer to some studies where the economic consequences of these problems are analysed.

Author Keywords
IT work support systems, computer-supported work, work environment problems, occupational health, stress.

INTRODUCTION
The rapidly increasing use of computer support systems in all sectors of working life has had a tremendous effect on efficiency, benefit for the organizations, flexibility and work environment – often a positive effect but sometimes, unfortunately, also negative. This development has in other words had undesirable side effects that have generate health risks, e.g. neck, upper limb, low back disorders and stress. When more and more of the work is computer supported, the effect of low usability will be severe. In many work situations today, there exists no work without the computer. The effect of low usability will here not only be loss of time, but it will also result in stress and in the long run to health problems.

Today, around 3 million people in Sweden, which is 66% of the Swedish work force, use computers in their work. 35% of the women and 30% of the men in the total work force use computers at least half of the working day (AV 2001). These figures have been increasing at a fast and steady pace since the workplace computerization began spreading in the late 1970s. According to the surveys on computer use performed by Statistics Sweden (SCB), the proportion of computer users in the total Swedish workforce increased from 24% in 1984 to 32% in 1989 and 51% in 1995 (SCB 1995). The proportion of users working half their working day or more at a visual display unit (VDU) has also increased, especially for women. (Ekman, Andersson, Hagberg & Wigaeus-Tornqvist 2001). Approximately one third of the work force has clerical office work and, as expected, the use of computers is higher in that group. From 1989 to 1997, the proportion of computer users among office workers increased from 65% to 90% (Marklund 2000). For women, the corresponding figure has continued to increase from 91% in 1997 to 94% in 1999 (Ekman, Andersson, Hagberg & Wigaeus-Tornqvist 2001). The introduction of computers in working life has had a great impact, dramatically changing the very nature of many jobs and the whole work situation for a vast number of people. As the use of computers has increased, health concerns and the reports of negative effects on users health have also increased steadily since the early 1980s (Bergqvist 1993, Punnett & Bergqvist, 1997, Sandsjö & Kadefors Eds. 2001).

The fact is that we are continuously introducing more and more information technology (IT) of a kind that is harmful to its users!

There are several interacting mechanisms through which work with VDUs affect the users, mechanisms related to the person, the work organization, the work tasks, the physical work environment and the technology that is used. To counteract the adverse health effects of VDU use, we need to know more about the relative importance of different factors and about interactions between these factors.

We also see that the systems development process can not handle this problem. All normal and commercially available

development models and methods for systems development fail to address usability and work environment issues in an efficient and functional way. The efficiency of the development process is also, according to both large investigations and our own experiences, too low. When the development project faces problems to keep dead-lines or to stay within budget limits, most often usability and work environment aspects are first to be traded off.

To be proactive, so that we can improve the quality of tomorrow's computerized work support systems, we have to gain a better understanding of the software development processes and try to improve and complement them. Within the research field of human-computer interaction (HCI), such efforts are made, often aiming at increased usability of computer systems. Considerable effort is spent on developing design methods emphasising the needs of the users, including methods for user-centred design (Norman 1986, Göransson 2001). However, the results of these efforts are in practise far from satisfying. The impact on software development from research on usability and user-centred design has been quite limited and health aspects are often completely ignored, (Clegg, Axtell, Damodarant, Farbey, Hull, Lloyd-Jones, Nicholls, Sell & Tomlinson 1997).

A class of work environment problems that becomes especially important to consider in intensively computer-supported work are the cognitive work environment problems. With cognitive work environment problems we mean when properties of the work environment hinder the workers to use their skills efficiently. These obstacles are often associated with the design of the information system. If the system has an inappropriate functionality, a poorly designed user interface or if the user does not have adequate skills such problems can occur.

Most activities aimed at improving the work environment for professionals in intensively computer supported work situations are adopted very late, i.e. when physical or psychosocial problems already have been manifested and reported. We believe that it is both important and fully feasible to address potential work environment problems already during the development process. We know, however, that organizations already have difficulties in addressing aspects related to the usability of artefacts under development. If we add the challenge to also include work environment aspects in the systems requirements, the task will become even more difficult in practice.

A MODEL FOR HEALTHY WORK

During the 1970s, Robert Karasek introduced a model to analyse work-related stressors associated with cardio-vascular illness. He used the two variables work demands and decision latitude (control opportunities). His demand-control model has then been further developed in collaboration with Töres Theorell (Karasek & Theorell 1990) and is now the most widely used model to analyse psychosocial work environment factors and their relation to health and well being. According to this model, the combination of perceived demands and perceived control at work is a determining factor underlying negative stress. High demands create stress responses that could be stimulating if combined with high personal control, but that cannot be effectively handled if control is low. The combination of high work demands and low decision latitude is referred to as high job strain and has been shown to be associated with the highest risks for health problems

Most work demands are mainly psychological stressors (e.g., time pressure) in the work situation. If we look closer into the psychological demands, we can separate them into cognitive and emotional categories. Cognitive demands have gained considerable attention in the HCI research field (Helander et al., Eds. 1997). Memory load is one typical example of a cognitive demand that has been thoroughly studied within that field. Control can also be divided into sub-concepts. It is often defined as consisting of two major components: the degree of personal control/decision latitude in the work situation and the degree of control over the competence used. Decision latitude describes the opportunity for the individual to exercise control over very concrete and practical decisions in the personal work situation, such as when to take a break. Competence control refers to the opportunity to use different parts of personal competence and to obtain stimulation and development through, e.g., variation in tasks.

The model was subsequently extended, becoming three-dimensional by adding the factor social support (i.e. support from supervisors and/or colleagues) (House 1981). House identifies several forms of social support; emotional, appraisal, informational and instrumental support. Scientific evidence exists suggesting that social support has an important effect on experienced work stress and on health (House 1981, Cohen, S., Syme, L. Eds., 1985, Wahlstedt, K. 2001). The feeling of having access to social support affects the individual's appraisal of and reactions to a stressful situation, and affects emotions, physiological responses and behaviour. The most favourable situation is one characterised by reasonable demands, high decision latitude and high social support.

Despite the strong scientific support for the demand-control model (Karasek & Theorell 1990), there has been some criticism and several studies have failed to find the expected correlations (Punnett & Bergqvist 1997). One possible explanation for this failure is that social support plays a more important role than suggested by earlier studies. This hypothesis is strongly supported by the work of Kurt Wahlstedt, which has shown that high decision latitude counteracts the negative impact of high workload only when social support is at an acceptable level (Wahlstedt, K. 2001). Studies by Töres Theorell and others (Barklöf, K. Ed. 2000b) have demonstrated the importance of support from management in order to counteract negative health effects that result from organizational changes.

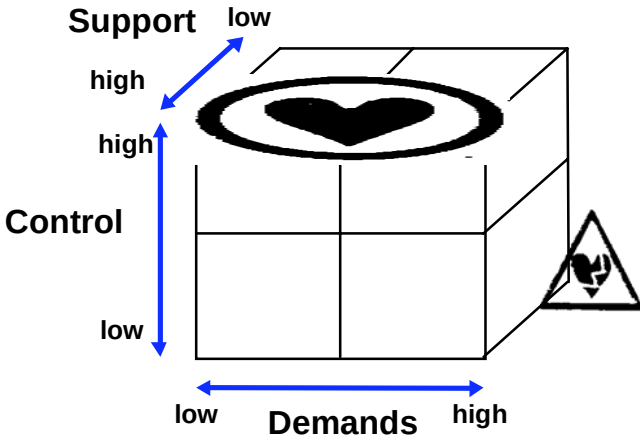

Fig 1. Relations between demands, control and social support in a work situation, according to the Karasek-Theorell model. High experienced demands are not a problem, if combined with high self control and high social support.

According to the classification in the demand-control model, work life is rapidly changing and an increasing number of people in more and more types of work find themselves in situations of high job strain (Marklund 2000). At the same time, there is also a growing group with both high demands and high decision latitude. Downsizing and lean production have characterised the structural changes in Swedish work life during the 1990s. Some of the effects are more jobs characterised by high decision latitude but also high demands (and often more unclear demands than before) as well as more stress-related health problems (Barklöf, K., Ed., 2000a, Barklöf, K., Ed. 2000b). In the debate on the growing problems with stress-related illness the term "honey trap" has been used to label this work situation, where people in "good" jobs with high salary and high decision latitude seem to voluntarily work so much that they become ill.

Analyses based on the demand-control-support model use psychological data, subjective perceptions and judgements. According to this psychological approach, negative stress arises when the person judges the demands in the specific situation as too high in relation to his or her resources to handle them. This means that it is not the objective situation but the person's interpretation of it that determines the type and degree of stress experienced. Cannon was probably the first to clearly describe the importance of psychological-emotional reactions and the interrelations between psychological and physiological reactions to threatening stimuli (Cannon 1949). He calls the emotions fear and aggression, "the keys" to all the astonishing complicated physiological reactions he studied in the human body. The emotional reactions are responses to threats, telling the individual that it is time for fight or flight, in which case the physiological responses prepare the organism for effective attack or escape.

PROBLEMS RELATED TO WORK ENVIRONMENT, USABILITY AND TO THE DEVELOPMENT PROCESS.

We will here only shortly discuss different types of problems related to qualities of computer support systems, and try to estimate some direct and indirect costs caused by deficiencies in the systems and processes.

The first class of problems are more traditional work environment problems, i.e. physical or psychosocial problems. There exist many studies of such problems

COSTS RELATED TO WORK ENVIRONMENT PROBLEMS

There exist, as shown above, a lot of research concerning different types of physical and psychosocial work environment problems. In some specific cases it is possible to make estimates of the costs caused by these problems. The problem is that there is almost never a simple relation between encountered problems and costs. The problem situation is most often very complex and we can not separate different causes from each other. If we assume that the results e.g. concerning costs related to health problems caused by bad work environment in IT-supported work (Wigaeus Tornqvist, E., Eriksson, N., Bergqvist, U., 2000), (Hagberg, M., A. Toomingas and E. Wigaeus-Tornqvist, 2002) are general to this kind of work, we end up with an annual cost for Swedish work life of 10 billion SEK.

COSTS RELATED TO USABILITY PROBLEMS

Also here we can in specific cases make more precise estimated of resulting costs. If we e.g. can reduce the time it takes to handle one administrative case by a factor 5, this will reduce the total working time. Additional effects, such as reduced physical work load, will make the effects even more positive.

If we make a rough calculation, we see that according to Swedish statistics roughly 15% of all work is directly computer supported. The Swedish work force consists of 3 million people. This adds up to an annual work time of almost 700 million hours. If we assume that 5% of the working time is lost because of bad usability in the support systems (investigations have shown that the figure probably is much higher) we have a total annual loss of production of 35 million hours and the corresponding cost for this is around 10 billion SEK.

COSTS RELATED TO INEFFICIENCY IN DEVELOPMENT PROCESSES

The most comprehensive study of problems related to the quality and result of development processes is the so called Chaos report, performed by the Standish Group (http://www.standishgroup.com/index.php). This classical study, first published in 1997, shows that of more then 5000 large IT development projects in USA only 20% delivered a system according to the initial specifications. About 30% of all projects were complete failures, i.e. they did not deliver any system at all. 50% of all projects delivered approximately 40% of the planned functionality to 1905 of

the planned project budget. A later report indicates that the situation is slightly improved, but shows roughly the same picture. There exist no Swedish study of the same nature, but experiences from a large number of It development projects indicate that the situation is the same here. If we estimate the annual Swedish investments in IT development to be more then 200 billion SEK, only the total failures will add up to more then 50 billion SEK.

The study also shows what are the most important success factors in IT development projects. Among these are active user involvement, focus on usability aspects early in the project and active management support in the requirement specification process.

CONCLUSION

The effects related to different kinds of work environment problems in IT-supported work are significant. Effects concerns e.g. personal well-being, health, work efficiency and work quality. If we restrict monetary calculations to only what can be very conservatory estimated, the figures are still extremely high.

REFERENCES

1. Göransson, B., Lif, M. & Gulliksen, J. (2001) Usability Design – Extending Rational Unified Process with a New Discipline. In proceedings of Design, Specification and Verification of Interactive Systems. DSV-IS 2003, Madeira June 2003.
2. Gould J.D., Boies S.J., Ukelson J. (1997) How to Design Usable Systems. In Helander, M., Landauer, T.K., Prabhu, P. (eds.) Handbook of Human-Computer Interaction, 2nd ed. pp. 231-254, Elsevier Science B.V.
3. Gulliksen J. & Göransson B. (2001) Reengineering the systems development process for user-centered design. In Michitaka Hirose (ed.) Proceedings of INTERACT 2001, Tokyo, Japan. IOS Press
4. Gulliksen J., Göransson B., & Lif M. (2001) A User centered Approach to Object-Oriented User Interface Design. In M. van Harmelen (ed.) Designing Interactive Systems: Object Modeling and User Interface Design. Addison Wesley Longman.
5. Gulliksen, J., Göransson, B. & Lif, M.: A User-Centered Approach to Object-Oriented User Interface Design, in Mark van Harmelen (Red.), Designing Interactive Systems: Object Modeling and User Interface Design, Addison-Wesley, ISBN 0-201-65789-9 (2001)
6. International Organization of Standardization (1998). ISO 9241 Ergonomic requirements for office work with visual display terminals (VDTs) Part 11 – Guidance on usability. (International standard).
7. Leavitt H.J. (1958) Managerial Psychology. University of Chicago Press, Ltd. London.
8. Bergqvist, U. (1993): Health problems during work with visual display terminals, Arbete & Hälsa 1993:28, Arbetsmiljöinstitutet, Solna, Sweden
9. Punnett, L., Bergqvist, U. (1997): Visual display unit work and upper extremity musculoskeletal disorders, Arbete & Hälsa 1997:16, Arbetslivsinstitutet, Solna, Sweden.
10. Sandsjö, L., & Kadefors, R. (eds), (2001): Prevention of muscle disorders in computer users: scientific basis and recommendations. The 2nd Procid symposium, Arbetslivsinstitutet, Göteborg, Sweden.
11. Wigaeus Tornqvist, E., Eriksson, N., Bergqvist, U. (2000): Dator- och kontorsarbetsplatsens fysiska och psykosociala arbetsmiljörisker, (The physical and psychosocial work environment risks at computer- and office workplaces), In: Marklund, S., ed, Arbetsliv och Hälsa 2000, (Working Life and Health 2000), Arbetslivsinstitutet, Solna, Sweden. (in Swedish).
12. Aronsson, G., Dallner, M. & Åborg, C. (1994): Winners and Losers from Computerization: A Study of the Psychosocial Work Conditions and Health of Swedish State Employees. International Journal of Human Computer Interaction, Vol. 6, No 1, 17-37.
13. Åborg, C., Fernström, E., Ericson, M. (2002): Telework – A Longitudinal Study of the Effects on Work Environment and Well being. Uppsala University, Department of Information Technology, Technical report 2002 – 031.
14. Karasek, R. & Theorell, T. (1990): Healthy work: Stress, productivity and the reconstruction of working life, Basic Books, New York, USA.
15. Hagberg, M., A. Toomingas and E. Wigaeus Tornqvist (2002). "Self-reported reduced productivity due to musculoskeletal symptoms: Associations with workplace and individual factors among white collar computer users." J Occup Rehabil 12: 151-62.

From IS Design to Workpractice Construction

Lars Taxén
Campus Norrköping, Linköping University
Department of Science and Technology
Rundan 91, 14645 Tullinge, Sweden
{lars.taxen@telia.com}

ABSTRACT
In this paper we describe an approach for information system design that aims at constructing the social reality in which the system is used. Thus, rather than designing the information system in a given context, the design target is the context itself, including the information system. The expertise knowledge of users and information system designers are jointly utilized in co-constructing the context, which is structured as a particular form of workpractice called the activity domain. In the activity domain, coordinating elements of a practice are integrated into a coherent whole. The theory behind the approach – the Activity Domain Theory – originated in the Ericsson telecommunication company where it has been gradually refined over more than a decade by the author. It has profoundly influenced the coordination of the development of the 3rd generation of mobile systems at Ericsson.

Keywords
IS design, coordination, praxis, co-construction of social reality, shared meaning.

INTRODUCTION

Product developing organizations are facing a turbulent reality today due to increased product complexity, diversification of organizational functions and an ever increasing rate of change. One of the most arduous tasks in these circumstances is to establish a workable, shared meaning among the actors concerning the coordination of development projects [10].

The issue of shared meaning with respect to coordination concerns several aspects. First, there must be a sufficient level of agreement about what should be coordinated and how. Items which are crucial for coordination must be identified, characterized and related to other items. Often, abstract concepts such as "increment" are introduced, something which is particularly difficult to acquire a shared meaning about [6]. Second, the actors may be geographically dispersed, have different roles, come from different traditions, speak different languages, etc. Third, the contents and structure of coordination will change according to new insights, new demands from the market, new tools and methods supporting coordination, etc. Finally, cues used in models and diagrams must make sense to the actors.

The coordination of complex system development projects is only possible with information system (IS) support. In this contribution, we describe an IS design approach that addresses both the technical and social issues as described above. The gist of the approach is to construct *the social reality* in which the IS is used [9]. Thus, rather than designing the IS in a given context, the design target is the context itself, including the IS. The expertise knowledge of users and IS designers are jointly utilized in co-constructing the context, which is structured as a particular form of *workpractice*. A workpractice is a meaningful, goal oriented social entity where some actors produce a result that other actors need [2].

In order to construct the workpractice, it is structured as an *activity domain*. The activity domain is the central construct in a new theory for coordinating human activity – the Activity Domain Theory (ADT) [11]. An activity domain may be regarded as a particular perspective of a workpractice where coordinating elements are emphasized.

DESIGNABLE ELEMENTS OF AN ACTIVITY DOMAIN

According to the ADT the following elements are designable in an activity domain:

The context model

This model signifies the structure and extension of the activity domain. It shows what types of phenomena are considered relevant in the domain, how these are related and how they are characterized in terms of attributes, state sets, revision rules, etc.

The coordination model
The coordination model signifies the dependencies between the activities in the domain. By coordination we understand "[…] managing dependencies between activities" [5, p 90]. This model has the same purpose as ordinary process models.

The transition model
The transition model signifies how different activity domains interact. This model is an elaboration of the Specification Based Data Model suggested by Gandhi & Robertson [1].

The domain core
The domain core is a place-holder for various items which provide stability to the domain. Examples of such items are habits, norms, traditions, rules, routines, domain specific languages, etc.

The running application: the IS supporting coordination
Typical features implemented are support for requirement management, configuration management, test management, project planning and control, etc.

THE CONSTRUCTION STRATEGY

The approach towards constructing the activity domain is called the "domain construction strategy" [10]. The results of the strategy are both intangible and tangible. The intangible form is a shared meaning among the actors about the social reality of coordination. The tangible form consists of domain elements as described above.

The construction strategy requires certain prerequisites. Besides personal and financial resources, management approval, etc., the most important prerequisite is the availability of the IS platform. In the applications at Ericsson the IS platform was Matrix [7]. This system is targeted as a backbone for managing product related data in large, globally distributed organizations. It can be characterized as a high performance, complex system of its own.

In addition, the capacity of the IS platform and the communication network must be secured. This is especially important if the IS is to be used globally. Also, strategies for replication and synchronizing data exchange must be defined and tried out.

The construction strategy is carried out in three phases: *exploration*, *trust boosting* and *expansion* (see Figure 3).

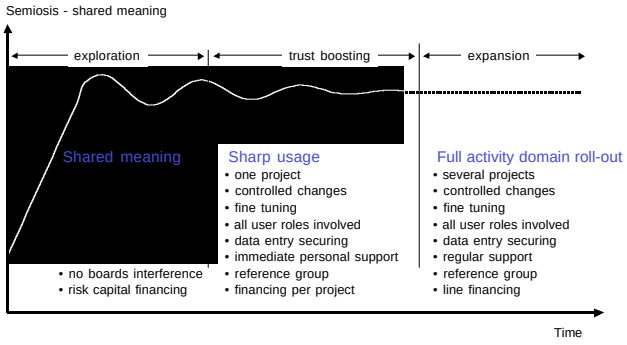

Figure 3: The domain construction strategy

In the first two phases the focus is on establishing the activity domain as a "bridgehead" in one project before expanding it to other projects in the third phase. This means that the gist of the strategy is to quickly establish a relatively stable "seed" domain which is then deployed to other actors in an ongoing domain construction process.

Exploration
In this phase the initial construction of the domain is carried out. The main purpose is to rapidly achieve a tentative consensus about the content and structure of the domain. The driver of the activity is the first project to use the domain. The work is carried out in a "daily build" manner by a small "task force" consisting of target users and IS designers. The work is financed on a risk capital basis.

Trust boosting
The purpose of this phase is to boost the trust about the feasibility of the domain as constructed in the exploration phase. Key issues are getting all actors in the project to trust the data in the IS and to make sure that the performance of the IS is acceptable at all units world-wide. This is done in a sharp project, that is, a project which develops a product for a customer. The task force is still driving the construction. Additional user roles around the project are involved and immediate, personalized support is provided. The construction of the domain in the trust boosting phase progresses by controlled changes. No major reconstruction of the domain is allowed at this stage. Reference groups and steering boards are consulted and the financing is done on a project basis.

Expansion
In this phase several projects are included in the domain. As in the trust boosting phase, the construction is done by controlled changes, however now in a formalized way. The financing is done by the line organization rather than the project organization to keep the domain intact between projects.

An example

As an illustration of the construction strategy we take an example from Ericsson in late 1998 [10]. The task was to

construct a domain for requirement management (RM) in a project developing switching equipments for the 3rd generation of mobile system network.

Traditionally, requirements were stated in requirement specification documents that were stored and managed in large databases. Thus, the document was the item put under revision control. However, this meant that it was not possible to directly trace individual requirements to impacted design items.

With the introduction of modern, object-relational based ISs it became possible to manage each requirement individually. This required the context of RM to be defined in terms of objects, relations, attributes, etc. The task force consisted of users, an IS design expert from the vendor of Matrix and a domain architect (this author). The role of the domain architect was to provide a bridge between the users and the IS design specialist. The users were represented by an experienced requirement manager and the project manager running the project where the new way of managing requirements was to be used for the first time.

The work was carried out as follows. A first version of the context model for RM was suggested, based on the established way of working. Individual requirements were loaded into Matrix from existing requirement specification documents. In a series of meetings the context model was gradually elaborated. Each version of the model was implemented in Matrix. Reports and on-line information were evaluated in the project by the user representatives. If the result was not satisfactory, the context model was changed and implemented anew in Matrix. In order to facilitate the signification process it was important that the model notation was easily understandable by all actors. This was achieved by using a notation based on the Object Modeling Technique (OMT) [8]. Standard drawing tools like PowerPoint were used to describe the model.

An example of the context model from early 1999 is given in Figure 4.

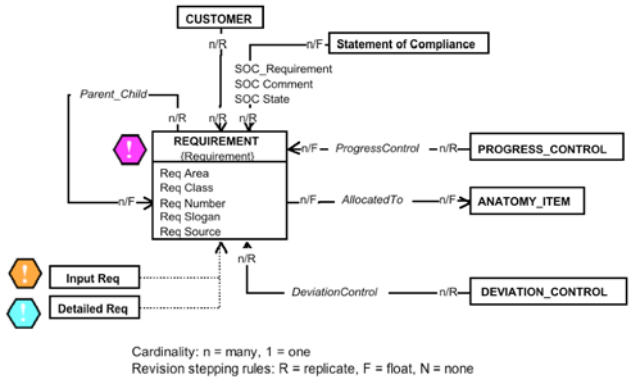

Figure 4: A context model for RM

The major obstacle in the definition of the RM context was to arrive at a shared meaning with respect to:

- Entities, i.e. what phenomena are relevant for RM (signified by boxes in Figure 4).
- Relations between entities.
- Icons signifying entities, names of entities and relations.
- Types and life cycle states of requirements.
- Attributes on requirements and relations.
- Cardinalities on relations, revision stepping rules.
- Actor roles and access rights for roles.

The construction strategy was repeated for other coordination areas until the entire scope of coordination was constructed (see Figure 5, where the RM domain as described above is encircled). During 1999 several hundreds of changes were made in the context model and its corresponding implementation in Matrix.

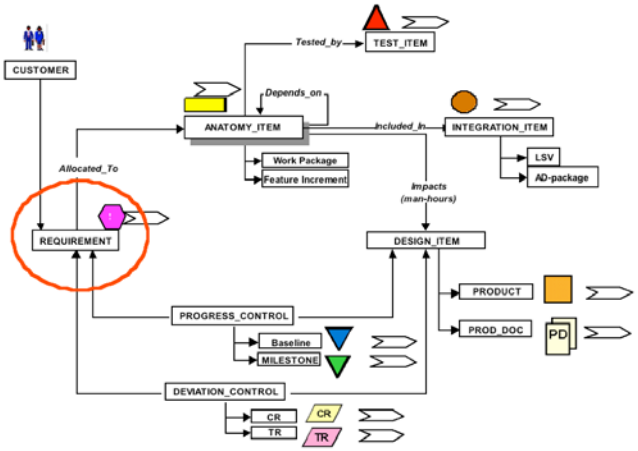

Figure 5: The context model for the entire domain

Conflicts concerning the structure and content of the domain were mostly solved by evaluating the usefulness of the domain in practice. If a certain construct worked satisfactory, it became gradually established as a good

way of working. However, if a working consensus could not be achieved, the sponsor of the construction activity, i.e. the project manager, decided how to proceed.

From a learning point of view, the approach can be characterized as *experiential learning* [4]. By constantly iterating between reflection over the models and trying their IS implementation out in action, a gradual, shared understanding emerged simultaneously with the evolution of the models and the IS design.

RESULTS

The domain construction strategy began to influence the Ericsson practice around 1997 with the introduction of a method package for incremental development of large software systems. The first sharp project to use Matrix was carried out in 1998. Between May 1999 and mid 2002 the number of projects using the strategy rose to around 140 distributed over more than 20 development sites worldwide. During this period four domains were constructed. As indicated in the following statement the impacts on the Ericsson practice were profound:

> "Especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have had to maintain that manually, that would have been a sheer disaster. [...] we had some, only in my part of the project, some 200 work packages or work packages groups or whatever you want to call them, deliveries, on the average 2-5 subprojects within them 5-10 blocks being delivered, just keeping track of that [...] would have been a hell of a job." (Project manager, 3G development)

It is beyond the scope of this paper to give a full account of the impacts. This is reported in [10].

DISCUSSION

Early IS design methods concentrated on the technical aspect of the IS [3]. A clear separation was made between users and designers. Largely influenced by Scandinavian researchers, the use context of the IS became more pronounced in design approaches such as the socio-technical, the trade-unionist, the language action, the professional work practice approach and others [ibid]. However, in all these approaches, the IS was still the target of design.

The suggested approach in this paper means that we are opening up a new line of investigation into IS design. The main target of design is no longer the IS but the entire workpractice in which the IS is used. This means that all actors performing coordination acts are contributing to the IS design, some more, some less. The users are one of several groups of actors participating in the co-construction of the workpractice.

The basic mode of design in the approach is an ongoing interaction between reflection and action. Thus, the approach does not follow the traditional phases of requirement analysis, design, implementation, testing and deployment. This means that the approach can be characterized as an evolutionary type of IS design method [3]. A similar approach is suggested by Truex et al. [12].

TRANSFERABILITY OF THE RESULTS

So far, the suggested approach has been proven operational in one area – the coordination of extremely complex system development tasks at Ericsson. However, Ericsson can be seen as a paradigmatic example of the very turbulent situation that product developing organizations are facing today. Thus, it is reasonable to expect that the approach is transferable other organizations. The applicability of the approach to other areas than coordination, however, is a matter for future research.

CONCLUSION

In this paper, we have described an approach for IS design based on the Activity Domain Theory. The experiences show that the proposed approach enables the design of IT artifacts which can support the coordination of very complex system development tasks while taking individual, social and technical aspects into consideration.

REFERENCES

1. Gandhi M, Robertson E L, A Specification-based Data Model. *Indiana University Computer Science Department Technical Report TR344*, Indiana University, Indiana. http://www.cs.indiana.edu/ftp/techreports/index.html
2. Goldkuhl G, Röstlinger A, The significance of workpractice diagnosis: Socio-pragmatic ontology and epistemology of change analysis, *The International workshop on Action in Language, Organisations and Information Systems (ALOIS-2003*), Linköping University, 2003.
3. Iivari J, Lyytinen, K, Research on Information Systems Development in Scandinavia - Unity in Plurality, *Scandinavian Journal of Information Systems*, 10 (1&2) (1998), 135-186.
4. Kolb D A, *Experiential Learning: Experience as the Source of Learning and Development*. New Jersey: Prentice Hall, Englewood Cliffs, 1984.
5. Malone T, Crowston K, The Interdisciplinary Study of Coordination, *ACM Computing Services*, Vol. 26, no 1 (1994), 87-119.

6. March J G, Simon H A, *Organizations.* Second edition, Cambridge, Massachusetts, USA: Blackwell Publishers, 1958.

7. Matrix-One (2004) http://www.matrixone.com/ (April 2005).

8. Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W, *Object-Oriented Modeling and Design*, New Jersey: Prentice-Hall International, Inc., 1991.

9. Searle J R, *The construction of social reality*, London: Allen Lane, 1995.

10. Taxén L, *A Framework for the Coordination of Complex Systems' Development*. Dissertation No. 800. Linköping University, Dep. of Computer &Information Science, 2003. Available at http://www.ep.liu.se/diss/science_technology/08/00/index.html (April 2005)

11. Taxén L, Articulating Coordination of Human Activity - the Activity Domain Theory. In *Proceedings of the 2nd International workshop on Action in Language, Organisations and Information Systems (ALOIS-2004)*, (2004), Linköping University. Available at http://www.vits.org/konferenser/alois2004/proceedings.asp (March 2004)

12. Truex D P, Baskerville R, Klein H, Growing Systems in Emergent Organizations, *Communication of the ACM*, Vol. 42, No. 8 (1999), 117-123.

Approaches for inclusion of usability and accessibility in ICT procurements

Clas Thorén
Statskontoret
Box 8110, 104 20 Stockholm, Sweden
clas.thoren@statskontoret.se

ABSTRACT
This article discusses the need for inclusion of usability and accessibility requirements in public procurements of ICT products and services as a response to EU eInclusion measures. Accessibility criteria exist, but there is a lack of usability criteria. The approach of Statskontoret, the Swedish Agency for Public Management, is presented. The possibility to refer to supplier's declarations is discussed as a future possible approach.

Author Keywords
Accessibility, usability, procurement, suppliers declaration.

INTRODUCTION
The European Commission has, within the concept of eInclusion, identified public procurement as an instrument to increase accessibility to the information society by people with disabilities and older persons. This is supported by a new directive on public procurement, where it is stated that technical specifications shall be set out in the contract documentation, and that "whenever possible, these technical specifications should be defined so as to take into account accessibility criteria for people with disabilities or design for all users". For public procurers of ICT products and services, this raises the question of how to define accessibility criteria that can be included in requests for tender.

Accessibility is closely related to the concept of usability. This means that the purchaser has a good reason to consider also usability when defining the criteria for awarding of the contract. In addition, the purchaser may have heard about the concept of Design for all and understood that this concept overlaps or maybe combines usability and accessibility. This gives rise to (at least) two questions:

- What is the difference between usability, accessibility and design for all? Are they three different criteria for awarding of contract?
- How should requirements for usability, accessibility and design for all be formulated in order to be compliant with the legislation on public procurement?

DIFFERENT CONCEPTS?
Firstly, let us look at the issue of the differences. Usability is defined by ISO 9241 [1] 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". The ISO TS 16071 [2] defines accessibility as „the usability of a product, service, environment or facility by people with the widest range of capabilities". Design for all (or Universal Design, the term used in USA) have many similar definitions, one is by G. Vanderheiden: „the process of creating products (devices, environments, systems and processes) which are usable by people with the widest possible range of abilities, operating within the widest possible range of situations (environments, conditions and circumstances)" [3]. Moreover, there are other concepts similar to Design for all, such as Inclusive Design and Accessible Design. This article does not intend to discuss these definitions, but it is obvious that they are overlapping. The concepts have a rather wide least common denominator, if we for a moment disregard that design for all is defined as a process. This suggests that it is probably possible to make a synthesis of the three concepts into one single concept, which would be welcomed by purchasers.

Suppose that a public purchaser has accepted that requirements on what most purchasers will call user-friendliness, i.e. usability, accessibility, Universal Design etc., are justified for inclusion in calls-for-tender. Then the purchaser is faced with the problem of if and how these concepts should be separated into two or more criteria for awarding of contract. The purchaser notices that usability occurs when using a product, and is thus not an inherent quality that is objectively observable. Accessibility, on the other hand, seem to be an inherent quality, at least when the purchaser judges from available guidelines. Universal Design and its synonyms seem sometimes to be a process, sometimes a quality. In addition, sometimes the concepts refer to all people, sometimes to as many as possible, sometimes to disabled people. This is confusing for the purchaser.

The life of public purchasers, and suppliers, of ICT equipment for public electronic services would be easier if there were one single definition which

- cover qualities that are objectively measurable and observable during the procurement phase;
- is not context dependent;
- cover as many users as is reasonably possible without causing higher costs („readily achievable", to use the term in the U.S. ADA legislation [4]).

The purchaser would therefore want this single definition to be something like *„the extent to which a product, service, environment or facility is designed to be used for its intended purpose with effectiveness, efficiency and satisfaction by people with the widest possible range of capabilities, irrespective of factors such as gender, age, disability or ethnical background"*. Whether this is called usability, Universal Design or whatever is of less importance for the purchaser; what is important is that it reflects the user needs and that its meaning can be communicated to the suppliers. Such a definition has many advantages for a purchaser: „to the extent" indicates the possibility to rank products. „For its intended purpose" corresponds to context of use, but focuses on the product, which is what the purchaser is interested in. „Widest possible" covers reasonably all citizens. „Gender, age …" indicates that the concept includes particular concerns, but not restricted to people with disabilities. „Effectiveness" and „efficiency" means that it is not only a matter of user-friendliness. In addition, this definition has no „specified users".

The definition of accessibility in ISO TS 16071 integrates accessibility into usability. This is favourable for software development: methods for ensuring that usability, as defined in ISO 9241, is taken into account will automatically include accessibility considerations. For the procurement process however, this means that accessibility will not anymore be an inherent quality. It will become observable only during interaction between the user and the product. The advantage is that the purchaser need to deal with only one concept; the disadvantage is that there will be no set of properties, independent of users and context of use, that can be labeled „accessibility criteria".

For the foreseeable future, the best approach for the purchaser of ICT seem to be to divide the user-friendliness into two parts:

- Usability, as defined in ISO 9241, and generally perceived as dealing with ease-of-use, ease-of-learning and ease-of –understanding etc.
- Accessibility, regarded as a set of properties that allows a product to be used by people within the widest possible range of capabilities. Basically, Design for all and its equivalences could be regarded as processes leading to accessibility.

Statskontoret, the Swedish Agency for Public Management, carries out procurements of ICT resulting in framework agreements that can be used by administrations in central and local government. In the calls for proposal, we state requirements on usability and accessibility under one headline: usability. Our approach is to regard accessibility as included in usability. This does not mean that accessibility has disappeared as a concept. Usability requirements and accessibility requirements come from different sources. Some requirements typically concern the needs of disabled people; they are derived from published accessibility guidelines. In relation to the e-government efforts, we communicate a message to ICT consultancies who want to offer their services to the government: we expect you to have competence in usability and accessibility, and we recommend you to integrate them, i.e. not regard accessibility as a niche competence. We wish to avoid a situation where two consultancies are needed, one for usability and one for accessibility.

PUBLIC VS PROFESSIONAL USE

The intended users of the subject of a public procurement can be employees or members of the general public (=citizens). Should the requirements on usability and accessibility be different for these user groups? It seems reasonable to argue that accessibility requirements are more important for citizens, since the purchased system has to cater for a wide set of user abilities and disabilities. Employees, on the other hand, are known and consequently most of their (dis)abilities. However, since current employees might acquire, and future employees might have, any kind of disability during the lifetime of the purchased system, the accessibility requirements are not less important where the users are employees of the contracting authority. The purchaser should not disregard any kind of disabilities.

As regards usability, do employees and citizens have different requirements? The public purchaser will equal the usability elements "efficiency" and "effectiveness" to factors such as functionality, performance, capacity, while the element "satisfaction" will be considered equivalent to ease-of-use, user-friendliness etc. This means that, in the request for proposal, requirements on effectiveness and efficiency are taken account of under other names, while satisfaction unfortunately tend to be disregarded. Public purchasers are faced with the problem of setting weights on requirements and might therefore ask if the elements of usability have different importance depending on e.g. frequency of use or for professionals and non-professionals. Is, for example, ease-of-use of a word processor more important for a professional handbook author than for office workers at large? Is learnability more important than effectiveness for a seldom-user of a certain public e-service? Usability experts seem to argue that there is no clear answer to such questions. The elements of usability are interdependent and can not be ranked, and whether something is more usable for one user group than another

depends on the context of use. The author have discussed these issues briefly with a few usability experts and would welcome further discussion, hopefully leading to some guidance on generic contexts-of-use where certain aspects of usability could be identified as more important than other.

NEED FOR USABILITY CRITERIA

The concept of usability imposes a problem for the purchaser. Usability occurs during the usage, i.e. after the procurement phase is finalized. This is why there are no guidelines providing usability requirements for different products to be used in procurements. Theoretically, it would be possible to apply an expert evaluation in order to evaluate offered products with respect to usability. An expert on human-computer interaction could, by a hands-on exercise, assess the usability of the offered products and set scores. From a public procurement point of view, there are two problems with this approach:

- Depending on the number of tenders, this might be very time-consuming.
- The evaluation is basically subjective, not objective, even if the qualities to be assessed are stated in advance and known to the tenderers. This is contradictory to two important principles of public procurement:
  - A procurement has to be, in a sense, predictable by the tenderers.
  - Tenderers must be treated equally.

Usability experts have suggested to Statskontoret that a more successful way than establishing usability criteria for goods and services could be to require that the supplier's development process, competence and organization include usability. As a response, Statskontoret has since 2003 cooperated with CID and UsersAward [5], on identifying a way of setting usability requirements in our calls-for-tender. UsersAward is a Swedish collaboration project between LO, the major trade union in Sweden, and CID. Application software manufacturers are invited to make a declaration of how their product satisfies a set of requirements specified by UsersAward. Representatives of UsersAward then interview users of the product. The Award is given to the product in combination with a specific implementation (i.e. „Product Name" at „Installation Site") and is given provided that a specified user satisfaction score is exceeded.

We had a hypothesis that it would be possible to find a mini version of the CID/UsersAward method that could fit into the procurement process of Statskontoret. A procurement of off-the-shelf software, for framework agreements with a number of suppliers, was selected as a pilot procurement. Four requirements were defined for the supplier appraisal phase of the procurement, and eight requirements for support, training and other services.

The evaluation of the tenders show that few suppliers work with usability in a professional and systematic way. The customer, not the end user, is the target despite that the impact of the offered products occur at the end user. Increasingly sharpened requirements on policies, methods, organization and competence might in the long run result in better products with less frustration, decreased usage costs and better productivity.

THE SUPPLIERS' DECLARATION APPROACH

Both purchasers and suppliers would benefit from a system where the requirements on and evaluation of tenders are based on suppliers' declaration of usability, i.e. the tenderer provides a certificate showing that an offered product satisfies a specified set of criteria. Such criteria could be process oriented and/or product specific. For example, a certificate could state that

- (Product Name) is developed by a method compliant to ISO 13407 [6];
- (Software package Name, version X.Y) got an average score of Z in a (Name of a recognized usability evaluation method) test.

The set of criteria could be defined and made publicly available by

- The purchaser in the call-for-tender;
- The tenderer (this is equivalent to a proprietary standard);
- An industry association (this is equivalent to a defacto standard);
- An organisation where both users and suppliers are participating, such as formal standard organisation.

The more widely recognized the set of criteria is, the better is its likelihood to be accepted by purchasers and suppliers in general. The alternatives above are listed in decreasing expected acceptance order.

The party issuing the certificate could be a recognized third party (this is what purchasers prefer) or the supplier.

For example, and given the lack of universally recognized usability criteria, it could be envisaged that a product with a certificate issued by a third party and based on a formal standard will get a higher score in the evaluation of tenders than a certificate issued by the tenderer and based on his proprietary criteria.

Some standards have been developed, and other are being developed, that deal with ICT usability and accessibility criteria and certification schemes. For example:

- The WAI Guidelines on web accessibility, WCAG 1.0, consist of criteria for accessible web sites.
- A CEN Workshop has recently started, aiming at defining a certification scheme for web accessibility.

- ISO/TS 16071 defines software accessibility.
- The European Commission is preparing a mandate on accessibility requirements to be used in public ICT procurements.
- ISO/TR 18529 [7] can be used by suppliers for self-evaluation of their ability to perform a user-centered development process.
- ISO 23025 (not yet publicly available) will define a report format for suppliers' declaration of software usability.
- ISO 20282 (not yet publicly available) will define a report format for suppliers' declaration of the usability of consumer products.

These efforts pave the way to a future situation, where purchasers require conformance to criteria oriented standards, and suppliers give proof of conformance by means of standardized certification schemes. Many of the standards require expertise in usability, which should increase the job opportunities for usability students.

CONCLUSION

To summarize and conclude: Public procurement staff is faced with some problems, at the same time raising challenges for the HCI community:

The many different but similar and overlapping concepts in the area of usability and accessibility is confusing for the purchasers. They would benefit from one single operable definition that merges the concepts into one. In the lack of such a definition, the best approach for the purchasers seem to be to distinguish between the interaction oriented usability on one hand and the criteria oriented accessibility with its equivalences Design for All, Inclusive Design etc. on the other hand.

The legislation on public procurement imply that there is a need for clear, well-defined, understandable criteria for accessibility and usability of various products and services. Accessibility criteria exist, usability criteria do not.

Given the lack of criteria, the emergence of suppliers' declarations offer a possible way of introducing usability requirements in calls-for-tender. There is a need for widely recognized regimes for development and management of such declarations of different kinds of products. Much work is going on; still more work will be welcomed by purchasers.

REFERENCES

1) ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Available via national standards organizations.

2) ISO TS 16071:2003 Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces. Available via national standards organizations.

3) quoted in Nicolle, C., Abascal, J. (editors) (2001) Inclusive Design Guidelines for HCI, page 3.

4) http://www.usdoj.gov/crt/ada/adahom1.htm

5) http://www.usersaward.se/

6) ISO 13407:1999 Human-centred design processes for interactive systems. Available via national standards organizations.

7) ISO/TR 18529 Ergonomics – Ergonomics of human-system interaction – Human-centered lifecycle process descriptions. Available via national standards organizations.

Buying Usable – the User-Centred Procurement Process

Magnus Lif, Bengt Göransson & Torsten Sandbäck
Guide Redina AB
Smedsgränd 9, SE-753 20 Uppsala, Sweden
{firstname.lastname}@guide.se

ABSTRACT
In Sweden, framework agreements can be used to simplify the purchasing process for the public sector. When carrying out the procurements leading to framework agreements, the responsible agency, Swedish Agency for Public Management (SAPM), is not able to compare the products/services usability. Instead, a guide has been published on how the contracting authorities utilizing the framework agreement could take usability into account when selecting products/services. The User-Centred Procurement Process (UCPP) described in the guide contains sex steps. It is aimed at different people in the organisation involved in buying products/services, i.e. not necessarily usability experts. The intention with the guide is both to create an awareness of the importance of usability and to provide a "tool" for people in the organisations to buy more usable products/services. The guide in itself has been developed with a strong user focus. It has been reviewed by both potential users of the guide and by different usability experts. The feedback gained has been a valuable input in our work.

Author Keywords
User-centred, usability, procurement, public sector

INTRODUCTION
An important task for the Swedish Agency for Public Management is to carry out procurements leading to framework agreements on IT products and services. The framework agreements can be used by other agencies by simply applying the terms laid down in the agreement. A framework agreement on IT products and services specifies functionality, performance, delivery conditions and prices for those. When procuring products comparing these characteristics is of course important. However SAPM has realised that the usability of a product or service can have a substantial impact on the users' health as well as on the costs involved when using the product or service.

Today the usability of products is not specified in the framework agreement, one reason being that an agreement covers a very large set of contexts of use, which to some extent are not known. A certain product can be very effective for one user performing a certain task, but another user performing the same task may find the product very difficult to use. A typical example is a Content Management System. For a novice user it is likely that the application is cumbersome to use, but for someone using it on a daily basis it may be very effective. This implies that a product can not be usable per se; it has to be evaluated while in use. In the definition of usability in the International Standard ISO/IS 9241-11 [4] this has been taken into account:

> *"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."*

The users, their tasks, and the context in which the product is going to be used are only known to the buying organization. Therefore, the usability evaluation has to be carried out in that known situation. Because of this, SAPM recently identified a need to assist contracting authorities to take usability into account when making a purchase decision based on a framework agreement. The goal was to produce a guide [7] for how to evaluate the usability of a product or service when purchasing from a framework agreement.

The target audience for the guide would be anyone involved in procuring products or services from a framework agreement. Typically it could be the IT-manager or the acquisition manager. It is not likely that the people making these decisions are very familiar with usability. However, they could be part of a large agency with their own usability experts or they could have experience of working with external usability consultants.

Therefore, the guide had to be written in a way so that it can be useful for both usability experts and for those who are interested in the subject but have only limited experience. An important aim was to get the agencies to at least take

usability into account when procuring products or services, not ignoring it completely.

Consultants from Guide Redina AB were hired to provide the necessary usability expertise for this work.

User-Centred Design – UCD

Involving users in the development process has shown to be an important step in making products or services more usable. An early, and well known project, is the 1984 Olympic Message System, where the aim was to design a usable system and test a user-centred methodology [2]. It is a success story both regarding the usability of the product and the methodology used. Similar attempts were made in Scandinavia by for instance Bjerknes, Ehn and Kyng [1]. What has come to be known as the Scandinavian school emphasise the importance of having users participating in the development process on equal terms as developers. The legal and democratic rights to influence and control their work situation are very important. Greenbaum & Kyng [3] coined the term cooperative design as they summarised and elaborated on the Scandinavian approach. In North America the term participatory design is commonly used for the theories based on this approach. None of these very important landmarks in user-centred design have become widely adopted. Contextual design [9] is probably the one that is mostly spread. In parallel to UCD, usability engineering has evolved, offering a more pragmatic view on developing usable systems [6]. The process has adopted principals from software engineering and offers a range of techniques for analysing users, establishing usability goals, evaluating design, etc. Today there is a standard covering some of the issues, ISO/IS 13407 Human-centred design processes for interactive systems [5].

Initiatives in Sweden by various interest groups in the public sector, like the UsersAward, maintain the long Swedish tradition of focusing technical development on the solutions that are the most appropriate for the employees as well as for the employers. UsersAward has its roots in the Swedish Trade Unions and promotes that "employees should take an active part in the long-term as well as the short-term development of the organisation" [8].

You could say that this initiative by SAPM to develop a guide for procuring usable product/services is another example of a contribution to this tradition. The efforts that are being made to ensure a high level of usability in products/services and a good working environment shows how strong this tradition is.

Based on our own and other peoples experience we find it necessary to work in a user-centred way to produce usable systems. So, when working on this guide we developed a User-Centred Procurement Process. The main difference between this process and the processes discussed earlier is that it does not support the *development* of software. Instead it supports the procurement of already existing products and services such as printers, mobile phones and content management systems. Nevertheless, involving users in this process is necessary to make sure that the selected product or service meets the users' needs.

THE GUIDE

The actual guide is a booklet that can be downloaded for free from SAPM´s website [7]. The guide gives an introduction to usability, accessibility and user-centred design. The core of the guide is the User-Centred Procurement Process. This is followed by some examples showing how to use the process when buying different products. A number of methods for user studies and usability evaluations are also described.

The User-Centred Procurement Process

In the User-Centred Procurement Process we are only focusing on the steps involved when making the actual purchase decision. This is of course part of a larger process that involves improving the users' work environment and tools, which in itself should be a user-centred process including for instance deployment and monitoring. The process consists of the following steps:

A. Need identified

B. Specify user profiles

C. Select products/services to evaluate

D. Evaluate the usability

E. Describe possible remedies

F. Purchase product/service (or not)

A. Need identified

When the process starts the actual need to invest in a product or service is already identified. This, in itself, could be the result of for instance a pre-study, an organisational change or because the current product no longer sufficiently supports the business.

B. Specify user profiles

Understanding the users and their needs is crucial when deciding what product or service to select. From the definition of usability we know that we need to understand who the users are, what goals they have, what they do and the context of use. Therefore, in this step the knowledge about the users are specified as user profiles. This will be an important input when selecting products to evaluate and also for the actual usability evaluations.

A user profile consists of the following:

– A description of the different user groups that will be using the product/service.

– A description of the context of use.

– A description of the tasks performed by the users.

– A description of each user group's usage goals.

Involving the users in this process will not only reduce the risk of making the wrong decision but will also help when deploying the new product or service.

In the guide a number of methods are presented for how to conduct user studies. Some methods should ideally be used by more experienced usability staff but also people with less experience should find them useful.

C. Select products/services to evaluate

Purchasing from a framework agreement can be done in two different ways. One way is to first define the needs and then compare the product specifications in the framework agreements and select the most appropriate product. The second way is to send a written request for proposal to all suppliers having a framework agreement for the actual product type. The products/services are evaluated on the basis of the responses from the suppliers.

D. Evaluate the usability

Suppliers are contractually obliged to provide the possibility for the client to test and evaluate the product/service.

In the guide a selection of methods are described. Some of them, such as scenario based evaluation, require the participation of users. When deciding what evaluation method to use it is necessary to consider for instance how critical the usability of the product is, time, budget, access to users and usability expertise. Typically an evaluation renders a prioritized list of potential usability problems.

When evaluating the usability of a product or service, it is of course preferable to involve the potential end users. In a lot of cases the products/services that are purchased will become an integrated part of their work which means that they should be actively involved the procurement process.

E. Describe possible remedies

In some cases it is possible to make certain adjustments to compensate for the identified problems. This does not include adding or changing the product's functionality, but could involve for instance training, support material and changes in the organization. However, it is important to understand that making such changes is risky since they may give rise to new problems.

F. Purchase product/service (or not)

When the usability evaluation is finished it is time to make a decision. The usability of the product/service will only be one factor to consider, there are several other criteria that needs to be taken into account as well. The client should now be able to choose one product/service or decide not to select any of them.

Examples

The guide also includes some examples showing how the process can be used when purchasing different kinds of products/services. When buying a laser printer a less thorough approach is described then when purchasing a content management system.

WORKING METHOD AND FEEDBACK GAINED

When we started to develop the guide our aim was to be as user centred as possible. A team with the responsible person from SAPM and the usability consultants was formed. Given the fairly limited budget most user input was gathered by the key person from SAPM, however, in addition a reference group was set-up. This reference group represented a large number of authorities, some having usability knowledge in-house and some not. Also, some other key people, e.g. accessibility experts, were interviewed. The reference group was at an early stage asked to comment on a draft version of the guide. Among the important point of views gathered from the reference group and the interviews were these:

- Do not separate accessibility from usability. Accessibility should be treated as a part of usability.
- Keep the guide itself simple – "easy to use".
- Support different levels of ambition, e.g. the usability of mobile phones are usually not as crucial for the business as it is for a case handling system.
- Support different levels of usability maturity. Some organizations have their own usability specialists, while some do not even understand the concept.
- Some of the respondents were very enthusiastic about the initiative, and hoped that the guide would not only help the procurement process, but would also help in promoting usability and improving the quality of work as such.

We used this important feedback and revised the guide. However, to make sure we were not heading in the wrong direction and that we had not forgotten anything important, we decided to seek additional help from other experts. We conducted a workshop with a group of specialists in evaluating usability, requirements engineering and procurement. The participants got the revised version of the guide in advance and prepared themselves for the workshop. The outcome of this workshop was very important for our remaining work. It affected the content as well as the structure of the guide. Major points made during the workshop were:

- The guide was to "heavy" and difficult to read. It should be completely reorganized to become easier to use.
- The method descriptions should be toned down, and some should not be included at all since they would require very skilled usability professionals.
- Emphasise should be on the practical examples and on achievement that can be made with little effort.

- We should assume that most readers are not that familiar with usability.

After analysing the results from the workshop, we totally revised the guide and re-wrote large parts of it. For example, we almost turned the guide "upside down". In the earlier version the guide started with a rich description of what usability and user-centeredness is about, followed by descriptions of a number of analysis and evaluation methods. In the published guide, all of that is at the end of the document. Instead the guide starts with a short introduction pointing out the objectives of the guide, and then it goes straight to the point with "checklist-like" chapters about how to consider usability using the user-centred procurement process. The richer descriptions of different methods for analysis and evaluations are "pushed" to the end of the guide, assuming that only a minority of the users will read and comprehend that content. We have also more explicitly introduced three ambition levels for using the guide by giving examples: buying a laser printer, buying mobile phones or investing in a content management system. This helped very much in structuring the content and made the guide easier to grasp, and in the end to use.

CONCLUSIONS

The guide has just been released and will now hopefully be used in a number of purchases. It will be interesting to follow these processes and evaluate the use of the guide. As indicated by the reference group we do also have expectations for the guide to be of value for organisations when it comes to selling and promoting usability. It might be a useful "tool" for usability champions to use in their own organisations as well as in contract situations.

Further, used in the right way, it can be a powerful and effective instrument for the employees. The user-centred procurement process described and the way of analysing and evaluating the usability outlined in the guide, aims at involving and empowering the users, in this case equal to the employees. In the long run, we anticipate this kind of initiatives from the authorities to have a major impact on the quality of work for employees, not only for the public sector but also for the industry. It will certainly put some pressure on the suppliers of IT-systems as well as on those developing consumer products. But, it cannot be just a one-way communication. The two parts, the procurer and the supplier, must both share the same interest to actively involve users, and make the necessary resources available for this to happen.

REFERENCES

11. Bjerknes, G., Ehn, P. & Kyng, M. (eds.) (1987). Computers and Democracy: A Scandinavian Challenge. Aldershot, UK, Avebury.
12. Gould, J.D., Boies, S.J., Levy, S., Richards, J.T. & Schoonard J. (1987). The 1984 Olympic Message System: A test of behavioral principles of system design. In: Communications of the ACM, Vol. 30 No. 9, pp. 758-769.
13. Greenbaum, J. & Kyng, M. (eds.) (1991). Design at Work: Cooperative Design of Computer Systems. Hillsdale, NJ, Lawrence Erlbaum Associates.
14. ISO 9241-11 (1998). Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. First edition 1998-03-15, ref. number ISO 9241–11:1998(E), International Organization for Standardization, Geneve.
15. ISO 13407 (1999). Human-centred design processes for interactive systems. International Organization for Standardization, Geneve.
16. Nielsen, J. (1993). Usability Engineering, Academic Press Inc., San Diego.
17. Statskontoret (2005). Avropa användbart. Vägledning för bedömning av användbarhet vid avrop från ramavtal. [In Swedish]. (Publication 2005:6). Statskontoret. Stockholm, Sweden. http://www.statskontoret.se.
18. UsersAward (2005). http://www.usersaward.se/. Visited April 27, 2005.
19. Wixon, D. & Holtzblatt, K. (1990). Contextual Design: An Emergent View of System Design. In: The proceedings of the CHI'90 conference

Ergonomic Customizing of SAP-Software: Results from two Studies

Stefanie Floegel
Bit e.V.
Unterstr. 51, D-44892 Bochum
stefanie.floegel@bit-bochum.de

Reinhard Linz
Bit e.V.
Bonner Talweg 33, D-53113 Bonn
reinhard.linz@bit-bochum.de

Jochen Prümper
FHTW-Berlin
Treskowallee 8, D-10318 Berlin
j.pruemper@fhtw-berlin.de

ABSTRACT
Users have evaluated the ergonomic quality of SAP-HR software installations as unsatisfactory due to deficiencies in the usefulness and usability of SAP-systems. These ergonomic deficiencies can be resolved supplementary ergonomic customizing, i.e. fine-tuning the system using a number of inbuilt "adjusting screws". However, a better solution is to apply the concept of integrated ergonomic customizing (IEC). IEC enables many ergonomic usability demands on the software to be identified early enough to be included in the system design during the introductory phase, resulting in a system that is user-friendly from the outset, more readily accepted, and easier to learn for later users, while avoiding costly subsequent changes due to functions that are either missing, faulty or inefficient. The following preconditions are essential for successfully integrating ergonomic customizing into the introduction of SAP-systems: a) the availability of practical procedures and suitable methods of investigation, prototyping and evaluation, b) the qualification of SAP-consultants in software-ergonomics and ergonomic customizing, and c) creating greater awareness of the concept of software-ergonomics amongst managers, project coordinators and system experts across the company.

Author Keywords
Ergonomic Customizing, SAP, Adjusting Screws, ISO 9241-10

INTRODUCTION
SAP Inc is one of the largest producers of Enterprise Resource Planning (ERP) Software. There are currently around 84,000 installations of SAP software in over 24,000 client sites spanning more than 120 countries, through which the SAP-system touches the daily lives of hundreds of thousands of workers worldwide. However, how many of these users have the opportunity to work effectively and efficiently with the SAP system, and do they actually enjoy working with SAP software? Is it possible for clients to adapt SAP software on-site to improve the ergonomic quality of the system? And if so, what procedures, instruments and methods are most suitable for the work place? These are among the questions that the two studies entitled "Ergusto" (Ergonomic Customizing of SAP) and "ErgoCust" (Integrated Ergonomic Customizing) attempted to answer[8].

ERGUSTO
The aims of the Ergusto-Project were

- To find out how SAP HR (Human Resources) installations differ ergonomically for personnel administrators;
- To develop procedures for the ergonomic customizing of productive SAP-systems which are already in use, and
- To use the ergonomic adjustment options within the SAP HR-system to adjust and improve their ergonomic quality.

[8] Ergusto and ErgoCust are two shared projects of three institutes: bao – Büro für Arbeits- und Organisations-psychologie GmbH in Berlin (under Jörn Hurtienne, Anne Jansen, Cornelius Müller), BIT – Berufsforschungs- und Beratungsinstitut für interdisziplinäre Technikgestaltung e.V. in Bochum (under Petra Abele, Stefanie Floegel, Reinhard Linz) and TBS – Technologieberatungsstelle beim DGB NRW e.V. in Oberhausen (under Bernd Stein). The projects are jointly financed by the North-Rhine-Westphalian Ministry for Economy and Labor and the European Union and are supported by SAP® as a dialogue partner.

To achieve these aims, three tests were applied to the SAP HR module of nine companies working with SAP. These were *analysis, customizing and qualification,* and *check up*.

Analysis

Three steps were employed to achieve a reasonably complete picture of how the clients currently use SAP-HR. Firstly, all SAP users filled in a *questionnaire* covering the ergonomic quality of the SAP software used to perform their tasks (ISONORM 9241/10, [1]), knowledge of user-specific "adjusting screws", levels of support for SAP users through the system, and the degree of participation in the introduction of SAP systems. Next, the work of three or four SAP users from each company was analysed in depth using *observational interviews* [2] lasting half a day each. On-screen work was videotaped and analysed [3]. Lastly, results from the questionnaires and details of the functionality and usability problems [4,5] identified in the interviews and video analyses were reported back to a *focus group* [6] within the company that categorised and weighted them according to their severity.

Customizing & Qualification

The list of deficiencies identified by the analysis became the basis of concrete interventions during the *Customizing and Qualification* phase. Adjustments to the system were prepared and introduced, while users and system experts underwent further training. Various system settings were adjusted by system experts through the use of so-called software-ergonomic "adjusting screws". These adjustments were tested and documented, and finally implemented into the 'live' system. IT staff and system users were trained in parallel in how these adaptations would impact on daily SAP-system work processes.

Check Up

All participants in the project (users, system experts and decision makers) then completed another questionnaire to measure how successful these changes were at enhancing the quality of work, and where there remained room for improvement.

RESULTS

This paper highlights three areas of *ergonomic customizing* from amongst the numerous results of the study[9].

Ergonomic Quality of Software

The overall scores of the ISONORM 9241/10 questionnaire from the nine companies are depicted in figure 1. The minimum limit of the software-ergonomic standard is marked by the line at +1 [1]. Two things are remarkable. Firstly, the software-ergonomic quality of the SAP R/3-HR Module in question varies notably from company to company and secondly, only one company reaches the minimum standard (+1) for satisfying software.

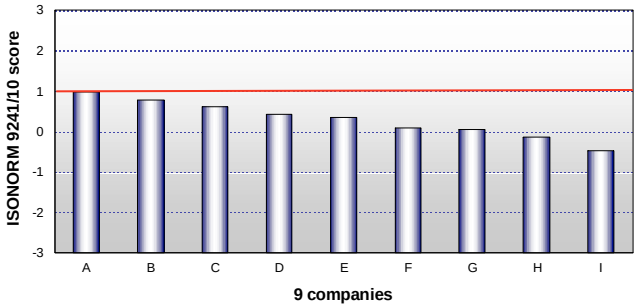

Figure 1: Ergonomic quality of SAP R/3 HR software in 9 companies (N=105)

These results arise from three factors: 1. from deployment processes neglecting software-ergonomic requirements, 2. from lacking knowledge of administrators and users about possibilities for optimisation and individualisation and 3. from different degrees of user-participation in the deployment project and the improvement process [7].

Deficiencies in Software Ergonomics

The observational interviews and video analyses highlighted hundreds of deficiencies and other findings in the software ergonomics. To facilitate the systematic resolution of these deficiencies, they were classified into three main categories; *deficiencies of the SAP-system*, *organizational deficiencies* and *user-specific deficiencies*; which were further divided into several sub-categories. The companies used these category lists on a day-to-day basis, for example to help focus groups identify common mistakes. They are also helpful for users as an ergonomic diary to continuously report problems.

„Adjusting Screws"

Many of the problems identified in the companies can be solved relatively easily by software-ergonomic "adjusting screws". These include, for instance, optional settings within the SAP-system that are suitable for both users and system experts to adapt the system to the needs and preferences of the users. A number of these "adjusting screws" were identified during the course of the project.

Amongst the adjusting screws that can be used by *users*, are, for example,

- Allocating default values to various data fields,
- Creating individual lists of possible values for entry fields;
- Changing the layout of tables in masks, etc.

Amongst the adjustment screws that can be used by *system experts* are, for example,

- Adding and hiding data fields in screen templates;

[9] For more information on the Ergusto and ErgoCust projects, see the homepage www.ergusto.de.

- Changing of masks with the help of the tool GuiXT;
- Introducing company-specific definitions for compulsory- and optional data fields, etc.

Knowledge of Adjusting Screws

The analyses highlighted the fact that many of these adjusting screws were unknown to both system experts and users. About half of the people working with the system did not know, for example, how to create individual value lists, or how to allocate default values to entry fields. This lack of knowledge is in the first place due to insufficient training for both groups. Trainings for administrators e.g. enable them mostly to keep the system going from a functional view, but they hardly get to know the various options to design a user-friendly system. Special "Tips and tricks" training courses for both system experts and users could significantly increase the use and awareness of these adjusting screws.

Utility of Ergonomic Customizing

When testing the effect of our measures in the companies using SAP, it became clear that ergonomic customizing could improve the ergonomic quality of SAP software. Statistically significant results for improvements ($p<.05$) were obtained for the four principles "suitability for the task", "conformity with users' expectations", "error tolerance" and "suitability for individualization" across the companies, as well as an increase in the ISONORM-9241/10-total score (see figure 2).

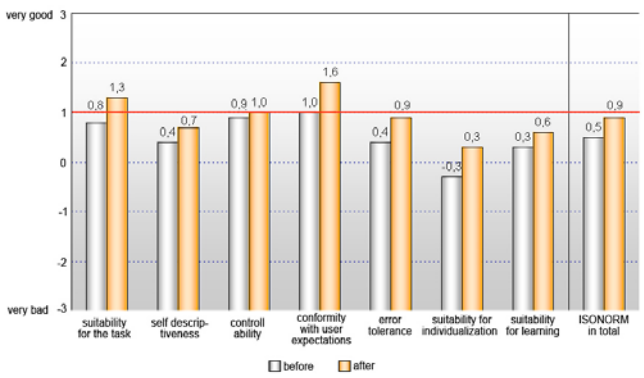

Figure 2: Changes in the ergonomic quality of SAP R/3 HR software through ergonomic customizing

Analysing the results achieved by correcting deficiencies in the system shows that ergonomic customizing has an effect in the following domains:

- Increased effectiveness: tasks that the system was previously unable to perform satisfactorily can now be fulfilled more completely and with greater accuracy.
- Increased efficiency: obstacles and complications are circumvented.
- Reduced effect of mistakes: costly errors are avoided.
- Reduced strain: stress and mental strain from working with SAP are reduced.
- Increased productivity: Ergonomic customizing is a return on investment.

Preliminary conclusion

The results of the Ergusto project show that post-implementation improvements in the usability of SAP-systems that are deficient in ergonomic quality are possible, sensible and advisable. However, from the point of view of costs, efficiency and user satisfaction, it seems much more prudent to guarantee usability from the outset. It is therefore necessary to familiarize the consultants, experts and companies who use SAP with the concept of ergonomic customizing, and to provide them with the necessary tools to apply them to SAP. The ErgoCust project suggests one way of achieving this aim.

ERGOCUST

The ErgoCust project aims to integrate ergonomic needs and targets into the introductory process: in short, Integrated Ergonomic Customizing (IEC).

The *conditions* for successfully integrating ergonomic customizing into the introduction of an SAP-system, the *IEC model*, the implications for *training* SAP-consultants and customizers, and the importance of a parallel *campaign* of software ergonomics are detailed below.

Conditions

Certain conditions are necessary for the successful integration of ergonomic customizing into the introduction of SAP-systems:

- Practical procedures must be in place and adequate data collection, prototyping, and evaluation tools available (procedural model);
- IT-consultants specializing in SAP-products (SAP-consultants) must be familiarized with software-ergonomics and trained in Integrated Ergonomic Customizing;
- An increased awareness of the importance of software-ergonomics must be created in the companies at management, project coordinator, and IT-staff levels.

The integration of ergonomic customizing into the SAP-introduction process on a broad basis can only be achieved in companies that satisfy these criteria.

Procedural model for IEC

In the first phase of the project, in cooperation with experts from SAP Inc., a procedural model for Integrated Ergonomic Customizing was developed and harmonized with SAP Inc.'s methods for introductory processes (ASAP Implementation Roadmap). This model considers the process of integration from an ergonomic standpoint, from initial planning using demand analysis, aim development and prototyping up to continuous improvements after the

SAP-system has been implemented. Ergonomic foci and suitable tools and methodologies for ergonomic optimisation were developed for each phase of the integration project. During this process some additional project tasks were identified using the SAP-method, but more frequently steps that had been foreseen were extended to include ergonomic aspects; for example, the integration of usability aims and indicators into anticipated project aims. However, contrary to SAP-methods, the procedure of the IEC-model is more user- than process oriented. As a result, one focus of the procedural model of IEC is early and extensive qualifying user participation, going beyond the Key-User-Concept as foreseen by SAP. Many options are opened by this approach, such as users being able to participate in the evaluation of aims and in prototyping. Since the procedural model for Integrated Ergonomic Customizing is conceived as a modular optimal model, it incorporates comprehensive options for ergonomic optimisation. To successfully integrate IEC into an implementation project, the elements of the procedural model that will be included must be specifically agreed at the outset.

Training

After testing our procedural model in SAP-introductory courses for companies working with SAP, a qualification unit in software ergonomics and in using IEC was developed for SAP-consultants, customizers, and project coordinators in companies using SAP. In addition to the classical use of seminars for instruction, the qualification includes software-ergonomic coaching in a real introductory project. Depending on the project-specific demands, the training can range from supporting the development of measures for a company, expert consultation regarding concrete problems that came up during the introductory process, to supervising the use of software-ergonomic instruments.

Campaign for Software-Ergonomics

In parallel to these activities, companies should develop a higher awareness of the importance of software-ergonomics in companies on a wide-ranging basis. Company decision makers should see it not only as an important factor in productivity and health-promotion, but according to legal regulations as an indispensable measure for those who work with computer screens. Only then can a long-term preventive strategy for the creation of company-specific user-friendly adaptive standard software, as propagated in the IEC-model, be developed in the work environment. For this reason, public relations work to raise awareness of the topic of software ergonomics forms a continuous, integral part of the ErgoCust project. In addition to company representatives, members of clubs, societies, associations, unions, professional trade associations, and so on are informed and motivated to spread the word.

CONCLUSION

There is room for improvement in the integration of ergonomics into current practices of SAP installation. The potential offered for the ergonomic customizing of SAP software is barely being tapped. The ones who suffer in this situation are end-users and companies who encounter restrictions in effectiveness, efficiency and satisfaction. The solution for these problems can be found in the concept of Integrated Ergonomic Customizing (IEC), since it explicitly considers software-ergonomic user demands from an early stage in the process and factors them into the system.

REFERENCES

1. Prümper, J. Test IT: ISONORM 9241/10. In H.J. Bullinger and J. Ziegler (Eds.), *Human-Computer Interaction – Communication, Cooperation, and Application Design* (1999), 1028-1032. Mahwah, New Jersey: Lawrence Erlbaum Associates.
2. Martin, A. and Eastman, D. *The user interface design book for the applications programmer*. New York: John Wiley & Sons, 1996.
3. Cornnally, C.E. and Tullis, T.S. Evaluating the user interface: Videotaping without a camera. In *Proceedings of The Human Factors Society*, Santa Monica, CA: Human Factors Society (1986), 1029-1033.
4. Prümper, J., Zapf, D., Brodbeck, F.C. and Frese, M. Some surprising differences between novice and expert errors in computerized office work. *Behaviour & Information Technology*, 11, 6 (1992), 319-328.
5. Zapf, D., Brodbeck, F.C., Frese, M., Peters, H. and Prümper, J. Errors in working with office computers. A first validation of a taxonomy for observed errors in a field setting. *International Journal of Human-Computer Interaction*, 4 (1992), 311-339.
6. Vaughn, S., Schumm, J.S. and Sinagub, J. *Focus Group Interviews in Education and Psychology*. Thousand Oaks, CA: SAGE Publications, 1996.
7. Hurtienne, J. and Prümper, J. Stress in the Office: the Influence of Software-Ergonomic Quality. In D. Harris, V. Duffy, M. Smith and C. Stephanidis (Eds.), *Human-Centred Computing: Cognitive, Social, and Ergonomic Aspects* (2003), 63-67. Mahwah, N.J., London: Lawrence Erlbaum Associates.
8. Hurtienne, J., Abele, P., Floegel, S., Prümper, J. and Stein, B. Usability von SAP-Systemen: Interventionen und Ergebnisse des Ergusto-Projektes. In M. Hassenzahl and M. Peissner (Eds), *Usability Professionals 2004* (2004), 34-37. Paderborn: German Chapter der Usability Professionals' Association e.V

Users? What users?

- shaping global corporations and generic users with ERP

Christian Koch
Dep. for Civil Engineering, Building 115
Technical University of Denmark, 2800 Lyngby
e-mail: ck@byg.dtu.dk

ABSTRACT
Most ERP-research, which worries about end users, focus on the implementation of the system. Since ERP-systems are mass produced, but also configurable systems, this focus continue to be important. It is however increasingly necessary to understand the role of design in this context of mass customisation. When creating a commodity under such circumstances, design should be kept at a distance and be able to span differences between a large number of customer organisations. It is attractive to preconfigures users in best practice workflows and standard user profiles.

The paper addresses som contemporary challenges in information systems research, such as the issues of multispatiality, proximity, spanning and the construction of the generic. The paper present two cases that are used to describe how design and interaction with users occur under such circumstances. The first case is a design network, denoted Octopus, which look at a ERP system and its community, and how it establish distant and/ or mediated users, which are a necessity in the long term development. The second case, Dolphin, describes how ERP is implemented and operated in a professional service enterprise. The cases illustrate the important arenas for development has moved away from the direct interaction between the designer and the user.

INTRODUCTION: OCTOPUSES, DOLPHINS AND SHRIMPS
The changing conditions of the global economy continue to place information systems and communication technologies into the core of desired enterprise transformation. Enterprise Resource Planning (ERP) is one core system in the shaping of global corporations with an estimated global volume of 8 billion USD in 2004 (Mansini 2004). ERP has over the last ten years been gradually developing away from first time implementation in manufacturing enterprises, into a much broader set of challenges, thereby also targeting management approaches and attempts to study and understand the interaction of ERP with organisations and people. Some of the main tendencies are

- From single site to global organisational change
- From single source to best of breed configurations in ICT-architecture(s)
- From manufacturing to a number of other sectors, i.e. public, service etc
- From core enterprise functions like accounting and production to external interfaces (SCM and CRM)
- From implementation to life cycle adjustments to emergent business strategies

These developments challenge several strands of research. This contribution addresses some of the theoretical imperfections that characterise several perspectives in ERP-research including information systems research and sociology of technology approaches. Trends in studies of ICT and ICT driven change do to some extend take into account the increased complexity of contemporary ICT and techno organisational change. However complex and multilocated "systems" such as ERP reveals a legacy, which rests on relatively simple metaphors or core concepts such as actants, boundary object and "embodied knowledge", accompanied by a restricted set of actors such as the designer, the manager and the user (Avgerou 2004) This contribution suggest that ERP-"systems" needs to be understood as heterogeneous assemblages of human and material elements. These assemblages can be understood as ERP-communities of software companies, customers,

professional associations, different kinds of hardware and software, implementation procedures, practices and rhetoric spanning time and space. The systems are not solely shapeable clay, rather they are heterogenous materiality composed with abstract discourse-elements with a certain hardness. Moreover ERP is (big) business and design of these systems occur under strategies of mass customization, where the encoding of the generic user is a necessary tool to reduce development costs and time to market. ERP is a also a business with hypercompetition and with constant restructuring of players.

I have elsewhere argued that two main arenas are important in designing and influencing ERP: the networked design arena and the context of the user organisation (Clausen & Koch 2002). The present paper present two cases that are used to describe how design and interaction with users occur under such circumstances. The first, where the network is denoted Octopus, mobilize a biography of an ERP systems and its community to establish that distant and/ or mediated users are a necessity in the long term development. The second case describe how ERP an information systems modeled over the template of an manufacturing company is implemented and operated in a professional service enterprise. The enterprise is called Dolphin, to describe its agility in developing new knowledge based services.

So what about the users? –well the users are the shrimps….

METHOD

The theoretical approach has its centre in interpretative sociology, but is multidisciplinary. It builds on technology studies, sociology of organisations and management studies (such as McLoughlin, 1999). The multidisciplinary approach enables an opening of the black box of innovation. It understands innovation as a social process of building coalitions and networks.

The case study of Octopus, a multinational software house with offices in Denmark, was carried out within the frame of BiCON, but supplemented with other material (BiCON, 2000). The case study builds on four semi-structured interviews with internal players of Octopus, including two interviews with a high level manager, and an interview with a software developer and a tester, all with close relations to the development of a particular module of Octopus's ERP-system. The network of socalled value adding resellers (VARs) around Octopus was covered by eight semi-structured interviews selecting the VARs with relation to the development of the particular software module studied. Finally, 10 customer enterprises using Octopus's ERP-system were visited carrying out ex-post evaluation of the implementation. In these case studies, several interviews were carried out at each enterprise. (Another version is published in Koch, 2004

The case study of Dolphin, the Denmark based consulting engineering company is developed over four years of interaction. Managers from the IT-department of DOLPHIN contributed to seminars held at DTU and DOLPHIN. Dialogue and one in-depth interview were carried out between 2000 and 2005. This interaction was supplemented with written material.

Finally, the discussion on ERP in general draws on projects carried out during the period of 1994-2000 (Koch 2001 and unpublished research on the financial sector). Koch (2001) is a major evaluation and report on ERP implementation in Danish manufacturing. This covers technical and organisational variants of enterprises, including enterprise choices in configuring ERP. The samples consist of 26 ex post cases and 4 longitudinal studies. This material is used to underpin the analysis of choices of configuration of an ERP-system in the use-setting. Koch (2001) also discuss the design communities and the financial sector research looked at a major multinational ERP- software house and its organisation of networked design of a industry package, which is a specialised version of the ERP software.

TAKING STOCK OF ERP- STUDIES

ERP surfaced roughly around 1993 and continue to develop. There is therefore a for researchers very useful paradox between on the one hand relatively well established lines of research and on the other hand research agendas that keep establishing themselves. The understanding of implementation processes, impact of technology at work and in organisations, including control and skill issues has been extensively elaborated, and is nevertheless still needed in ever-new areas. Off course the approaches need fine tuning, but the paradigm and a host of methods is established. When ERP change is introduced in new forms of work and organisation, sensitivity towards the new domain and its micro sociological life can be mobilized (Benders et al. fo, Elmes etal 2005, Grant et al 2003, Cornford & Pollock 2003 etc).

The relationship between ERP, design and use is complex and interactive. From one perspective ERP embodies work procedures and practices which unfolded in an organisation would promote certain ways of organising. ERP as a materiality here contributes to stabilizing certain organisings and inhibiting others. However local sociality interacts with ERP and gets co constructed in this process also meaning elements of ERP are changed. In this process immaterial elements of ERP are active. From a wider perspective ERP is spread as a template for organisations, which attracts organisational models in one context and transport it to another. In this way ERP plays a role as a type of material and immaterial institution.

Nevertheless there is major impairs and lacunae to be addressed. To the authors a move away from implementation studies, which sometime are too much snapshots (Williams, 1997, Pettigrew, 1985) is central in two ways. First because the separation between micro

	Short term	Long term
Micro	**Implementation**	**Life cycle "after going live"**
Meso	**Professional associations Network constellations**	**Institutions of technology (MRP II) Biography of system elements**
Macro	**Technology policy, promotion**	**Global technological and company change Communities**

processes in enterprises on the one hand and meso and macro processes holds to a lesser and lesser extent. Second lifecycle oriented studies on continous technological change is need. Figure 1 illustrates the point.

Multispatiality

Early studies of technology, such as Pettigrew (1973) established an understanding of the implementation process taking the information technology for given by external factors. At least from around 1990 some contributions expanded to a much broader understanding of technological development.

Fleck (1993), Salzman &Rosenthal (1994), Williams & Clausen (1997) and others pointed at the issue of multilocation of technological change as more or less a duality between two spaces: the developing (software) company and the consuming/implementing company. Mcloughlin et al (2001) provide an up to date example of the same kind of constellation, extending it into a hybrid organisation between the software developers and the consumer company. Clausen & Koch (2002) argue that small IT-vendors in the mid nineties operated in segments of a few customers around a software houses.

However the main point that neither the single organisation, nor the "two spaces in interaction approach" encompass the global features of the development and implementation of ERP. There is a need to go much further that the dual arena-concepts discussed above.

At a closer look ERP-systems are not developed in a single-placed software house. The systems/ and their vendors have developed into worldwide organisations, where further development of software is occurring literally hundreds of places in parallel. Companies like -SAP might have a majority of development resources located in one country (Germany), but encompass development in most of their offices worldwide (for example so-called country specific engineeering). The company engage in a multitude of development alliances with representatives of future customer groups, such as contractors in construction, universities and other public institutions (Pollock et al 2003).

In such a constellation of companies developing the same suite of software there are internal tension on how and when to develop what (Mcloughlin et al 2001). Some actors see it as their competitive advantage to engage in what starts as bespoke software with a small customer group. In doing so they dont wait for the "mother company" to develop a new facility as part of the next version. Rather they develop their own an thereby create pressure on the mother-company. In parallel to this usergroups, consultancies and others develop interpretations and political programs on what it now needed (Koch 2003).

For the time being there seems to be two complementary roads to go: First developing the new materialism approach that see technology as material, immaterial and social (Pels 2002, Law 2002, Hetherington 2002, Latour 2002, Woolgar 2002, Turnbull 2002). And to add a discussion of the inbuilt organisation in ERP packages, looking at spatiality producing elements and lookingat how ERP produce certain spaces in organisations and at the same time tend to shape these organisational element

Second to view the constellation as a community or even a community of practice (Koch 2003 a.o).

The community approach as proposed by Koch (2002) is an empirically "induced" term, which attempts to grasp a vast heterogenous grouping, which have at least a technology, the ERP system in common. The distance between the producers and consumers in the community is intentionally kept high by the producers since competitiveness is seen to develop from being able to standardise and mass produce software. A number of organisations act as mediators like consultants, VAR (value added resellers), professional associations, user groups, education and training units. A certain level of common discourse on the capabilities of the ERP-software and inbuilt organisational models is developed through the closer encounters like implementation processes, spread of information at seminars, magazines and other types of communication.

In understanding the community as described it becomes different from Wengers community of practice. Although Wenger allow artefacts and reification to play an (important) role, technologies are hardly seen as constituting the community of practice (COP). Wenger gives preference to smaller, less invasive artifacts (Wenger 1998: 83). At least in the 1998-version, there is considerable emphasis on co-presence as part of the way local constitutive practice enfolds.

Finally although Wenger assures us that COPs not necessarily are peaceful and harmonic (Wenger 1998: 77, 85), the dominant stance build on the notion of "joint enterprise", which is not just a stated goal but something which is continually redeveloped among participants.

In contrast to this, the constellation around an ERP-system is usually multilocated in a way that creates islands of arenas for co present practice, but also vast distances, with inbuilt tensions and conflicts. The central in an ERP-community is not the practice but the constitutive joint technology.

Proximity and Spanning

Since the phenomenon of multispatiality also represents a contemporary development of user- domains, it becomes crucial to understand proximity and the spanning between geographically dispersed locations. Proximity is important for early information systems development thinkers in their attempt to promote the collaboration of designer and users (Kyng & Mathiassen 1997). It is however today other dynamics that govern design, and the argument below is that strategies of mass customisation imply that social distance is aimed at in order to create the generic functionality.

It is here important to underline that proximity is more of a social construct than it is a natural given feature. Studies of the user organisation shows (Hinds & Kiesler 2002 a.o.) that new proximities are created when ICT mediate and help spanning the geographical distances. The social shaping of work in new proximities and the shaping of new relations between design and use enabled by ICT is important issues to address. Or the put it with Mackenzie (2003) who studies IT systems as infrastructural system; "Intensely invested issues such as configurability, scalability flexibility and distributing process, imply that infrastructural design and implementation have a complicated relation to place".

A number of authors have proposed solutions to these issues. Castells proposes the term Space of Flows, to understand that place and locality is somehow dissolved and substituted with an importance of non spatial communication over the web (Castells 1999). Harvey speaks of

Time Space Compression (Harvey 1996), Giddens of Time Space Distanciation (Giddens), Wenger and Star of communities of practice , brokers and Boundary objects (Star and Wenger). Many have proposed virtual Organisation (Koch 2001 a.o.). It is however still relatively few which actually changes their research strategy in order to be able to conceptualuse and better understand spanning and proximity. Those who did propose imaging locality (Mackenzie 2003), travelling risk (Rolland 2004) and global ethnography (Burawoy et al.2000) as terms. On a much more practical level Holmstrøm proposes the use of the net to directly incalcutate distant users in design (Holmstrøm 2004). It seems to me however that there still is a long way to go to understand the reconceptualised spaces for design and use.

Mass customizations

The IS-research community has been slow in accepting and reorganizing the research according to the occurrence of big business enterprise in software development and use (Avgerou et al 2005 a.o.). Williams et al (2005) labes this perception the "design fallacy" describing the too optimistic view, that the problems of the users basicly is an issue for good craftsmanship by designers. This stand in quite a contrast with the development in the ERP markets where hypercompetition wiped out a series of locally operating software houses which emphasis bespoke solutions (Clausen & Koch 2002), only leaving a few large players to survive and pass 2000.

The counterstrategy to the competition developed during the nineties. Mass production and generic packages (here ERP) boomed. The configurability, choice levels (modules, submodules, preconfigured workflows,) and user profiles are the central element of the massproduced customization possibilities. As the case below discuss and Pollock describes in his research (Pollock et al 2004) the design of generic packages makes it necessary to create distance to users and to mediate between the few taken in as making the "span" of differentiated organisations.

ERP- companies like SAP have here been successful in creating a belief in that their product were representing "best practice", thus creating a situation where local used were driven into the defensive, since specificies of the setting was construed as unnecessary barriers for development.

It is important to point at the possibilities of local appropriations with the configurability. As the DOLPHIN case below shows, packages can be reconfigured quite profoundly. Seen from a globalised user perspective, the use of a common technology is moreover a condition of possibility for the creation of alternative experiences with more user oriented configurations. Koch (2001) did not find but a few examples of ERP-support for teamworking in manufacturing, revealing that at that time organised users (unions etc) did not exploit this potential.

CASE 1: BIOGRAPHY OF AN ERP COMMUNITY: OCTOPUS AND MASS CUSTOMISATION

The global company in this case has an extensive international network with a number of software development locations. The company is called "Octopus" here and its Danish division "Hansen". The software is a portfolio of several generic ERP packages, where the focus here is on one. The installed base of the system covered at the time of study more than 50,000 customers within Denmark and more than 15,000 abroad. The ERP system at the time was sold in more than 20 countries. Hansen was founded in the 1980s, independently of Octopus, and its growth was moderate in the first 6 years, bringing the turnover up to 100 million DKK (approximately 14 million US$). From 1994 to 1998, the turnover tripled and the number of employees went up from 150 to 450. The company was merged with another ERP-player in 1999.

The development, sales and implementation of this software involve a complex collaboration between Hansen itself and a network of VARs and a small number of major customers in the private and public sector. Many of the VARs are small whereas a significant group had a comparable turnover to the ones of Hansen itself. The VAR network continued to develop with new entrants, existing members leaving and other restructuring effects (mergers between VARs and so on) throughout the 1990s. The eight VARs studied represent both small and very locally operating companies with 10–30 employees and larger ones with around 5–700 employees. The VARs both co-operated and competed within this framework. Many had overlapping customer groups, while others focused on more restricted market niches. Within this framework, a range of additional services had been developed and 'bundled' with the main software product, such as consulting, training and additional software modules.

Within Denmark, the network of VARs consists of more than 100 companies. Internationally, there are approximately another 500 VARs linked to Hansen. These are legally independent companies with various types of formalised relationships with Hansen and 'end-user' customer enterprises. Therefore, it is important to note that, in contrast to classical software design and development of bespoke systems, Hansen does not have a direct relationship with most of its customers. The development of the collaborative networks with the VARs was a consequence of a deliberate strategy. This sought to use such inter-organisational collaborations as a means of 'outsourcing' sales and implementation, while maintaining product development activities in house. However, the larger and some of the more specialised VARs started developing additional software. The result was a distributed system of product development.

In this case, the focus is on the development of the third generation of the ERP-system and a specific module within this. This project involved the development of collaborative networks within Hansen itself, which then interacted with the broader network of VARs and selected customers described above.

The software development process was initiated in the mid-1990s. It is a clear example of a top–down 'classical product development' where innovations in the technical content of the product initiated by the core enterprise are preferred to building on experience of developing and using experience with the earlier generation of the product gained by the VARs and end-users. The overall business objective behind was to make the product more appropriate for use by medium sized (not just small) enterprises and to expand in the international market. The organisation of the product development process was based upon the microsoft solutions framework (MSF) (see Cusumano and Selby, 1995). This represented a shift from a traditional functional project organisation to a form of matrix organisation. This involved the decentralisation of decision making to product teams and the shortening of development cycles. The objectives behind were: first, a reduction in 'time-to-market'. Secondly, the perception sustaining growth of the company was dependent on finding new ways in which to 'leverage' the skills, expertise and knowledge of programmers and system developers during the product development process.

The formation of teams for the software development broadly followed the MSF rules and procedures. One of the teams was followed in their work to realise one module of the package (the project management module). The team was particularly successful in negotiating, with the overall project management, an appropriate fit of its task to available resources. The team was able to limit the scope of the tasks it was required to undertake and was able to persuade the project management to take a task away from the team. Similarly, in the planning phase, the team was able to take the initiative in prioritizing certain tasks and downplaying others. Subsequently, the team was able to win additional human resources.

Internal communications within the team appeared to work effectively; as specified by MSF, the team included a product manager, recruited externally, who had practical experience in the domain the software module was to address. In most of the MSF phases, the team was able to agree internally most of its priorities and design and to resist 'interference' from outside. At 'post-mortem' meetings held at the end of each cycle of the MSF, several activities were evaluated by the team. These included the internal collaboration within the team itself and how their respective roles were functioning. The team established external communication about the customer requirement with the external intermediary network of VARs and significant major customers. In the first phase, there were informal interactions between the team and the external VAR network. Here three VARs and one significant end-user/customer were consulted. These largely informal linkages served to open up information and communication channels between the VARs (who had a more direct

experience of customer requirements) and the team (who was also able to manage the VARs expectations as to what the new module would actually deliver). In a parallel process, the VARs were more 'formally' consulted. A committee of VARs held three meetings before project management decided to halt the activity. This reflected a continuing debate within Hansen on the role of the VARs. Several different departments of Hansen articulated different views on this issue. Within the team studied, some members proffered an interpretation that 'listening to the customers is in contradiction with being ahead of the competitors'. The beta version of the module from this first cycle was released against the wishes of the team. This resulted in a heavy bombardment of telephone calls to the team from VAR representatives and others, who wanted specific details incorporated in the next cycle.

Two further forums served to facilitate the flow of information between Hansen and the VARs and between the VARs and end-user/customers. These were monthly strategic meetings with both the Hansen distribution function and project management and project development workshops organised by the VARs for their customers, which, in some instances, have resulted in joint specification of requirements. However, from the point of view of the VARs network the overall development process posed a number of problems. While all VARs were keen to inform and support the development of the new ERP package, not all were convinced that the end product was superior to competitor offerings. In some cases, VARs chose to develop their own additional modules in order to make their total offer more competitive from their viewpoint. Some VARs indicated that early product releases lacked the necessary quality and created problems with customers. At the end of the research period there were still some VARs who would not implement the main releases of the ERP package because of perceived quality problems. Several VARs expressed consternation regarding infrequent releases of service packages for servicing the existing base, and some mentioned the lack of help from Hansen in creating sales arguments in relation to competing systems. To this end, VARs used informal networks and contacts with software development project teams to gain product information of this type. In some cases, these flows of information contradicted internal structures and procedures within Hansen.

Such tensions also highlight a differentiated landscape of the VARs. Many are 'total systems solutions' providers where additional tailor made programming is a central offer. Some have a role as developers whereas others are mere implementers of a standardised system. If developers and total systems solutions providers flourish, it is a problem for Octopus in the long term, so far as the company is primarily interested in branding its ERP products as very flexible standard solution with little need of subsequent customisation. The transformation from Hansen to Octopus occurred as a process in parallel to the described.

CASE 2: IMPLEMENTATION USE AND FURTHER DEVELOPMENT OF ERP IN A PROFESSIONAL SERVICE

The case study covers a Danish consulting engineering enterprise, DOLPHIN being an international firm with main office in Denmark. The organisation is matrix- like. The focus horizontally is on customer groups and/or products, whereas the vertical focus is major areas of competence. The major competencies are:

- Energy
- Environment
- Development Planning
- Major infrastructure
- Building and Operation
- Construction Management
- Economics and Management
- Information Technology

These are organised in 9 divisions supported by a general services department encompassing the IT-organisation and staff. The company employs some 1500 in the parent company in Denmark, supplemented with an international organisation, which however, for the time being is not supported by the SAP-solution.

The process of DOLPHIN with SAP R/3 falls, basically, in two phases: firstly, a SAP R/3 configuration and implementation, and secondly, the further development.

The Initial Vision: Substitution of legacy systems, R/3 as (almost) single source

The central elements of the initial vision of information systems management were developed in 1995-96. The old fragmented legacy systems architecture should be replaced by standardised IT solutions, using SAP R/3, Microsoft Exchange and Documentum as the three main elements. At the same time, DOLPHIN did not have ambitions on reengineering the organisation and the company's CAD-solution was also kept outside the area of renewal. DOLPHIN established a project organisation and made a market survey finally short-listing SAP R/3 and Oracle. A contract with an implementation consultant was signed in 1996.

The central business processes in engineering are the projects. From initial quoting to final delivery DOLPHIN have to manage more than 5000 projects per year. Moreover, the main resource of the enterprise is the human resource. Finally, the main content of the projects is handling and developing of information of various kinds. These basic preconditions tell a lot about the differences

between business processes and resources at DOLPHIN compared to the ERP-template of a manufacturing company.

DOLPHIN considered acquiring the business solution for construction and engineering, but did not feel that it sufficiently fitted with the central demands. Especially, registration of hours spent on projects by employees was seen as central. Since SAP found this demand with other companies in related sectors, such as management consultants, this became a formation of the grounding for a new business solution for service providers (Lykke, 2000). DOLPHIN thus became part of one of SAP 's sector alliances, which enforced both parties' strategy. SAP 's diversification strategy and DOLPHIN 's strive for strong solutions in its business processes.

Through the initial phase of the project, it was decided to try to stick to the features of standard R/3. Initially in the configuring phase, some main modules ware taken on board whereas others were left behind. Those included are financial modules, human resource, sales and project management. This choice implies that DOLPHIN 's configuration are markedly different from the manufacturing template, which would usually encompass material management and production planning as central for controlling of manufacturing resources, mainly machines and materials. Also many manufacturing companies would choose quality and maintenance management.

On the sub-module level, certain choices are characteristic. Within Human Resources, the cross application time sheet module (CATS) is thus crucial in this configuration. All 1500 employees on a daily basis use this module. Also travel is important in human resources. To support financial reporting on projects, the special ledger sub-module (FI-SL) was configured. The project group ambitions of adjustments of parameters overthrew the schedule for the implementation. Moreover, co-operation with the external implementation consultants had to be cancelled. Configuration of user profiles and training was carried out in a relatively traditional way. The project group considered the appropriateness of narrow or broader profiles, and developed an ordering of access elements in 200 activity groups. These are mixed for the individual user according to user groups. For example, employees have two central profiles for CATS and travel respectively and each department needs to have a specific activity group. Other important meta-groups of user profiles include those for line managers and financial management.

The implementation process of the system revealed a number of small problems related to users' use of the system. Training was developed in two waves, the first at implementation in 1999 and a second wave as a campaign to improve user support, user interface and other elements improving end-user situations. The present status is that the 1500 employees now use the system for hours spent registration and 7-800 project managers for project management accounting. Around 60-70 line managers use the system for financial management. Finally, central functions like finance and human resource are using the system. However, project management in general is still carried out using other products like Microsoft project, Artemis, Primavera and others (see below).

Further development: best of breed architecture, knowledge management and acquisitions

DOLPHIN has continued to develop its installation since 1999. First and foremost, an upgrade from 3.1 to 4.6 was realised in the autumn of 2000. A number of smaller upgrades and developments have been carried out using ABAP/4 programming, resetting of parameters, changes of screens and design of new reports. In 2001, a larger configuration and programming of automatic data generation for intellectual capital accounting was implemented. The enterprise strategy for knowledge management is in this way supported by R/3, along with the other systems in DOLPHIN' s present IT-architecture. Although the original belief in 1995 was that DOLPHIN could realise an integrated architecture with a few systems, the IT-architecture in 2001 is still combined by a number of systems. External co-operation in the construction area, for example, implies that DOLPHIN needs to be able to tackle different system interfaces almost in every new project. Moreover, intranet, projectweb, project management and computer aided design continue to be areas where others can deliver better systems than SAP. The present best-of-breed architecture consists of nine integrated applications/ systems:

- Intranet
- Project Extranet
- Internet Interface
- PRIS (project information datamining)
- ERP
- CAD
- MS Office
- MS Project

PRIS is an in-house developed system handling information on previous projects. The integration between software packages using object-oriented techniques is therefore central and will even gain importance in the future, which parallels the general increased focus on integration of information systems (Hasselbring (2000), Ring and Ward-Dutton (1999))

After roughly a year of operation it was decided to configure the system to support and "automise" the intellectual capital accounting of the company, which was only possible after considerable effort by the internal IT-

department. Lately business intelligence functions have been configured and offered to department managers.

In parallel to this the company decided to globalise further with a special emphasis on a single service product, but also with a broader set of service products to the European market. The corporate management saw mergers and acquisitions as central for this strategy. The acquisition of two major companies meant developing two different IT-integration approaches. In the first merger the user profiles were enlarged, accounts etc. in a full embarkment of SAP-functionality, which collided with organisational practices in the acquired company. The other merger was much more cautious in recognizing the existing ERP-system in use in the acquired company.

DISCUSSION AND CONCLUSION: USERS AND SHRIMPS

In the Octopus case and ERP product based on mass-customization was reached early. The commodity feature became central and the hub-software house successfully keeps the users and the customers in general at a distance. The VAR network is instrumental in doing this, but also represents a potential competions , which means that a continual alignment is carried out in the design network.

The biography approach reveals that the long term development does not represent accumulative functionality, but rather waves of more or less profoundly new functionality. Also in this respect one can see the commodity aspect- to put it bluntly: the software gladly sold the same functionality several times. The case shows room for classical design activities. The SW-designers are thus active in shaping of the generic users. However the customers as such are not let in. It is one customer, which is active at the “birth” of the packages. This result resonates with Pollock study, where a few customers are invited onboard in the initial design phase.

The DOLPHIN case shows that, when R/3 is configured for consulting engineering the solution becomes markedly different from the manufacturing template. The R/3 implementation and use is after monthe of initial struggle evaluated as largely a success for DOLPHIN. Although DOLPHIN often have problems meeting detailed demands of existing processes when configuring SAP to support them. In other words R/3 is not endless malleable, but DOLPHIN 's considerable investment in internal R/3 competency pays off in the tailored solutions that the IT-department is able to develop. Through these internal resources, DOLPHIN is also able to compensate for dependency of the development of SAP 's service provider sector solution. DOLPHIN does have a number of wishes for further development of the sector solution.

Another central result is the way DOLPHIN combines different systems. Apart from the “best-of- breed” argument, it can be seen as a struggle of several paths and socio-technical constituencies, which is developing in a competitive way. SAP and its proponents would offer project management, document handling and intranet/ enterprise portal solutions. But other products are available and social actors within DOLPHIN would “stick” to MS-project, Microstation, Documentum or other tools.

In this development, users and management faces manoeuvring challenges where pitfalls are situated at both sides of the road. At least in Denmark, a number of mid size enterprises got used to close "partner like" collaboration with their IT-supplier in the eighties and early nineties. Such co-operation meant trust to, and dependency on an external supplier. They now face mass-produced packaged software, where partnership is, roughly speaking, out of the question. And the constituencies around the technologies are unstable; new versions and new companies keep emerging. This new situation challenges the skills of all types user representatives maybe especially information system managers, it seems more important than ever to create temporary and flexible alliances externally and internally. Internally, it will be important to create policy processes with broad participation, enabling the enterprise actors to formulate basic policies for the needed software. These policies can be used to decide where the resources for reshaping the systems should be set in. Priorities have to be made, and the interpretation of the IT-systems will continue to be equivocal (Weick, 1990, Knights and Murray, 1994). Different actors using different part of the system have developed different experiences and get different development resources. This is also echoed in the case presented above, where some employees possibly have developed an understanding of the system as inflexible and unduly complicated whereas other groups, like the supporter-users, interpret it quite differently. At the same time, external developments in the IT-sector and among collaborators will continue to create tensions for the IT-solutions adopted in the enterprises.

It hopefully clearly has derived from the cases and the analyses that creating a collective user body has some potential in supporting the development of alternative configurations of ERP-systems. The user power is limited on an individual level, they are but shrimps for the octopus. However as a collective movement, the apparent weakness turn out to hold more potential.

REFERENCES

Appadurai, A. (ed.). (1992). The Social Life of Things: Commodities in Cultural Perspective. Cambridge Cambridge University Press.

Avgerou C., Ciborra C. & Land F. (2004) The Social Study of Information and Communication Technology: Innovation, Actors, and Contexts. Oxford university Press. Oxford.

Barry A. (2001) Political Machines: Governing a technological society. Athlone Press. London.

Badham R., Couchman P. and Mcloughlin I.(1997). Implementing Vulnerable socio-technical change projects. In Mcloughlin I. and Harris, M. (eds). Innovation Organisational Change and Technology. London. ITB press.

Benders, J., R. Batenburg & H. van der Blonk, (fo) Sticking to Standards; Technical and other Isomorphic Pressures in Deploying ERP-systems, Information & Management.

Burawoy M., Blum J.A., George S., Gille Z, Gowan T., Haney L., Klawiter M., Lopez S.H., Riain S., Thayer M. (2000): Global Ethnography: Forces, Connections, and Imaginations in a Postmodern World. Berkeley. University of California Press.

Ciborra, C.U., Braa, K., Cordella, A., Dahlbom, B., Failla, A., Hanseth, O., Hepsø, V.,

Ljungberg, J., Monteiro, E. and Simon, K.A. (2000) From Control to Drift. Oxford:

Oxford University Press

Clark,P & Staunton,N (1989). Innovation in Technology and Organi-zation. Routledge London.

Clausen C. et al (1984). Arbejdsindhold i automatiseret produktion. Teknologisk Institut. Taastrup.

Clausen C and Koch C (2002). Spaces and Occasions in the social shaping of Information Technologies. The transformation of IT-systems for manufacturing in a Danish Context. Chapter Seven in Sørensen K. & Williams R.(eds) (2002). Shaping Technology Guiding Policy: Concepts, spaces and tools. Edward Elgar. Cheltenham. Pp 215-240.

Clausen C, Dawson P and Nielsen K.T. (eds)(2000). Special issue on political processes, Technology Analysis and Strategic Management. Vol. 12 no 1.

Clegg S & Tantoush T (eds)(2001). Special issue on political technologies and the technological politics of organisational change. Journal of Organisational Change Management vol 14 no 1.

Cornford J.& Pollock N. (2003) Putting the University Online: Information, Technology and Organisational Change. Milton Keynes, Open University Press.

Dawson P (2002). Behind Closed Doors: Multiple Histories and Competing Narratives in the Political Process of Technological Change. EGOS, Barcelona.

Dechow N.& Mouritsen J.(2004): ERP Manuscripts of Accounting and Informatics systems. In

Andersen K.V. & Vendelø M.T. (ed): The Past and Future of Information Systems. Elsevier. Amsterdam. Pp 95-110.

Elmes, M.B., Strong, D.M., & Volkoff, O. (2005) Panoptic empowerment and reflective conformity in enterprise systems-enabled organizations. Information and Organization 15(1): 1-37.

Fleck, J. (1993), Configurations: Crystallizing Contingency. The Inter-national Journal of Human Factors in Manufacturing. Vol 3. Wiley & Sons.

Gergen K.(1992). The Saturated Self. Penguin Books. New York.

Granovetter M.(1973). The strength of weak ties. American Journal of Sociology 78 pp 1360-1380.

Grant D. et al (2003): The false promise of Technological Determinism: The case of Enterprise Resource Planning Systems. Paper Third Critical Management Studies Conference. Lancaster.

Grint, K. & Woolgar, S. (1997). The machine at work. Technology Work and Organization, Polity Press.London.

Hanseth, O., Ciborra, C.U, and Braa, K. (2002) The Control Devolution: ERP and the Side

Effects of Globalization. The DATA BASE for Advances in Information Systems – Fall 2001,

32:4, pp. 34-46

Hall R. (2002) Enterprise Resource planning systems and organizational change: transforming work organization? Strategic Change vol 11. pp 263-270.

Hirsch Kreinsen H. (1993). NC- entwicklung als gesellschaftlicher process. ISF München Campus

Hvid H. & Møller N (1992). Det udviklende arbejde. København. Fremad

Hinds P. & S. Kiesler, Eds (2002)Distributed work: New ways of working across distance using technology, MIT Press,

Holmström H. (2004): Involving Distant Users in Packaged software Development: A user Community Approach. In Flensburg P. & Ihlström C (eds): IRIS 27 plenary papers. IRIS.pp159-179.

Ip W.H.& Chen B. (2002) Organisation structure and enterprise modelling in ERP realization. International Journal of Networking and Virtual Organisations 2002 - Vol. 1, No.2 pp. 170-183

Jones O. & Beckinsale M (2001). Micropolitics and Network Mapping: Innovation Mapping in a Mature Firm. In Jones O., F. Steward and S.Conway (2001) Social Interaction in Innovation Networks.

Jones O., Steward F.and Conway S.(1998). Introduction: Social Interaction and Innovation Networks. International Journal of Innovation Management 2 (2).pp123-136.

Kallinikos J. (2004) Deconstructing information packages, Information Technology and People, 17(1): 8-30, 2004

Koch C.(1997). Social and Technological Development in Context. In Nielsen K. & Clausen C. (eds.). Working Environment and Technological Development -positions and perspectives. -Working paper no.3 AMOTEK-centeret. Lyngby.

Koch C (2000). The Ventriloquist's Dummy? - The Role of Technology in Political Processes. Technology Analysis and Strategic Management. Vol. 12 no 1. 2000. Pp 119-138.

Koch (2001) Enterprise Resource Planning, erfaringer, ressourcer, forandringer. Ingeniøren Bøger. København.

Koch C. (2002). Enterprise Resource Planning-How to handle and get handled by a Heterogeneous Multilocated Abstract/Material Innovation. Paper prepared for subtheme 19; Developing Political Process Perspectives on Technological Change. EGOS 2002. Barcelona

Koch C. (2003). ERP-software packages – between mass production communities and intraorganisational political processes . Preece D.& Laurila J (eds.). Technological Change and Organizational Action. Routledge. London.

Koch C. (2004) The Octopus effect of Generic Software – The Case of ERP. Proceedings IRIS 27. Falkenberg. (IRIS is the association of Information Systems Research in Scandinavia).

Mackenzie A. (2003): These Things Called Systems: Collective Imaginings and Infrastructural Software. Social Studies of Science 33/3(June) pp365–387

Kopytoff, I. (1992). The Cultural Biography of Things: Commoditization as Process. In Appadurai, A. (ed.). (1992). The Social Life of Things: Commodities in Cultural Perspective. Cambridge Cambridge University Press.

Kræmmergaard P. and Koch C (2002). Managing ERP-systems after going live. In Christiansen & Boer (eds)(2002). Operations Management and the New Economy. Proceedings the 9th international Conference European Operations Management Association. Copenhagen Business School & Center for Industrial Production, University of Aalborg. Copenhagen & Ålborg.. Pp 731-744.

Kyng, M., Mathiassen, L., (1997). Computers and Design in Context. MIT Press, Cambridge, Mass.

Latour B. (1996): Aramis or the love of technology. Cambridge, Mass. Harvard University Press.

Latour, B. (1997). On Actor-Network Theory - A few clarifications. Soziale Welt.

Law J(ed) (1991). A Sociology of Monsters, Essays on Power, Technology and Domination, Routledge. London.

Law J. (2000). Objects, Spaces, Others. . http://www/comp.lancs.ac.uk/sociology

Law, J. (2002). Objects and Spaces. Theory, Culture, & Society 19(5/6).91-105.

Law J.,Cooper R. and Hassard J (1995). Technology and still life. Paper presented at The workshop on New Perspectives on Organisation, Technology and Innovation. CRICT Brunel university. London.

Law J. & Hetherington K (1999). Materialities, Spatialities, Globalities. Centre for Science Studies. Lancaster University. http://www/comp.lancs.ac.uk/sociology.

Mackenzie A. (2003): These Things Called Systems: Collective Imaginings and Infrastructural Software. Social Studies of Science 33/3(June) pp365–387

Mansini A. (2004) Presentation at EUROMA, Insead. Fontainebleu.

McLaughlin J., Rosen P., Skinner, D. And Webster A.(1999). Valuing Technology. Routledge. London.

McLoughlin, I. (1999). Creative Technological Change. Routledge. London.

McLoughlin I., Koch C. and Dickson K (2001). What's this "Tosh"? Innovation Networks and new product development as a political process. International Journal of Innovation Management. vol 5 no3. 2001. Pp 275-298.

Munt S.R.(ed)(2001). Technospaces- inside the new media. Continuum.London.

Nilsson A & Petterson J (2001). On Methods for Systems development in Professional Organisations. Studentlitteratur. Lund.

Orlikowski W. & Barley S.(2001): Technology and Institutions: What can Research on Information Technology and Research on organisations learn from each other. MIS quarterly. Vol 25 no 2 pp 145-165.

Pels, D., Hetherington, K., & Vandenberghe, F. (2002). The Status of the Object: Performances, Mediations, and Techniques Theory, Culture, & Society 19(5/6).1-21.

Pettigrew A. (1973). The Politics of Organizational Decision Making. Tavistock. London.

Pettigrew A. (1985). The Awakening Giant. Blackwell. London.

Pollock N, Williams R. & Proctor R.(2004). Fitting Standard Software Packages to Non-Standard Organisations: The 'Biography' of an Enterprise-Wide System. Technology Analysis & Strategic Management. Vol 15 no3.

Preece D. and Laurila D. (2003). Technological Change and Organizational Action. Routledge. London.

Rip A. and Schot J. (2002) Identifying Loci for Influencing the Dynamics of Technological Development. Chapter 5 in Sørensen K. & Williams R.(eds) (2002). Shaping Technology Guiding Policy: Concepts, spaces and tools. Edward Elgar. Cheltenham. Pp 155-173.

Rolland, KH (2004): 'Travelling risks': On the reflexive dynamics of global IT

Infrastructures. Proceedings of IRIS27. Falkenberg.

Salzman D. and Rosenthal (1994). Software by design. Oxford University Press. New York.

Sia, S.K., Tang, M., Soh, C., & Boh, W.F. 2002. Enterprise Resource Planning (ERP) Systems as a

Technology of Power: Empowerment or Panoptic Control? Data base for Advances in Information

Systems, 33(1): 23-37.

Walsham G (2001): Globalization and ICTs Working Across Cultures.Research papers in Management Studies. WP8. University oif Campbridge. Cambridge.

Weick K.1990: Technology as equivoque. in Goodman P.& Sproull L. (eds.).Technology and Organisations. San Francisco. Jossey Bass.

Wenger E. (1998) Communities of Practice. Cambridge University. Cambridge.

Williams R.(1997). Universal Solutions or Local Contingencies?. In Mcloughlin I & Harris M. (eds). Innovation, Organizational Change and Technology. London ITB Press.

Williams R.(2002). Introduction. in Sørensen K. & Williams R.(eds) (2002). Shaping Technology Guiding Policy: Concepts, spaces and tools. Edward Elgar. Cheltenham.

Williams, R. and Clausen C. (eds.) (1997) The Social Shaping of Computer-Aided Manu-facturing and Computer Integrated Manufacture, COST A4 Social Sciences. European Commission. Luxembourg.

Williams, R. Stewart, J. and Slack, R. (2005). Social Learning and Technological Innovation: Experimenting with ICTs, Edward Elgar. Aldershot.

Woolgar, S. (2002). After Word? – On some dynamics of duality interrogation, Or why bonfires are not enough. Theory, Culture, & Society 19(5/6).261-270.

Software is Orgware – A Semiotic Perspective on Computer Artifacts

Peter Brödner
Institute for Work and Technology
Munscheidstr. 14, D - 45886 Gelsenkirchen
Peter.Broedner@t-online.de

ABSTRACT
Contrary to common belief, IT systems often disappoint the expectations to increase productivity and flexibility of work and value creation processes. Moreover, most IT design and implementation projects still fail or burst time and cost budgets to a high extent. After presenting significant empirical evidence for these phenomena, the paper reflects on the reasons for their persistence by developing a semiotic perspective on the processes of dealing with computer artifacts in organisations. This semiotic view allows to understand these processes of designing, implementing and using IT systems as efforts of structuring social practices in organisations. Finally, a number of guidelines for an improved practice of designing and appropriating IT systems for effective use in organisations are derived from these theoretical reflections.

Author Keywords
Software crisis, IT productivity paradox, semiotic perspective on computer artifacts, computers as means of organising.

INTRODUCTION

Information Technology (IT) has often been characterised as "enabling technology" connected with far reaching promises. IT should allow for new forms of work organisation, open up new ways of organising value creation processes or even provide opportunities to create new businesses. Moreover, it should, according to common belief, lay the ground as a basic general-purpose technology for doing work more effectively and efficiently in a flexible environment.

Some of these promises have doubtlessly come true. However, most real IT implementations have turned out to be a barrier to rather than an enabler for organising flexible and more productive work and value creation processes. In essence, there are two strongly investigated empirical indicators for the unfulfilled promises and disappointed expectations: the so-called IT productivity paradox and the persistence of the software crisis.

Although there are growing bodies of empirical evidence for both phenomena, they are widely neglected in practice. In contrast, this paper wants to take the empirical evidence seriously and intends to reflect on the reasons for it. Why is it that only so few organisations succeed to substantially improve their economic performance by the use of IT systems? What are the reasons for the fact that, after forty years of strong software engineering efforts, still so many IT development and implementation projects fail again and again?

To this end, the paper starts with some significant empirical findings for both phenomena. It then develops a theoretical perspective on the nature of computer artifacts and their use for two reasons: First it can explain the empirical findings and second it serves as basis of cognition from which a number of guidelines can be derived for an improved practice of designing, implementing and using IT systems.

PERMANENT SOFTWARE CRISIS AND PRODUCTIVITY PARADOX: EMPIRICAL EVIDENCE

The Persistence of the Software Crisis

On a famous NATO Conference in 1968, the software crisis has been analysed and declared for the first time. Twenty-three years later, in 1991, Mitchell Kapor, the founder of Lotus Development Corp., stated in his Software Design Manifesto: "The lack of usability of software and poor design of programs is the secret shame of the industry" [11: 3]. And in 2004, another thirteen years later, a high level expert group in the UK put forward still again basically the same complaints about extraordinary high failure rates in the software industry culminating in the paradox: „We know why projects fail, we know how to prevent their failure – so why do they still fail?" [26: 10].

As a matter of fact, IT application projects do completely fail or at least burst their cost and time budgets to an extent and frequency which is markedly higher than in classical engineering disciplines. This is impressively confirmed by empirical findings from the Standish Group whose regular investigations collect data from large numbers of software application projects. In a recent survey from 2001 [30], based on data from over 30.000 projects, they found that only slightly more than one quarter of the projects succeeded, i.e. that they were completed on time and on budget, with all features and functions originally specified. All other projects either failed completely (cancelled before completion) or were challenged, i.e. completed and operational, but over-budget, over time estimate and with less functions than initially specified. And this did not substantially change over time (Fig. 1).

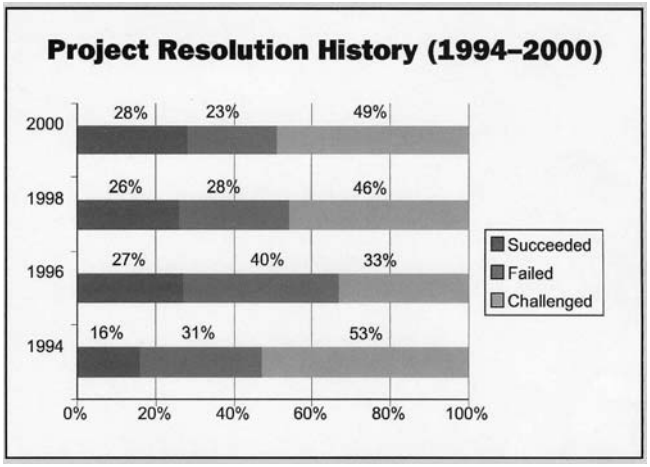

Figure 1: Success and failure of IT application projects (Standish Group)

From another empirical study on software failures we further know that the probability for failure highly depends on the size and complexity of the IT application projects. This probability grows exponentially with size up to a 50% cancellation probability for large projects with over 10.000 software functions [9].

Those complete software failures that have become known to the public, of course, form a peak of an iceberg only, since most failures remain hidden. However, the few that have been analysed all point, despite big differences between them, to the same reasons for failure again and again (see e.g. [23, 27]): insufficient project management and project controlling, underestimated complexity, lack of communication between designers and users, frequent changes of requirements during design and implementation, delayed decisions for progress, and incomplete documentation. Only recently, this has again been confirmed by the Royal Academy of Engineers: Alarming numbers of IT application projects „fail to deliver key benefits on time and to target cost and specification. ... This can be ascribed to general absence of collective professionalism in the IT industry, as well as inadequacies in the education and training of customer and supplier staff at all levels“ [26: 4].

The IT Productivity Paradox

In accordance with these observations, we find the widely investigated so-called IT productivity paradox according to which IT often fails to increase productivity (for an overview cf. [2, 12]). Despite huge and ever growing investments in IT over decades, no noticeable additional productivity effects have been observed on the macro level of the economy. In the USA e.g., real annual IT investments have increased by more than ten times from a level of 20 Billion USD in 1975 to a level of 220 billion USD in 1990. In the same period of time, productivity in manufacturing has increased by the same small average annual growth rates as before while productivity in the non-manufacturing sectors has even stagnated [3].

This has not changed so much since, although productivity in the USA – where investments in IT regularly surpass those in manufacturing technology since 1991 – has significantly increased in the second half of the 1990ies from an average annual growth rate of 1% in the years 1987-1994 up to an average annual growth rate of almost 2,5% in the period between 1995-2000. Many observers have ascribed this productivity growth to IT. However, as the most recent productivity study analyses, this extraordinary productivity leap was solely caused by specific and unique developments in just six sectors: wholesale and retail trades, security and commodity brokers, electronic and electric equipment, industrial machinery and equipment, and telecom services. Surprisingly, these unique developments mainly deal with organisational redesign of the value chains rather than higher efforts in IT system implementations [15].

Since productivity investigations on the macro level are admittedly problematic due to a number of measuring problems and to possible compensating effects of a multitude of simultaneous changes, the focus of interest in studying the paradox has switched to the micro level of firm performance. Firm level investigations have indeed produced a number of remarkable results. Besides a great number of case studies, econometric analysis of data from ca. 400 big US companies [4] points out that

- IT systems may improve the economic performance of companies, if and only if their implementation goes hand in hand with decentralisation, object-oriented reorganisation of work and investment in human capital,
- „intangible assets“, e.g. collective action competence, strongly influence the benefit of IT systems,
- companies decentralising their organisational structures achieve higher productivity in using IT systems than those who invest in IT only,
- the expenses for organisational renewal and training, e.g. in case of ERP systems implementation, are by a factor of four higher than the expenses for hard- and software.

Our own research on the implementation and use of ERP systems in German manufacturing enterprises produced comparable findings. Seven out of ten companies follow a purely technology-centered strategy and a top-down system implementation procedure with highly detrimental consequences for their economic performance. Thus, IT implementation projects regularly burst time and cost budgets to a considerable extent, while relevant performance indicators such as productivity, lead-time and in-process inventories are hardly improved, despite the extremely high expenses. The implementation process mainly concentrates on requirements engineering and design issues without end user participation, and efforts for appropriation and training are low. As a consequence, many functions of the system are not or poorly used, necessary knowledge about the integration in underlying business processes, their working principles and conditions is lacking, and large amounts of deficient or redundant data are being produced in use.

A small minority of firms only follows a more sophisticated and economically much more advantageous strategy starting with organisational redesign of their business processes and object oriented reorganisation of work with a clear customer focus. With these new organisational structures in mind, they simultaneously implement the functionally adapted IT system as a supporting tool and medium for cooperation. Accordingly, end users are strongly involved in these processes of organisational design and system implementation from the beginning and collective learning processes for appropriating and enacting the new ways of working are systematically organised [2, 13,14].

Similar findings have also been reported from case studies by other researchers [5, 6]. They obviously point to what is behind the paradox: How organisations understand and deal with computer artifacts either as means to automate existing work or as enabling and supportive media for creating and enacting an improved organisational practice decides about the economic benefits that can be gained. Making effective and beneficial use of computer artifacts is obviously more than implementing a functionally appropriate system.

THEORETICAL REFLECTIONS: COMPUTERS AS SEMIOTIC MACHINES

Semiotic Analysis of IT Systems: A Necessity

The misery indicated by these empirical data is, among other things, deeply rooted in conceptual deficiencies. So far mainstream computing science has – to some degree with the exception of the Scandinavian school – treated computer artifacts in much the same way as traditional engineering disciplines have treated their artifacts: By analysing relevant processes, functional specifications could be derived which the envisaged machine then had, as the result of a design process, to comply with. However, computers are symbolic machines manipulating data that represent information; their working principles obviously are fundamentally different from devices transforming energy or matter. Unfortunately, computing science has failed so far to develop an appropriate conceptual understanding of information or sign processes in which computers are embedded. Instead, the discipline has, besides its physical and mathematical foundations, strongly elaborated its requirements engineering and design methodology, but more of the same remedy only produces more of the same misery.

Sign processes, however, are a ubiquitous phenomenon: “Through almost all our life we are treating things as signs” [19]. The creation and use of signs as well as the treatment of information and meaning clearly are results of social interaction and, hence, their analysis falls into the domain of sociology. Unfortunately, the realm of things, how people conceive, sensibly act and interact with the objects they deal with in everyday life, reversely is being almost neglected in modern sociology. As a result, the comprehension of how people make sense of their artifacts in use, in particular computer artifacts, is poorly developed in sociology. As some kind of symmetric ignorance, both conceptual deficits, the lack of understanding sign processes and information in computing science as well as the missing comprehension of human interaction with technical artifacts in sociology, can at least partially be made responsible for the misery of inappropriate design and unproductive use of computer artifacts. Consequently, conceptual considerations must start to deal with these deficits. Those presented here are based on the pragmatic tradition of thinking, namely on the concept of sign by C.S. Peirce and the comprehension of things by G.H. Mead.

Signs are, according to Peirce, objects or processes that, in the view of an interpreter, stand for other objects or processes: A sign is “standing for something to someone in some respect.” Signs are our windows to reality, without them we could not even perceive it or sensibly act within it. In this perspective, a sign is a triadic relation ((R–>O)<–I) between three entities: (1) the representamen R as a material substrate of the sign (the object being interpreted as sign), (2) the designated object O and (3) the interpretant I as the meaning being assigned to the pair (R,O) through interpretation [24]. This sign concept is recursive: The interpretation is itself a sign that can be interpreted again.

In this perspective, computers can be identified as semiotic machines forming an own class of machines that can be well distinguished from the class of machines transforming energy or matter [1]. In the first instance, both types of machines have in common their close relationship with language, since they incorporate intentionally designed functions on the basis of concept formation and explicit knowledge. Humans have to interpret these functions within their action context in order to make sensible use of them (the functional “language” of the artifacts). The effects produced by these well-defined functions are then solely determined by the inputs. In order to make sensible inputs, intended use actions must be expressed in the functional

language of the artifacts. This holds for all technical artifacts, from the hand-axe to the computer.

The fundamental differences between both classes of machines, however, lie in their domains of operation, their working principles and their purposes. The *operational domain* of energy or matter transforming machines as well as of chemical or biological artificial processes lies in nature as they purposefully intervene in natural processes transforming energy or matter, while the operational domain of semiotic machines is completely embedded in the social space of human interaction as they aim at converting signals or data within related sign processes. The processing of semiotic machines does not leave the social space of sign processes and meaningful interaction at all. Accordingly, the *working principles* of energy or matter transforming machines are completely based on natural effects as perceived by knowledge and their *purpose* is to make use of natural forces. The working principles of semiotic machines, by contrast, are based on acting instructions derived from explicit modeling of sign or interaction processes and their purpose is to organise and coordinate collective acting.

According to these distinctions, the interpretatory flexibility in dealing with energy or matter transforming machines is bound to and constrained by natural conditions, while in case of semiotic machines it is based on habits and conventions that themselves are affected by the models and instructions implemented in the machines. Consequently, their design and use face all problems of "double hermeneutics" present in sign processes of social systems [10]. In particular, the practice of dealing with semiotic machines in organisations[10] needs to be based on the development of a sufficiently shared information space and frame of interpretation [21].

Signals and Signs: From Physics to Semantics

Signs being used for computer processing can be specified as "algorithmic signs" [17, 18]: As precise analysis reveals, the use of computers in organisations is based on two coupled sign processes interlinked by the same representamen. While interacting with the computer, humans use signs as input that are meaningful to them in their action context. Inside the IT system, these signs, being readable and meaningfully interpretable in the outside context, are reduced to pure electronic signals as their material substrate. The signals don't "know" any more for what they stand. Rather, they are being processed through a program according to the completely determined instructions of the underlying algorithm. In Peircean notation the algorithmic instructions in this sign process reduced to syntactical operations on signals take the role of an interpretant, however a "causal interpretant" that formally falls in one with the designated object (Fig. 2).

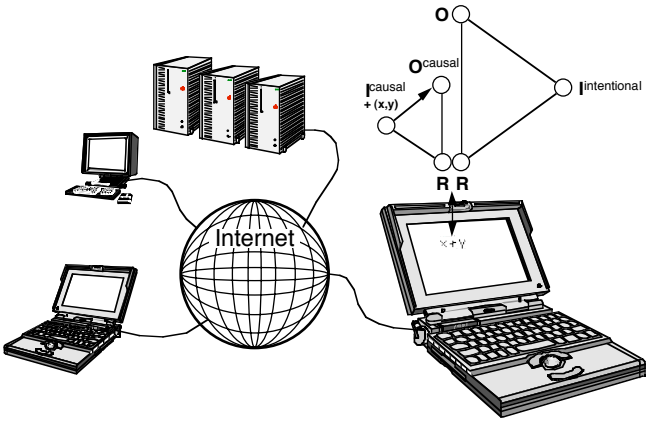

Figure 2: "Algorithmic sign" [18]: Unity of internal signal and external sign

The completely determined result of these syntactical operations on signals can, as its representamen appears on the interface, be interpreted again as sign within the social space of the action context. Consequently, computer-mediated social interaction is internally characterised by causal determination ("causal interpretant") of signal processing and externally by sense making interpretation ("intentional interpretant") of the signs associated with the signals. Inside the semiotic machine we find the effects of pure semiconductor physics and formal logic, while the events of social interaction outside are determined by semantics, the assignment of meaning in human action. The social space of sign processes in interaction has not been deserted at any time. Rather, certain aspects of social interaction are being modeled within the computer system as a sequence of program instructions or "auto-operational form" [7]. Hence, the semiotic machine can also serve as a medium of organising sign processes.

This perspective discloses the semiotic nature of software: It exists as a finite description in form of a program text, that in turn determines, as operational code, a set of sequences of signal states of the hardware. These signal processes can, as they are embedded in human action contexts, be purposefully designed and meaningfully interpreted. Accordingly, software is double-faced in nature: It is (however awkwardly) readable text on one hand, and executable operational code, i.e. a machine, on the other. This exactly is a remarkable difference to descriptions of traditional machines (drawings and parts lists) that cannot directly execute themselves as machines. As a consequence of the semiotic nature of software, its usability, irrespective of its correctness, cannot be evaluated but in the users' action context.

According to Mead, even exploratory and instrumental acting in dealing with things is of social nature: Things do exist only so far as they also exist for others. Through our intentional relationship to the world around us as well as

[10] The class of semiotic machines can be further divided into the subclasses of organisational systems and embedded systems. The latter serve as control devices for natural processes or machines in which they are embedded; they are not considered in this paper.

enabled by the action competence developed through socialisation and previous acting, we are able to assign meaning to things or events we encounter. By exploratory acting with them, we conceive their functions and comprehend how we can use them intentionally and purposefully. By remembering the action schemes and their recurring characteristics, we form classes or concepts of objects or events in the outside world. By acting and interacting with others in a shared world, we "create" the things and ourselves, seeing them as taken for granted [16].

Mental reflections on our acting and its conditions are caused only, if hindrances or surprises occur in the flow of acting. Such action problems lead to a situation in which the things taken for granted are losing their "objectivity", since objectivity is not naturally given, but ascribed through shared understanding. Obstacles in acting trigger a reflection and search process in order to re-establish the "vanished object" and to regain the capacity to act (cf. the notions of "break-down" and "reflection-in-action" with Schon [28]). However, the experienced disorientation in such acting crises not only relates to the object, but also concerns the acting person itself. In the moment of uncertainty not only the world outside, but also the own power of judgment is being questioned. The acting person is unable to "distinguish between subject and predicate": "I want to emphasise that, as long as we don't have a predicate, we also don't have a subject" [16]. Nevertheless, through such processes of reflecting we can regain the capacity of fluid acting. This capacity includes the ability to anticipate the functions and properties of things learned from previous actions and to organise own actions according to the anticipated "thing behaviour" (cf. the notion of "situated action" with Suchman [29]).

Software is Orgware

By virtue of the Peircean concept of sign and the Meadan comprehension of dealing with things one gets seamless access to modern theories of social systems that mediate between the views of subjective acting and objective acting structures and that can, in particular, appropriately explain both inertia and dynamics of collective acting in organisations. In this theoretical perspective, organisations emerge and reproduce themselves as social systems through the continued sense making, mutually related and coordinated acting of their members which itself is based on grown routines and assumed expectations.

In the course of their continuous action flow, actors may generate explicit knowledge through reflection and concept formation about certain aspects of their experiences in acting and in dealing with things as described. This knowledge can be expressed and objectified again in the form of linguistic signs, of organisational schemes or of technical artifacts. In particular, technical systems like computer artifacts can thus be designed as a product of reflection on human activities, as objectified explicit knowledge by modeling certain courses of practical action.

This model formation, in principle, undergoes the following three steps of abstraction and formalisation:

- *Semiotisation*: describing courses of action by signs as a prerequisite for communication (result: *application model*);
- *Formalisation*: abstracting from interpretations bound to situation and context by using standard signs and operations (result: formal model, *specification*);
- *Algorithmisation*: describing courses of action as formally computable procedures by means of the standard signs and operations (result: *computing model*).

In this way, computer artifacts emerge as objectified propositional knowledge about purposeful human acting. They are, as such, used again as means for further acting. As "congealed knowledge" inscribed in their functions and properties, they embody aspects of human practice, and as means of work to practical ends they set specific action requirements for effective use for which they must be appropriated again. Appropriation for skillful and effective use thus constitutes a new practice, new ways of doing things ([2]; cf. Fig. 3). Since they are derived from abstract, decontextualised knowledge, technical artifacts always contain empty "slots" that have to be filled in use through "recontextualisation", i.e. by interpretation and application suited to the situation. As a consequence, their use value is constituted in the application that is, due to the scope of interpretation within the limits of the action requirements, open for diverse use.

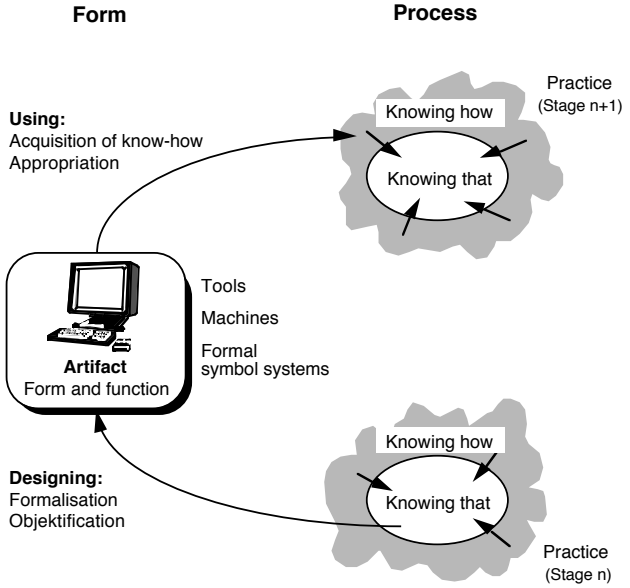

Figure 3: Genesis and use of technical artifacts: Dialectics of form and process

By routinely enacting the artifacts' forms and functions in use, they 'structure' human action, and in this way they become implicated as rules and resources in the constitution of a particular recurrent social practice. Through such recurrent interaction with the artifact at hand, certain of the

artifact's properties become implicated in an ongoing process of structuration in which rules and routines of using it emerge. The resulting recurrent social practice produces and reproduces a particular structure of technology use [20|. Consequently, the design and use of technical artifacts have to be regarded as integral part of social systems' dynamics and, hence, the development of organisational practices.

According to this dialectics of expressive form (objectified knowledge) and process (appropriation for use), technical acting, the interaction with computer artifacts to accomplish a given task, can be understood as a process of "social construction of reality" [8]. Since the meaning of an artifacts' functions is created through interpretation in the process of acting with them, they can also be interpreted by others acting in the same action context. Successful and mutually confirmed acting thus leads to a shared understanding among the co-workers. Like practicing a language or organisational acting, computer artifacts, thus, are socially embedded in sign processes. In all these activities conceptual knowledge is externalised or objectified as forms – be they technical artifacts, language terms or organisational schemes – together with emerging rules how to interpret and how to sensibly act with them.

The externalised forms, in turn, can be used as resources for further acting; they even enable or allow for new ways of acting, if interpreted differently. As far as the rules of acting with them are being appropriated and internalised, they establish, together with the objectified forms they refer to, a new practice. It is these mutually shared (but mostly unconscious) rules (the formative context) that enable the actors to appropriately interpret situations or facts as well as data, instruments or instructions, in short: to fluently act in the organisational environment.

The expressive forms as resources together with the rules to deal with them, i.e. the attitudes, values, ways of thinking and acting, and schemes of interpretation, constitute a social structure that enables and, at the same time, constrain collective acting ("duality of social structure"). What the actors in an organisation can imagine and which opportunities to act they see in a given situation thus depends on the expressive forms they created as well as on the interpretative rules they developed to deal with them. The actors thus are socially constructing their reality, however not of their own free will, but as prisoners of the conditions they have developed to enable and regulate their collective acting. By making sense of resources at hand through interpretation (*signification*), by sanctionising actions according to norms (*legitimation*), by determining administrative resources from formal organisation or by prescribing the use of technical artifacts (*domination*), each time in these social practices they create rules that constrain the scope for future action and negotiation. The better the expressive forms are adjusted to the action context and the more appropriately they are interpreted, the more effective the social practice of collective acting can develop (2, 10, 20, 22]; see Fig. 4).

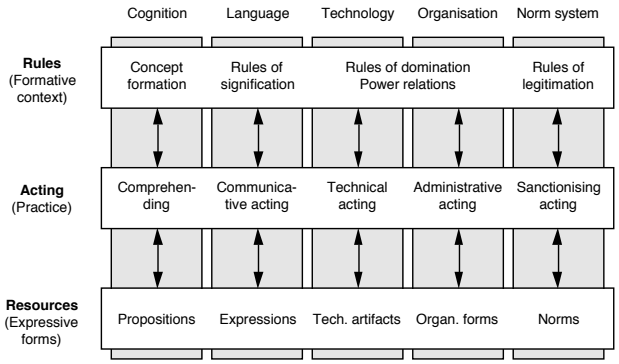

Figure 4: Structuration: Mutual constitution of acting and social structure

A paramount consequence of the semiotic nature of computer artifacts and their embeddedness in sign processes of social interaction is the indispensable fact of "double hermeneutics" [10]. In contrast to natural sciences, where (with the exception of quantum mechanics) cognition and the object of cognition are independent of each other, in social sciences observations do change their own object of observation. Hence, the object of observation, the social system, is reflexive in the sense that the explicit knowledge gained about the system – as well as the technical artifacts derived from that knowledge – becomes part of the system's resources and rules being changed by this. Social scientists, like system designers, have to interpret features of a social system as object of observation, in which they themselves take part as observers. Their thinking belongs to the same system they think about.

Formalisation and algorithmisation as central computing science activities of system analysis, modeling and design exactly are such events of observation that change the object of observation. Sign processes observed and modeled in this way, therefore, are being changed by exactly these activities: The object of modeling undergoes change by the process of modeling itself – a fact that has been almost neglected so far in software engineering with fatal consequences.

Moreover, the development of technical artifacts (and of software in particular) so far has been predominantly concentrated on processes of design according to functional requirements and almost neglected the reverse process of appropriation and enactment for effective use. However, the skill to make sense of the artifacts, to find adequate interpretations for accomplishing the working tasks is at least of equal importance and requires creative acting as well [25]. And the collective learning efforts necessary for the effective appropriation and enactment are much more expensive than design.

These are, according to the theoretical perspective presented here, the main reasons for failure in IT application projects. The following section focuses on practical consequences that can be derived from this.

CONCLUSIONS FOR IMPROVING PROJECT PRACTICES

A number of conclusions can be drawn from the semiotic perspective on computer artifacts with respect to self-comprehension of computing science as a discipline on one hand and with respect to effective improvements in the social practices of implementing and using IT systems in organisations on the other.

First, the semiotic perspective opens the mind for a new comprehension of computing science as a discipline of *technical semiotics* which allows to conceive IT systems as signal processing artifacts embedded in the sign processes of social interaction. As such they serve, provided that they are appropriately designed, adopted and enacted, as media for organising work or value creation and knowledge transformation processes. On the basis of the triadic sign concept by Peirce, it can also bridge the gap to modern sociological theories of organisations in order to gain a holistic view and integrated procedures on system design and organisational development.

Second, the semiotic perspective, thus, also delivers the key for understanding the reasons behind the permanent software crisis and the IT productivity paradox. As digital devices and media for organising, IT systems are not just models or representations of work processes but rather serve as supportive technical artifacts that, in the course of organisational development, must be co-designed, appropriated and enacted for effective use together with other organisational resources in the social practices of an organisation. Due to the self-referential nature of these activities, the social practices are themselves changed by this. Consequently, the effects produced are not solely dependent on the implemented system functionality, but are a result of how they have been socially embedded and enacted for practical use. System quality can, therefore, only be evaluated in the context of practical use.

Third, as, a consequence of this, it is indispensable to involve end users in design and implementation of both the technical and the organisational features of the new work system from the beginning. As designers normally have only little understanding of the real work tasks and procedures and users have only little knowledge about the options IT has to offer for organisational redesign, both main actors in the design and implementation process must cooperate. In order to overcome their symmetrical ignorance, they are compelled to develop a shared understanding of the underlying work processes and frame conditions. A number of practically proven methods exist to support user participation including future workshops, design scenarios or social simulation and rapid prototyping.

Fourth, as the design of IT systems is a reflexive endeavour in the sense that the systems' appropriation and use change the work processes they are designed for, frequent changes of functional requirements during system design and implementation are inevitable. Software engineering methodology, therefore, must cope with this inescapable fact and organise design and implementation processes in a reflexive or evolutionary way with iteratively revised and improved versions of the system or its modules. This requires sound methods for software engineering and project management that combine aspects of modular design, formative evaluation and collective learning with constrained range in order to confine the risks. Moreover, project management must conceive and organise the joined evolutionary design, implementation and appropriation efforts as integral part of organisational development.

Fifth, all actors involved must realise the fact that implementation and use of IT systems have strong impact on the balance of organisational flexibility and rigidity. All human acting must be sufficiently supported by routines in order to be fluent and efficient. Formal organisational procedures and routines, therefore, help to organise efficient collective acting. It actually is the purpose of organisations to reduce contingency and to confine the space of communication by rules, routines and formal procedures. And as IT systems, by definition, operate on the basis of completely determined procedures in form of algorithms, they appear as a most appropriate organisational medium. However, as they in turn impose rigid action requirements on the users working with them, they may overly constrain the necessary flexibility in action that is needed to cope with uncertainties and surprises in the organisation's environment. Hence, the actors must, during the process of integrated organisational redesign and system implementation, find a reasonable balance in this field of conflict between flexibility and rigidity.

In sum, taking these considerations together, the institution of the UsersAward and the procedures around it appear as an adequate approach to raise the consciousness for the problems presented, to intensify communication between suppliers and users and to improve the usability of systems.

REFERENCES

1. Brödner, P., 2002: Der Held von Caputh steht nicht allein. Wie Wissenschaft die Nutzungsprobleme der Informationstechnik ignoriert, in: Moldaschl, M. (Hg.): *Neue Arbeit – Neue Wissenschaft der Arbeit?* Heidelberg: Asanger 2002, 339-364

2. Brödner, P., 1997: *Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen*, Berlin: edition sigma

3. Brynjolfsson, E. (1993): The Productivity Paradox of Information Technology, *CACM 36*, No. 12, 67-77

4. Brynjolfsson, E.; Hitt, L.M., 2000: Beyond Computation: Information Technology, Organizational Transformation and Business Performance, *Journal of Economic Perspectives 14 (4)*, 23-48

5. Davenport, T.H. (1998): Putting the Enterprise into the Enterprise System, *Harvard Business Review* July-August, 121-131

6. Farrell, D. (2003): The Real New Economy, *Harvard Business Review* October, 105-112

7. Floyd, C., 2002: Developing and Embedding Autooperational Form, in: Dittrich, Y.; Floyd, C.; Klischewski, R. (Eds.), 2002: *Social Thinking – Software Practice*, Cambridge (MA): MIT Press, 5-28

8. Floyd, C., 1992: Software Development as Reality Construction, in: Floyd, C.; Züllighoven, H.; Budde, R.; Keil-Slawik, R. (Eds.): *Software Development and Reality Construction*, Berlin Heidelberg New York: Springer, 86 – 100

9. Gibbs, W.W., 1994: Software: chronisch mangelhaft, *Spektrum der Wissenschaft*, Dezember 1994, 56-63

10. Giddens, A., 1984: *The Constitution of Society. Outline of the Theory of Structuration*, Cambridge: Polity Press

11. Kapor, M., 1996: A Software Design Manifesto, reprint in Winograd, T. (ed.): *Bringing Design to Software*, Reading (MA): Addison-Wesley

12. Landauer, T.K., 1995: *The Trouble with Computers. Usefulness, Usability, and Productivity*, Cambridge (MA): MIT Press

13. Maucher, I. (2001): *Komplexitätsbewältigung durcph Entwicklung und Gestaltung von Organisation*, München: Hampp

14. Maucher, I. (Hg.) (1998): *Wandel der Leitbilder zur Entwicklung und Nutzung von PPS-Systemen*, München: Hampp

15. McKinsey Global Institute (2001): Productivity in the United States, http://www.mckinsey.com/knowledge/mgi/reports/productivity.asp

16. Mead, G.H., 1903: Die Definition des Psychischen, in: Gesammelte Aufsätze, hg. von Hans Joas, Bd. 1, Frankfurt / M: Suhrkamp 1987, 83-148

17. Nake, F., 2001: Das algorithmische Zeichen, in: Bauknecht, W.; Brauer, W.; Mück, T. (Hg.): *Informatik 2001. Tagungsband der GI/OCG Jahrestagung*, 736-742

18. Nake, F.; Grabowski, S., 2001: Human-Computer Interaction Viewed as Pseudo-Communication, *Knowledge-Based Systems 14*, 441-447

19. Ogden, C.K.; Richards, I.A., 1989: *The Meaning of Meaning* (reprint of 1923 edition), San Diego New York London: HBJ

20. Orlikowski, W.J., 2000: Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations, *Organization Science 11 (4)*, 404-428

21. Orlikowski, W.J.; Gash, D.C., 1994: Technological Frames: Making Sense of Information Technology in Organizations, *ACM Transactions on Information Systems 12*, 174-207.

22. Ortmann, G.; 1995: *Formen der Produktion. Organisation und Rekursivität*, Opladen: Westdeutscher Verlag

23. Oz, E., 1994: When Professional Standards are Lax. The CONFIRM Failure and its Lessons, *CACM 37*, No.10, 29 - 36

24. Peirce, C.S., 1903: *A Syllabus of Certain Topics of Logic*, Collected Papers, 1.180-202, 2.219-225 and other paragrafs; German translation: Phänomen und Logik der Zeichen, Frankfurt/M: Suhrkamp 1983

25. Pipek, V., 2005: *From Tailoring to Appropriation Support: Negotiating Groupware Usage*, Acta Universitatis Ouluensis A 430, Oulu: Oulu University Press

26. Royal Academy of Engineering (ed.), 2004: *The Challenges of Complex IT Projects*. The report from a working group of the Royal Academy of Engineering and the British Computer Society, London: The Royal Academy of Engineering

27. Scott, E.J.; Vessey, I., 2002: Managing Risks in Enterprise Systems Implementations, *CACM 45* April, 74-81

28. Schon, D.A., 1983: *The Reflective Practitioner: How Professionals Think in Action*, New York: Basic Books

29. Suchman, L., 1987: *Plans and Situated Actions. The Problem of Human Machine Communication*, Cambridge (MA): Cambridge University Press

30. The Standish Group International, 2001: Extreme CHAOS. The 2001 update to the CHAOS report, http://www.standishgroup.com/sample_research/index.php