Comprehensive Summaries of Uppsala Dissertations
from the Faculty of Science and Technology 981

User-Centred Systems Design

Designing Usable Interactive Systems in Practice

BY

BENGT GÖRANSSON

ACTA UNIVERSITATIS UPSALIENSIS
UPPSALA 2004

Dissertation presented at Uppsala University to be publicly examined in Room 211, MIC, Polacksbacken Hus 1, Uppsala, Tuesday, June 1, 2004 at 13:15 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract

Göransson, B. 2004. User-Centred Systems Design. Designing Usable Interactive Systems in Practice. (Användarcentrerad systemdesign. Design av användbara interaktiva system i praktiken). Acta Universitatis Upsaliensis. *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 981. 80 pp. Uppsala. ISBN 91-554-5975-7

Have you ever been frustrated with that IT system at work that does not behave the way you expect it to? Or had problems with using the features on your new mobile phone? When systems and appliances do not support us in what we are doing, and do not behave the way we expect them to, then usability is neglected. Poor usability may be frustrating and irritating when trying out your mobile phone, but in a critical work situation poor usability may be disastrous.

In this thesis, user-centred systems design (UCSD) is advocated as an approach for facilitating the development of usable interactive systems. Systems that suit their intended use and users do not just "emerge". They are the result of a UCSD *process* and a user-centred *attitude* during the development. This means in short that the real users and their needs, goals, context of use, abilities and limitations, drive the development – in contrast to technology-driven development. We define UCSD as: a process focusing on usability throughout the entire development process and further throughout the system life cycle. I argue that this definition along with a set of key principles do help organisations and individual projects in the process of developing usable interactive systems. The key principles include the necessity of having an explicit focus on users and making sure that users are actively involved in the process.

The thesis provides knowledge and insights gained from real-life situations about what UCSD is and how it can be put into practice. The most significant results are: the proposal of a clear definition of UCSD and a set of key principles encompassing UCSD; a process for usability design and the usability designer role. Furthermore, design cases from different domains are provided as examples and illustrations.

Keywords: user-centred systems design, usability, human-computer interaction, system development, user involvement, usability design

Bengt Göransson, Department of Information Technology, Box 337, Uppsala University, SE-75105 Uppsala, Sweden

ISSN 1104-232X
ISBN 91-554-5975-7
urn:nbn:se:uu:diva-4273 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-4273)

For Ellinor, Carl and Emma with love

List of Papers

This thesis is made up of two parts: first a *summary* of my research and then the following *papers*:

Paper 1

Key Principles for User-Centred Systems Design

Gulliksen J., Göransson B., Boivie I., Blomkvist S., Persson J. & Cajander Å. (2003)

Published in an international journal: Special section “Designing IT for Healthy Work” in Behaviour & Information Technology, November–December 2003, Vol. 22, No. 6, pp. 397–409, Taylor & Francis, www.tandf.co.uk.

Paper 2

The Usability Design Process—Integrating User-Centred Systems Design in the Software Development Process

Göransson B., Gulliksen J. & Boivie I. (2003)

Published in an international journal: Software Process: Improvement and Practice (SPIP), vol. 8, issue 2, Wiley & Sons.

Paper 3

Usability Design—Extending Rational Unified Process with a New Discipline

Göransson B., Lif M. & Gulliksen J. (2003)

Presented at a conference and published as a revised paper in a book: Jorge J., Nunes N. & Cunha J. (eds.), Interactive Systems: Design, Specification and Verification, 10th International Workshop, DSV-IS 2003, Funchal, Madeira Island, Portugal, June 2003, Revised Papers, LNCS 2844, ISBN 3-540-20159-9 Springer-Verlag Berlin Heidelberg New York, pp. 316–330.

Paper 4

The Lonesome Cowboy – A Study of the Usability Designer Role in Systems Development

Boivie I., Gulliksen J. & Göransson B. (2004)

Submitted to an international journal: Interacting with Computers: The Interdisciplinary Journal of Human-Computer Interaction, Elsevier, http://www.elsevier.com/locate/issn/0953-5438.

Paper 5

A User-Centred Approach to Object-Oriented User Interface Design

Gulliksen J., Göransson B. & Lif M. (2001)

Published as a book chapter: Mark van Harmelen (ed.), Designing Interactive Systems: Object Modeling and User Interface Design, Addison-Wesley: Boston, ISBN 0-201-65789-9.

Paper 6

Reengineering the Systems Development Process for User Centred Design

Gulliksen J. & Göransson B. (2001)

Published in conference proceedings: The IFIP TC.13 International Conference on Human-Computer Interaction, 9th – 13th July 2001, Tokyo, Japan. Michitaka Hirose (ed.) Human-Computer Interaction, INTERACT ‘01, IOS Press, Amsterdam, pp. 359–366.

Paper 7

The Re-Design of a PDA-based System for Supporting People with Parkinson’s Disease

Göransson B. (2004)

Accepted for publication in conference proceedings: The 18th British HCI Group Annual Conference, Leeds Metropolitan University, UK 6–10 September 2004.

Paper 8

The Design of a Smart Card-Based Home-Help System

Frisk A., Göransson B., Sandbäck T. & Thomasson V. (2001)

Published in conference proceedings: The IFIP TC.13 International Conference on Human-Computer Interaction, 9th – 13th July 2001, Tokyo, Japan. Michitaka Hirose (ed.) Human-Computer Interaction, INTERACT ‘01, IOS Press, Amsterdam.

Paper 9

A Teleradiology System Design Case

Borälv E. & Göransson B. (1997)

Published in conference proceedings: Gerritt van der Veer, Austin Henderson, Susan Coles (eds.): Designing Interactive Systems: Processes, Practices, Methods and Techniques. DIS’97 Conference Proceedings of the ACM Special Interest Group in Computer-Human Interaction (SIGCHI) in co-operation with the International Federation for Information Processing (IFIP WG 13.2), Amsterdam August 18–20, pp. 27–30.

Reprints were made with permission from publishers where applicable. The papers will in the summary be referred to as “paper 1”, “paper 2”, etc.

Author’s Guide to the Reader

This thesis is about how to develop usable interactive systems through practicing user-centred systems design (UCSD). The summary primarily contains a background to why I consider a user-centred approach essential and a discussion of my research. It also provides some guidance to UCSD.

About the papers: paper 1 provides an account of what it takes to work in accordance with a user-centred philosophy. Paper 2, paper 3 and paper 4 are process-oriented and primarily intended for readers working in project teams and at method departments who wish to apply a UCSD process in practice. Paper 5 and paper 6 highlight experiences of integrating UCSD into organisations and systems development processes. Paper 7, paper 8 and paper 9 are all design cases. They focus on design and the use of certain design patterns. They also briefly describe the tailored UCSD process applied in each of the projects. I included these papers since they provide important background and accounts of how practitioners work.

About my co-authors

I have worked with a great number of people over the years and found it inspiring to write together with people coming from different disciplines as well as having different professions. My co-authors’ backgrounds and jobs reflect the multidisciplinary nature of my research and my desire to combine research with practice.

My co-authors from the Department of Information Technology, Human-Computer Interaction, Uppsala University, Sweden are: *Stefan Blomkvist*, *Inger Boivie*, *Erik Borälv*, *Åsa Cajander*, *Jan Gulliksen* and *Jenny Persson*.

From the industry: *Anders Frisk*, *Magnus Lif*, *Torsten Sandbäck* and *Vello Thomasson*.

Table of Contents

Innovation starts with people, not enabling technologies or manufacturing plans or distributor preferences. If you forget this you risk delivering feature-rich rubbish into already over-crowded lives.

Richard Seymour, Design in Business Week, 2001

Introduction

> *The high-tech industry is in denial of a simple fact that every person with a cell phone or a word processor can clearly see: Our computerized tools are too hard to use. The software engineers who create them have tried as hard as they can to make them easy to use and they have made some minor progress. They believe that their products are as easy to use as it is technically possible to make them. As engineers, their belief is in technology, and they have faith that only some new technology, like voice recognition or artificial intelligence, will improve the user's experience.*
>
> Alan Cooper "The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity", 1999, p. 15.

My ultimate goal as a researcher and professional is to facilitate the development of *usable interactive systems* – systems that support and engage people, rather than get in the way and hinder them. Systems that suit their intended use and users do not just "emerge". They are ultimately products of a user-centred development *process* and a user-centred *attitude* during the development. This means, in short, that the *real users* and their needs, goals, context of use, abilities and limitations drive the development – in contrast to technology-driven development. Developing usable products is about meeting the real needs of the users, rather than basing the design on what is possible to do from a technical perspective.

My approach for accomplishing this goal is to work both as a researcher and a practitioner applying *user-centred systems design* (UCSD). Putting research into action—*action research*, and acting as a *reflective practitioner*, provide opportunities to communicate best practices and results to other practitioners, and to researchers for further exploration.

My research contributions are detailed and discussed in the chapter *Guidance to UCSD and Discussion on Lessons Learnt*. In brief they cover: knowledge and insights about what UCSD is and how it can be put into practice, systematically gained from real-life situations; the proposal of a clear definition of UCSD and a set of key principles for UCSD; a process for usability design and the usability designer role.

There are several reasons for doing research on UCSD. Although the concepts of UCSD and usability have, in a broad sense, been recognised and adopted by the industry, applying UCSD in practice is still an area where more research is needed. UCSD has been on the agenda for about 20 years or so, primarily within research (Karat & Karat, 2003), but lately to an increasing extent within the industry as well. Nevertheless, it is not yet clear how UCSD should be applied in real development projects. There seems, for instance, to be no commonly agreed-upon understanding of usability and UCSD. Is this because it is not necessary to have a common understanding or common ground? Or, is it because we take things as "easy to understand", "easy to use" and "easy to learn" for granted? Or, is it because we interpret "easy" in different ways depending on our education, profession or cultural background? I strongly believe that there is an advantage and strength in sharing a *common ground*. This is not to say that there is always one "correct" way of doing things, but when talking about different approaches there are good reasons for being clear on the subject, avoiding misunderstandings and "political" discussions. I believe that UCSD has the potential to create a common ground in the development of usable interactive system. It can also provide a powerful knowledge base and framework for organisations, projects as well as individual stakeholders.

In my experience, the industry is truly interested in using user-centred methods and processes for producing usable interactive systems. This has also been described in several independent reports over the years. One example is Katzeff & Svärd (1995). Their investigation showed that the Swedish industry was not particularly mature regarding usability in interactive systems. However, the companies in the study were very interested in learning more about the area. They ranked the development of methods and processes highly, together with the integration of usability-related activities into their development process. Later research such as Vredenburg et al. (2002) reports that usability practitioners feel that user-centred methods are: "[…] gaining momentum across the industry and that they will likely achieve even wider use and greater impact in the next five years". (ibid p. 478) Evidently, there is a time factor to consider. It takes time for new practices to become adopted and also to become adapted to a special context. Of course, the overall picture is diverse. Different sectors of the industry have adopted and started to practice UCSD in many different ways.

In my research and practice I have observed that companies and organisations in Sweden approach the challenge of developing usable interactive systems in some different ways:

- Many organisations do not see the need for paying any particular attention to usability. They believe that usability can be addressed by means of some minor hands-on activities, such as adding graphics to the user interface.

- Others incorporate user-centred activities into their current development practice, i.e. they add user-centred activities to whatever development process they are using, but do not make any major changes to it.
- A few companies and organisations are willing to “go all the way,” so to speak, and adopt a truly user-centred process. These companies and organisations realise that being committed to usability and UCSD requires a major shift of focus in their development process. They take a stance and make a commitment to a user-centred philosophy.

These different approaches have been described in similar ways by others. Deborah Mayhew (2002) proposes a simplified usability maturity model in which the approaches described above fit. This model proposes a number of stages where organisations go from little awareness (promoting) of usability, through establishing usability, to usability being institutionalised. Dray and Siegel (1998, p.16) report that: “[…] companies tend to adopt UCSD practices and methods in stages or adopt a particular method or practice only when a complex set of factors align to create readiness”. When companies and organisations now discover UCSD and usability, they sometimes tend to repeat mistakes made by others. In his book, *The inmates are running the asylum* (1999), Cooper describes some of the problems with a great deal of insight and humour: “Programmers trade simplicity for control. They exchange success for understanding. They focus on what is possible to the exclusion of what is probable” (ibid, p. 93). A user-centred development process focuses on what is adequate and usable for the user, and not on what is technically possible to develop. Other challenges include taking the full meaning of usability into account and introducing a process for that, instead of focusing on isolated parts, such as the user interface.

Institutionalising UCSD, making people and organisations realise its full potential and obtaining acceptance for it, is complex and time consuming. I consider it a process of change, in that it requires changes in work practices as well as in attitudes. This process involves many factors. These factors are diverse in nature, with presumably different backgrounds. But, in one way or another, they all point to the need for focusing on humans or users in the systems design process. Very often people ask for reasons for applying UCSD in their organisation; what are the benefits of using a UCSD process, etc. In the next section I discuss and group the reasons that I have come across most frequently. I use these reasons, both in my role as a researcher and as a practitioner, as a way of providing incentives for UCSD and illustrating how UCSD can play a role in systems development.

Reasons for UCSD

I argue that the best reason for developing usable interactive systems is *having users benefit from a better work situation*. This means, for instance, that the users are in control of their work; they are more efficient; they experience less stress; they run into fewer errors leading to increased quality of service; they can improve their work skills; they enjoy work and social communication in the workplace is encouraged. But often this is not enough. CEOs, managers, procurers, business people, project managers, etc. need other arguments to adopt a user-centred approach. Often their goals and agendas are based on economic rationales, short-term as well as long-term. Issues, such as, the return of investment (ROI) are important when considering an investment in an information system. One major problem when estimating the ROI is that it is almost impossible to know all the factors that will be affected by the investment. Sometimes it seems that the decision to invest in an information system is based partly on economic facts and partly on a general feeling that the company or organisation needs a new system. Many of the arguments for UCSD have their origin in the belief that we need to understand users better and need to involve them in systems development. If we focus on the users and involve them in the design process early on, it is logical to assume that the outcome (the system) will suit their needs better than if we do not focus on them. This implies that making an investment up front will give more in return in the long run. UCSD is often thought of as costly in terms of budgeted money for a project. It does to some extent involve activities that are usually not part of a systems development process, e.g. studying people at work, prototyping with users, trying out designs, evaluating with users and iterating solutions. These activities take time and require extra effort. But, simply seeing this as an additional cost means that you do not understand the benefits and advantages that can be gained in the long-term perspective. A user-centred approach is valuable for a number of reasons, including long-term economic aspects. These aspects can act as driving forces in adopting UCSD and provide guidance on how the area of UCSD may be further explored through research and practice. They include:

- Economic; opportunities for decreasing costs and increasing earnings.
- Business and organisational; UCSD broadens the view of systems development and may facilitate the integration of the IT systems with the business.
- Quality; in systems and in the development process.
- Legal; laws and regulations, e.g. occupational health and safety legislation.
- Ethics and moral; developing systems to be used by people also involves “soft” aspects, such as, basic human values.

The Economic Reason

> *After the New York Stock Exchange upgraded its core trading systems using user-centred design techniques, productivity rose dramatically and users' error rates fell by a factor of 10 even though workloads more than doubled.*
>
> Cited in Gibbs, W W ,1997. Taking Computers to Task. Scientific America, July 1997.
>
> *Norwich Union, an insurance company in Australia, found that calls to its help desk reduced dramatically by two thirds after one of its core applications was improved using user-centred design techniques.*
>
> Norwich Rethinks Customer Service, Computer World, 24 November 1995.

There are numerous reports and quotes like the ones cited above, describing how much money can be saved by means of a user-centred approach. Such reports are often called *cost justification* and intended to demonstrate that by spending a little bit extra on user-centred activities, you can gain much more in the long run[1]. Cost justification is important in some situations, and contexts, but the examples are really only snap-shots taken of a multifaceted environment. It is in fact very difficult to correlate overall effects to a single development activity, or a sequence of activities. However, cost justification examples provide one piece in a "puzzle" when promoting UCSD. Other, often cited, potential savings include reduced costs for training and producing training materials. Other factors that may improve business include users spending less time on figuring out how to use a system and providing users with user interfaces adapted to tasks such as qualified decision support. Reduced irritation and a less stressful situation for users are also likely to boost the efficiency.

In his dissertation Åborg (2002) discusses the increasing use of computers in the workplace, and the related increase of various negative effects on the users' health. The health complaints range from physical problems such as the "mouse arm-syndrome" to stress and mentally related symptoms. Despite the positive effects on productivity that information technology (IT) may have, it seems that the overall positive effects can be questioned (ibid). Poor usability of computer systems is described as one of the underlying causes. The costs of occupational health problems are enormous. The industry as well as the public sector has to bear the costs for absence due to illness and decreased productivity. Thus, there is a huge potential to improve the health of the people affected by the problems as well as to save money.

Saving money is only one part. Making money is equally important. There seems for instance to be a close connection between earnings and usability in

[1] A collection of case studies and quotes on usability and ROI is provided by Aaron Marcus (2002) in his report "Return on Investment for Usable User-Interface Design: Examples and Statistics".

e-commerce sites on the Web. Figures about the potential earnings from e-commerce sites are frequently communicated. Some examples include:

- One study estimated that improving the customer experience increases the number of buyers by 40 % and the order size by 10 %. (Creative Good, 2000)
- Two studies have shown that the user success rate in purchasing from e-commerce websites is in the range of 25–60 %. Small improvements in user performance could lead to substantial increases in revenue. (PRUE, 2002)
- You can increase sales on your website by as much as 225 % by providing sufficient product information to your customers at the right time. And further, by understanding your customers' expectations and needs, and designing your product lists accordingly, you can significantly increase your sales. (UI Engineering, 2001)

When considering the comparatively short period of time that the Web has been around, the potential of e-commerce and publicly available on-line services has just started to be utilised. The majority of users have not yet explored the possibilities and the advantages of such websites (see Figure 1). The potential to attract large numbers of users (i.e. the early and late majorities in the figure) will increase for websites that are usable and developed with a user focus. While there is no indisputable evidence that usability will lead to increased figures for visits and sales, I believe that it will at least help in making such increases more probable.

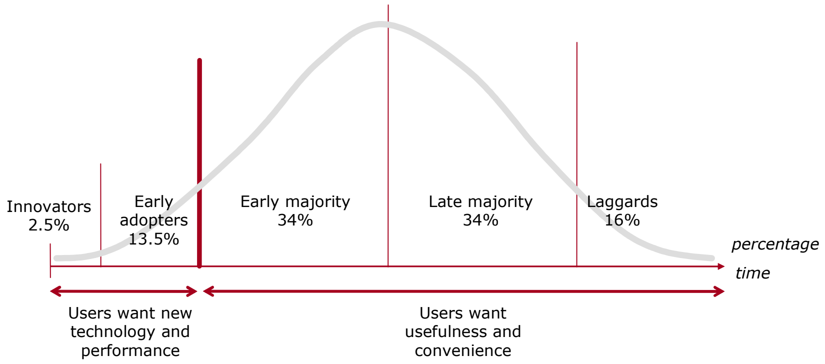


Figure 1: How individuals seem to adopt innovations. A majority of individuals (users) benefit from innovations only when they have proven to be useful and are convenient to use. (Adapted from Rogers E.M. & Scott K.L., 1997 and partly Nielsen, 1993, pp. 265—267)

Moreover, having a product that is more usable than competing products provides an increased competitive edge and will probably result in satisfied users and customers who come back (repeated sales).

However, economic effects are typically difficult to isolate. They depend on, for instance, the business model of the company and the current market situation. Systems developers, like myself, can make no promises. We can only put forward the opportunities and possibilities, but it is up to the company to make things happen.

Business and Organisational Benefits

One major business and organisational benefit is that UCSD encompasses more than "simply" the construction of an IT system. UCSD broadens the view on systems development and has the potential of integrating the business, organisation and the supporting IT systems. During analysis and design in UCSD, the current and the future use situation, as well as the users' needs, assumptions and expectations are explored. This gives a much richer picture of how business and organisational aspects can be taken into account in systems development, than when using technology-driven and/or engineering-oriented approaches. Work practices can be enhanced and improved if users are involved from the start. In addition, user involvement facilitates the deployment of the system and user acceptance. Not just because the users have been able to influence the design and development of the system, increasing their sense of ownership, but also because they are more likely to accept and use a system that fits into their work situation.

Quality in Systems and Processes

Developing computer systems is a risky business. We know from reports that development projects are more likely to fail than to succeed (see for example Smith & Keil, 2003; Standish Group, 1995). The CHAOS report on success factors in projects states that the number one criterion for success is *user involvement* (ibid, 1995)[2]. Even though user involvement does not guarantee project success, it is a good argument to consider. There are also quantitative factors pointing to the impact of user involvement:

> *Eighty percent of software life cycle costs occur after the product is released, in the maintenance phase. Of that work, 80 % is due to unmet or unseen user requirements only 20 % of this is due to bugs or reliability problems.*
>
> Karat, 1993

> *The rule of thumb in many usability-aware organizations is that the cost-benefit ratio for usability is $1:$10–$100. Once a system is in development, correcting a problem costs 10 times as much as fixing the same problem in design. If the system has been released, it costs 100 times as much relative to fixing in design.*
>
> Gilb, 1988 referenced in Marcus, 2002

[2] The Standish Group defines successful projects as completed on time and on budget, with all features and functions as initially specified.

These figures should be seen as indications rather than scientific data. But it is quite clear that using a user-centred process has the potential to improve both the quality of the development process and the quality of the system—the outcome of the process. A user-centred approach provides the organisation and the project with a better understanding of the users' real needs, and how to meet these needs. Given that, I would like to argue that is more likely that a user-centred process will produce systems that better match the needs of the business and the users than one developed using a technology-driven approach. The developers will have a much better understanding of what functionality is needed and will be able to focus on those functions that are important for the users. The development can therefore be carried out more effectively and with less effort. This will ultimately reduce costs and risks.

The Legal Reason

The European Directive on Display Screen Equipment (implemented in the national legislation of the EU countries) is primarily concerned with the physical working environment and working conditions. However, it also includes requirements such as "Software must be suitable for the task" and "Software must be easy to use and where appropriate adaptable to the user's level of knowledge or experience" (Bevan 1991). In Sweden these directives are implemented by the Swedish Work Environment Authority through legislation such as regulations on work with display terminals (Swedish Work Environment Authority, 1998).

There are also further legislations promoting user-centred methods and techniques stated in the Work Environment Act (Swedish Work Environment Authority, 2001) in chapter 2 and section 1: The State of the Working Environment:

- The employee shall be given the opportunity to participate in the design of his own work situation and in processes of change and development affecting his work.

In the U.S., the Department of Justice has issued legislation on accessibility to electronic and information technology for disabled. This is the so-called "Section 508 Electronic and Information Technology" of the Workforce Investment Act of 1998. It requires that the U.S. Federal agencies' electronic and information technology is accessible to people with disabilities, including employees and members of the public. Furthermore, the U.S. Department of Health and Human Services, Food and Drug Administration (FDA) has issued "General Principles of Software Validation; Final Guidance for Industry and FDA Staff". In section *5.2.3. Design* it is stated that:

- The software design needs to address human factors. Use error caused by designs that are either overly complex or contrary to users' intuitive

> expectations for operation is one of the most persistent and critical problems encountered by FDA. Frequently, the design of the software is a factor in such use errors. Human factors engineering should be woven into the entire design and development process, including the device design requirements, analyses, and tests. Device safety and usability issues should be considered when developing flowcharts, state diagrams, prototyping tools, and test plans. Also, task and function analyses, risk analyses, prototype tests and reviews, and full usability tests should be performed. Participants from the user population should be included when applying these methodologies. (U.S. Department Of Health and Human Services, Food and Drug Administration, 2002)

Requirements and guidelines from the FDA have an impact all over the world since many vendors target the U.S. market. Thus, it is likely that development projects in other countries need to address issues raised by the FDA.

Adding the body of international standards such as ISO/IS 9241-10 (1996), ISO/IS 9241-11 (1998) and ISO/IS 13407 (1999) to this picture, it becomes evident that there are incitements for the industry as well as the public sector to take usability and UCSD seriously. It is to a great extent enforced by the law and regulated by international standards and guidelines.

Even though there is no a law against designing websites and systems with poor usability, it is the responsibility of the development organisation as well as of the business organisation to make them as usable as possible, given the circumstances.

> *Poor usability is a potential element in lawsuits and other litigation. The US government's recent case against Microsoft hinged on a usability question: Are users well-served when the browser and operating system are closely integrated?*
>
> Donahue, 2001

Ethics and Moral

The fact that interactive technology and IT systems have an effect on the quality of people's lives and the quality of their work – i.e. the systems should be usable and not cause physical or mental damage – requires that ethical considerations are taken into account in the development process. Developers have power over other people and must exercise that power in an ethical fashion. Ethics are also concerned when we know that we can do better, but neglect to do so because it takes extra effort. Knowing, for instance, that you should involve users actively in the design process is something that many stakeholders in systems development do not bother find out. Such knowledge can easily be acquired by reading parts of any of the numerous books on the subject (one example is Beyer & Holtzblatt, 1998) or by hiring UCSD expertise.

People taking part in development, organisational changes, defining work procedures, etc. have a duty to listen to and act upon users' real needs and their explicit and implicit requirements. This is particularly important in situations where complex technology makes it hard for anyone but the engineers/computer experts to foresee the effects it will have on people (see for example Bynum & Rogerson, 2003 for discussions on this). We cannot ignore users and pretend that they do not exist. By taking on this responsibility, we will gain better acceptance for the system, or service, and create business value.

Another major concern is safety critical systems. Developing, for example, a control system for a nuclear power plant does not only involve complying with regulations and specifications, it also means taking precaution to avoid possible hazards.

In Sweden, it is generally assumed that all citizens should have equal opportunities to take part in the public debate and take advantage of the services provided by the public sector, i.e. the national authorities. This is best illustrated by the governmental initiative to make authorities available 24 hours a day and seven days a week on the Internet, "The 24/7 Agency" (Swedish Agency for Public Management, 2000). This includes designing electronic services that are accessible to all citizens, regardless of disabilities, cultural background and age. There are general guidelines on how to achieve accessibility provided by the Swedish Agency for Public Management (2002).

Another dimension is the long tradition in Sweden of unions acting as stakeholders in everything that concerns workplaces and work practices. Basic values for social awareness and democracy, for instance, equal opportunities regardless of abilities, are of course important aspects in systems development.

In Summary

All these reasons provide a palette of arguments for introducing and applying user-centred systems development. They range from strictly economic considerations to ethical and social issues. Different reasons appeal to different stakeholders: economic reasons are strong arguments when talking to managers and business people; suitability for business, quality in use and the validity of the system are concerns for the business and the user organisation; quality in terms of process efficiency is an objective for the development organisation; etc. Legislation and standards have an undeserved reputation of being difficult to comply with. But, they can have an impact if they are considered up-front and used with care. Ethic aspects are often treated with indulgence or seen as a "luxury". However, the real challenge is to take on the whole picture.

This introduction provides a background to my research and highlights the unused potential that I believe lies in the concepts of usability and UCSD.

Through my research and practice I try to understand how we better can communicate that potential and encourage more organisations to adopt UCSD. I do not want to sort the reasons in any particular order. They all have their own merits. But, reflecting on how I promote UCSD, I can see that I put less emphasis on the economic arguments and more on the other reasons. The one reason that I constantly return to is: *having users benefit from a better work situation*, and the advantages that people as well as organisations will experience by that.

Purpose of the Thesis

My research is dedicated to the task of providing knowledge and experiences about how to design usable interactive systems in practice. In my experience, to build usable interactive systems you need to apply some kind of UCSD process and maintain a user-centred approach during the whole development cycle. To me, usability is about supporting people in what they are doing and in what they are trying to accomplish, and further, the way they go about doing that. This includes how people communicate and interact. My main focus is on interactive systems used by people as supporting tools in their daily work at their workplaces. However, I also have experiences from other contexts, such as e-commerce and mobile handheld computers.

I conduct my research and work in the wide context of systems development. Within systems development the production of software is essential and in many situations the systems development project is a software development project. Often development organisations talk about software development or software engineering on the same level as systems development. But, where software engineering focuses primarily on the production of software solutions for given applications, UCSD provides a wider scope in that it is about finding ways of supporting people, without in advance knowing exactly how.

This thesis is a compilation of journal articles, conference papers and book chapters that I have contributed to. I have tried to put them together into a format that can appeal to people, practitioners as well as researchers, who want to learn about and explore the areas of usability and UCSD. I have also chosen a pragmatic and kind of “lessons learned” approach to describing my research, placing less emphasis on in-depth theoretical discussions.

In the summary of this thesis I start by describing my research aims and briefly explaining the foundations of Human-Computer Interaction (HCI), usability and UCSD. In the later chapters I discuss the outcome of my research, present the papers and indicate some interesting future research areas.

Research Objectives

> *Action research combines theory and practice (and researchers and practitioners) through change and reflection in an immediate problematic situation within a mutually acceptable framework. Action research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning.*
>
> Avison, Lau, Myers and Nielsen "Action Research", in Communications of the ACM, January 1999, Vol. 42, No 1, p. 94.

A great number of users using interactive systems as tools in their work, from time to time encounter difficulties in using these tools. These difficulties can be related to a number of causes, many of them linked to poor usability of the systems. The problems have been described in numerous reports, books etc. over the last decade (extensively reported by: *usability experts* such as Jakob Nielsen on his web site http://www.useit.com/ and in his books "Usability Engineering" (1993) and lately "Designing Web Usability" (2000) and *interaction designers* such as Alan Cooper in his book "The inmates are running the asylum" (1999)). Even if there are no figures available describing the total effects of poor usability throughout the world, we can read about it in HCI literature and in papers, hear about it at conferences, listen to gurus talking contemptuously about it, hear users talk about it with frustration, experience it ourselves, etc. Today, we can even hear about it on the news. On top of that, there are all the systems that we do not hear about, used within companies and organisations where the users do not have a choice. In many situations people are forced to use certain systems as a part of their daily work, no matter how poor the usability is.

I would like to use three reasonably recent examples to illustrate how serious and crucial usability problems can be. Serious usability problems do not necessarily mean that people's lives are at stake, such as in poor designs of airplane cockpits. But, the effects of poor usability may be huge nevertheless, given systems that are used frequently and/or are critical for business. The examples are taken from domains that I am familiar with from my research and practice. Furthermore, they show that there are still problems in applying UCSD even though HCI, usability and UCSD are well known research topics since decades. It seems that much of the knowledge gained within the HCI research community has not been put to practice.

- In an article in a Swedish medical journal, physicians complained about the computerised medical records. One physician claimed that a disadvantage with the computerised medical record is that it does not provide a proper overview of the information. He was forced to use different parts of the system to get all the information needed to make a decision. Even worse was the fact that the poor overview made him less confident in making that decision. He estimated that the use of the computerised medical record prolonged his working day by one hour (Dagens Medicin, 04/01). These kinds of usability problems are by no means novel. Our department at Uppsala University has conducted research on the topic of making computerised medical records usable for a long time (see for instance: Allard, Lind, Sandblad and Schneider, 1984). It seems, however, that this research has not reached the practitioners developing systems like the one described above.

- A case-handling system introduced at a large Swedish authority made it more difficult for the users to do their jobs. The new system has prolonged the time for handling cases and this has led to delayed services to the public. The major cause for the delays is that the staff has had to adjust to the new system and find new work practices. Moreover, the development project consumes huge resources and this will eventually lead to staff cuts. (Sources: Dagens Nyheter, November 21, 2002 and Uppsala Nya Tidning, November, 21, 2002. Both are Swedish newspapers)
- An extensive survey among 1.200 Swedish companies (conducted by five major unions within the Swedish Trade Union Confederation (LO) and UsersAward (2002)) reveals that:
  - Less than 50 % of the users report that the IT systems make their work easier.
  - Only two out of ten claim that they have had any influence on the development of the IT systems.
  - Less than 50 % of the users report that the IT-systems provide a proper overview of their work tasks.
  - Only four out of ten report that internal and external communication has improved.

I have asked myself the question—how can I as a researcher and practitioner improve the situation and promote the development of usable systems? My approach is to do research and to work in such a way that I am present in the situations where systems are defined, designed and built. In that position, I try to influence and guide organisations and the individual projects to adopt a user-centred approach. I can do this by: introducing methods, processes and practices; observe and act; analyse and generalise, and moreover, have a *dialogue* about usability and UCSD. Further on, I can reflect *in*, as well as *on* practice, analyse and communicate my results to others for further use, and hopefully make some change in systems development that is beneficial to the users.

One may ask if research on UCSD is not somewhat outdated since the concept has been around for quite some time now. I would be happy if the current situation confirmed that further research on UCSD is superfluous, but that is unfortunately not the case. I firmly believe that further research is necessary. Recently, this was also articulated by two influential UCSD advocates, Karat & Karat (2003): "[…] the field [UCSD] is becoming clearer about the methods, but there is still a great deal of work to do before practice could be considered as fully mature." (ibid, p. 538) and further: "We do believe that we have come a long way from the old days in understanding what human-factors practitioners can (and should) do to contribute to creating usable systems, but it is currently more in the heads of experienced practitioners than in the HCI literature." (ibid, pp. 538–539) It is obvious that we still do not have all the answers on how to develop usable interactive systems in practice, despite the long record of the research in the field. The question is what the track record looks like.

In my research, practical applicability is very important. I would therefore like to clarify my view on the definition of practice: "Practice. A technical or management activity that contributes to the creation of the output (work products) of a process or enhances the capability of a process." (ISO/TR 18529:2000(E)). Practice is obviously connected to some sort of activity conducted, preferably by a skilled practitioner[3]. This relationship is important since I believe that the role of the practitioner is essential for the outcome as well as the performance of a process.

Research Scope and My Aims

My research focus is foremost on usability and user-centred processes for designing interactive systems to be used by users at workplaces. Is UCSD the only way to achieve usable interactive systems? Most certainly not, but I believe that there are enough reasons to continue to explore the potential in UCSD and I think that every single reason mentioned in the previous chapter is in itself enough to continue to do research on how interactive systems can be made more usable.

The systems development process is for many reasons central in my research. The foremost is that *the systems development process is essential for the outcome and the quality of a system*. Moreover, *the large body of knowledge in HCI, usability and UCSD is rarely applied in the systems development process.* The overall aim of my research is therefore to provide the industry and organisations with better instruments for practicing UCSD. These instruments

[3] Practitioner is: "One person who practices an art or profession", Webster's student dictionary, 1997.

include knowledge, insights, arguments, methods, and processes. I ground my research on the below assumptions and simple hypotheses:

- Making people, project members and organisations aware of usability and UCSD will increase the chances of improved usability in interactive systems.
- Introducing methods and processes, based on a set of principles for UCSD that are comprehensible, familiar and make sense to a development organisation, will increase the possibility that the methods and processes will be used.
- Making usability and UCSD explicit and even appealing, i.e. through a set of key principles, a process and a development role, will make it more likely that these concepts receive proper attention and even become demanded.

The above is more like a vision than a research objective. In order to be useful as a research objective, it has to be described in more detail. Therefore, what I do is to focus on the problems and areas that I have found to be the most urgent ones, based on my and my research colleagues' experiences from several organisations and projects.

A central problem is how to integrate usability considerations and a user-centred perspective into organisations and existing systems development processes in an efficient way. This problem can be broken down into a set of interesting research topics and areas that I use as guidance in my research:

- How can we raise the awareness in companies and organisations about the importance of usability and UCSD?
- How can we make UCSD the "standard operating procedure" in organisations?
- Can we develop "easy-to-apply" techniques: methods, activities, processes, project roles, etc. and make them attractive for the development organisations? Many organisations have problems with applying existing methods and processes. Although people in the organisations have heard about usability, they do not know how to achieve it. Even though there are numerous usability methods available, the methods have to be used regularly.
- How can we increase the "lowest level" of usability awareness among developers? On the whole, many developers need to learn more about usability and UCSD. This means that not only the specialists and experts should have knowledge and skills in UCSD and usability, but also the other participants in a development project – e.g. project managers and software engineers.

These topics are rather "big" issues, somewhat imprecise and perhaps not sufficiently detailed. However, they illustrate the wide scope of my research. I deliberately concentrate on the breadth since I believe that it is a key to the question: *how to develop usable interactive systems in practice?*

An alternative approach would have been to conduct narrow and controlled studies of well-defined parts of the development process, or to develop a new method for measuring the usability of a system. I have chosen not to do so, since I consider it essential to study and understand *the process* of developing systems. This means that not only the small parts of a process are of interest; the combination of the parts, as well as the dynamics and the context in which a process is used, are very important for the outcome.

Research Context and Limitations of My Research

I work part time as a developer/consultant at the consultancy company Enea Redina AB[4] and part time as a researcher at Uppsala University. This is important background to my research. I have chosen to combine these two roles since I am confident that research and practice benefit from each other. Furthermore, I believe that systematic research combined with experiences from practice offer me a good and solid ground for making my research contributions applicable in real life settings.

I have been involved in projects, conducting research and working as a practitioner, in a number of organisations and companies over the last decade. Here are some examples: The Swedish Tax Agency (SKV, formerly known as RSV), The Swedish National Social Insurance Board (RFV), The Swedish Medical Products Agency (Läkemedelsverket), The Swedish National Road Administration (Vägverket), PharmaPoint, AffärsData, Dagens Nyheter, Amersham Biosciences, Meditelligence, Telia and The German Cancer Research Center (DKFZ). Working with this mix of organisations has been of great value for my research. They represent organisations with quite different characteristics. They develop systems in different contexts and within different domains. Some of them have in-house development departments, others are product companies and some of them engage in contract development.

Primarily, I have studied the development of administrative systems used by professionals at their workplace. These systems are typically bespoke systems and not "standard" office applications like, e.g. Microsoft Office. They are often used in complex, sometimes technical tasks, for instance, a workflow

[4] Information about the company can be found at: http://www.redina.se.

system within the biosciences field. I was involved in the design of the workflow system, which can be considered an administrative system, but the system was used to administrate a complicated flow of events in different biochemical processes. I also have experiences from other types of systems, such as e-commerce, handheld computers and teleradiology. Despite the breadth of my experiences, I have not been able to explore a number of highly interesting domains. One such domain is public websites intended for public information services or advertising. Further, I have not been involved in projects specifically targeting children, interactive education or computer games. Neither have I taken part in any really safety critical systems development projects, i.e. for nuclear power plants.

Research Approach and Method

Action research makes it possible for the researcher to apply his/her theories in practice in a realistic work situation, and to take action and make a change in that situation. The *action* brings about change in some community or organisation or program, and the *research* increases understanding on the part of the researcher or the client, or both (and often some wider community) (Dick, 1993). The mix of action and research can be tuned to the level that is accurate for the researcher's aims. One important aspect is that the researcher takes part in the studied situation, for instance a project, not just as an observer but also as a participating project member.

Action research differs from experimental research with fixed parameters in a controlled laboratory setting. Instead, action research projects are conducted in real life settings. Action research has its origins in socio-psychological studies of social and work life issues. It is based on the idea that the researcher can better understand the social system if she is a part of it rather than a detached observer. Our action research approach involves studies of the systems development process in different organisations, suggesting certain changes and activities to achieve those changes, participating in the activities and observing the outcome.

In the article *The History of Action Research*, Masters (1995) describes the fundamentals of this research school. She summarises the history and background of action research into four basics themes: empowerment of participants; collaboration through participation; acquisition of knowledge; and social change. I try to apply the essence of these themes in my research. The empowerment of the developers and the users in development projects is important, as well as full participation on equal terms for everyone involved in our project teams. This is not an easy task since resources; time, etc. are typically very limited. To fully understand and take on a user-centred systems design approach involves a major attitudinal and social change in a development organisation. Such a change does not happen quickly or easily. On the contrary it is an evolutionary process that takes years. Knowledge acquisition is part of the learning process that takes place within every organisation (or at least in most of them). For every project, knowledge is added to the organisation's knowledge base. It is the researcher's responsibility to reflect upon, analyse and generalise the results from projects and to disseminate the knowledge thus gained and communicate it to others for further use.

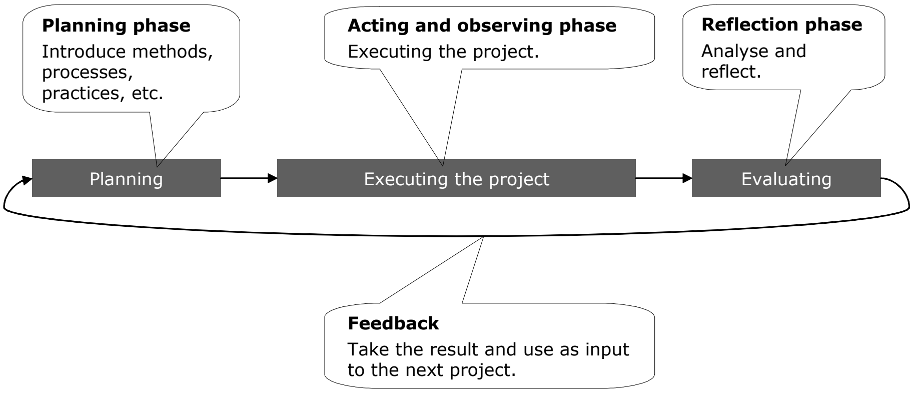


Figure 2: A cyclic process of: planning, acting, observing and reflecting (illustration by the author, inspired by Masters (1995)).

The process that the researcher goes through in action research is a spiral of cycles consisting of four major phases: planning, acting, observing and reflecting (Masters, 1995). Figure 2 attempts to place this cycle in a project context. The researcher goes through all the steps in the cycle, whereas the developers usually come in at project start, work in the development process and then move on when the project is finished, or even earlier. The researcher starts the research in advance by planning the research case, then participates in the project, observes, collects data and continues after the project to conduct systematic analyses and reflect on the project.

Practitioner-Centred Research

Lately, I have been increasingly concerned with the question: How can we increase the impact of research in professional practice? In parts, the issue can be resolved by means of action research as discussed above, but I would like to go further. Practitioner-Centred Research (PCR), related to action research, has inspired both my research and my practice. PCR has its background in the insight of the limited contribution of research to professional practice (Bourner & O'Hara, 1999). PCR comes from areas such as higher education, practice of law and medical practice. I can see that there are similarities in the way that I practice UCSD in organisations and the way a lecturer teaches a class. We are both concerned with communicating knowledge in such a way that the student/apprentice can put it into practice.

The interesting thing about PCR is the mixture of professional practice and research. Bourner and O'Hara discuss this topic in terms of PCR being located within the domain of the researching professional rather than the professional researcher. It is one thing to share new knowledge, but a completely different thing to share new practice. The sort of new knowledge that practitioners need to adopt a new practice is both objective, answering the question *"can it work?"* and subjective, answering the question *"can it work for me?"*

Being a practitioner, trying to help other practitioners, makes me particularly concerned about my research being applicable to systems development in practice – sorting out what works and what does not work. Action research and PCR are two ways of linking research and practice. In PCR the "researching practitioner" aims: "[…] to create improvements in professional practice by adding to the stock of usable knowledge available to practitioners." (Bourner & O'Hara, 1999). Through PCR I, as a practitioner, can test and refine my ideas through application in my own practice. The results coming out of such projects is not only an interactive system, but also knowledge and experiences about the process itself and the practice of it.

The Reflective Practitioner

Reflection is an essential ingredient in both PCR and action research. This reflection consists of a dialogue between: a number of researcher-participants; or two researcher-participants; or even in its most extreme form the reflections of a single researcher-participant. Action research includes researchers collaborating with practitioners, engaged in a systematic attempt to understand and change practice by means of their own actions and reflection upon those actions—*reflective practice* (Webb, 1995).

To *reflect-in-action* and *reflect-on-action* are important for a practitioner. The *reflective practitioner* was introduced by Donald Schön (1983, 1987) and describes a way for professional practitioners to evaluate their own work, to learn lessons and improve their practice. Reflection-*in*-action helps us as we complete a task. It is a process that allows us to reshape what we are working on, while we are working on it. It is that on-going experimentation that helps us find a viable solution. Reflecting-in-action typically takes place when a surprise, or something unexpected, appears in the process of accomplishing the task, and that surprise causes one to question how and why the surprise occurred.

In reflection-*on*-action we evaluate our own process: "We reflect on action, thinking back on what we have done in order to discover how our knowing-in-action may have contributed to an unexpected outcome" (Schön, 1983, p. 26). The results from reflection-on-action often make up parts of our documentation such as "lessons learned" sections in conference papers or in journal articles. Whilst the reflective practitioner does not necessarily share her knowledge with others, as reflecting can be intensely personal, the action research and PCR approaches encourage practitioners to share lessons learnt and new knowledge publicly.

Action Research and PCR – Applied with Reflection

Action research and PCR involve qualitative analysis and interpretation. Within our projects we use various methods to collect data, for instance, questionnaires, field studies, observations, interviews and video recordings. As practitioners we also use UCSD methods, e.g. user analysis, task analysis, contextual inquiries, prototyping, scenario-based design, and various methods for usability evaluations. These are primarily practical methods for accomplishing something in a project, but they can also be used to collect data for the analysis and evaluation of a project. Furthermore, we also use, directly or indirectly, software engineering methods, such as formal methods for modelling system behaviour.

To me, the most important outcomes of my action research and PCR projects are positive changes in a development organisation, improved work practices and processes, and often an interactive system. In order to make it possible for others to make use of our work and our results, we write conference papers, books, journal articles, etc. Those “real-life reports” typically include a case description, reflections, findings, etc. They are usually descriptive and aimed to illustrate: why we did something; what we did; how we did it; and the lessons learnt.

I keep a diary in every project. This is an easy-to-apply method for gathering data. It is a great help when analysing and reflecting on what we accomplished in the project. Figure 3 shows the tools I use most frequently as a researcher and practitioner. I use my laptop for the diary as well as for all kinds of documentation, and of course for prototyping. My typical tools for prototyping include: Microsoft PowerPoint, Visio and Visual Basic; Jasc Paint Shop Pro and Adobe PhotoShop; HTML and XEmacs. I always have paper and pencil at hand to capture ad-hoc information and to take notes “on-the-fly”. A digital camera is a great tool for capturing situations and contexts. You can also use it for taking snapshots of whiteboards during meetings and workshops.

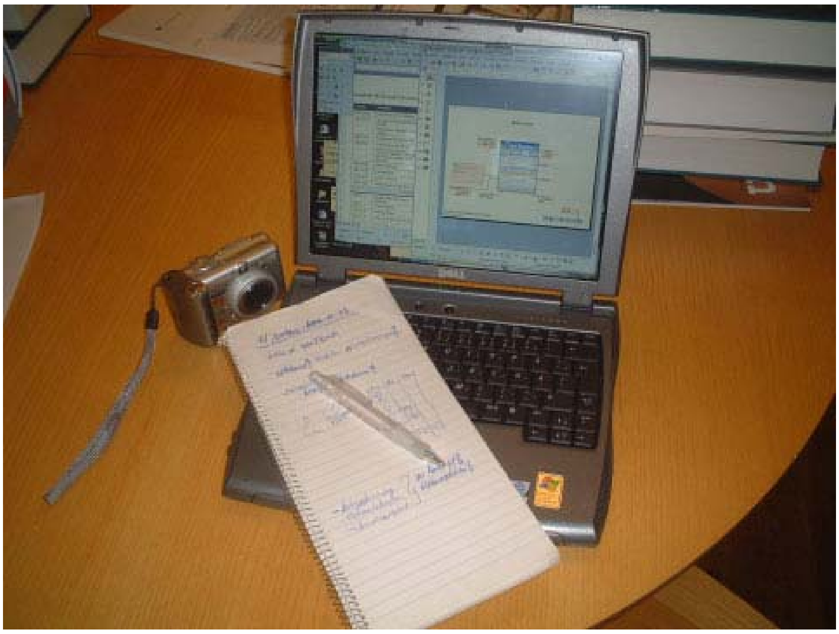


Figure 3: Typical tools for capturing information used by me in projects: a laptop computer, a digital camera and paper & pencil.

I try to conduct action research in parallel with PCR. In my role as researcher I am typically part of an action research team. In my role as consultant and practitioner (frequently as a mentor) I can take on the PCR approach. The two approaches show many similarities, but where the action researcher is someone coming from the outside, not taking full part, the practitioner is expected to contribute to and make a difference in the project. As a researching practitioner I believe that I can get a better insight since the practitioner is more "involved" than the professional researcher. I argue that combining the roles of the researcher and the practitioner in action research and PCR will provide me with the best conditions for making research contributions, as well as contributing to improved practice. Being a practitioner as well as an action researcher I can make reflections while taking action in projects. Then, later on, I can reflect on the actions, analyse, make abstractions, try to generalise and communicate the lessons learnt. Moreover, being a researcher involves being a practitioner; in the area of research.

In my practice I try to maintain my role as a researching professional. I always look for research opportunities in the projects where I am, or will be involved. Most of the projects are very interesting and challenging, such as the ones illustrated in the design cases (papers 7, 8 and 9), but sometimes they are more routine tasks. But, even when performing "routine" usability evaluations I try to be critical and reflective, both during the project and afterwards. Combining the roles of the practitioner and the researcher is of course not unproblematic. There are situations in projects where I am faced with questions like: is this really in the best interest of the users? Or, if I do not act now it will be a very nice research case, but the result might be disastrous for the users. The conflicts between research, business, the client and the users are sometimes evident. But, at the same time, this is a real life situation where compromises always have to be made. I have never been involved in a project where my own practice and the business interests of my company have put up obstacles to working with users. On the contrary, my company and I are always vouching for active user involvement. My experience is that the real tension is usually within the client organisation; between business management and the development organisation or even within the development organisation.

In our development or modification of systems development methods and processes, we try to adhere to certain criteria for making sure that they are applicable and of high quality. Goldkuhl (1994) argues that one way of establishing methods are to "ground" them. He proposes three criteria for proper grounding of methods:

- *Internally grounded.* The method should be internally congruent; the different parts of the method should be consistent and fit well together.
- *Empirically grounded.* The method should be empirically tested and evaluated. Empirical data should be collected and the method should be evaluated in relation to these results. This can be done using different approaches, such as action research, experiments in laboratories or post-studies of the use of the method.
- *Theoretically grounded.* The method should be evaluated against and related to existing theory.

To "ground" a method is an iterative process that takes time. A method must be practiced and refined before it can be proven to work. Moreover, when a method has been established, it must be applied with care and adapted to the existing context and conditions. My experience is that a method must by adjustable to be useful in practice.

Reliability and Validity of My Research

Reliability and validity are important aspects of research in order for the reader to assess the results. Can the reader repeat the same "experiment" and get the same results? Can the reader be sure that the measures used measure the "right thing" and that there are no systematic errors? In action research, the researcher tries to introduce something new and presumably valuable to a project and then observe the outcome to see if it had any positive effects. We cannot possibly run controlled experimental studies on systems development projects. The settings for systems development projects differ from project to project, including the goals or directive, context, constraints, budget, time frame, people involved, attitudes, etc. All of these factors will affect the outcome. Such factors are never identical from one project to another, nor can they be controlled. It would be impossible to keep certain factors constant and vary others, e.g. running two development projects that are identical apart from the development process (Lind, 2000). Neither is it possible to run two real life development projects in parallel, using one as a "control" project and varying some factor in the other. What you can do is to study many projects, look for patterns, analyse and reflect on why some activities seem to be more successful than others.

There are, however, a number of potential problems in action research. For instance, since the researcher suggests and introduces changes, she may find it difficult to conduct objective evaluations of the results. There is also a risk that the researcher may bias the results during data collection (e.g. observations and interviews). Moreover, the outcome may be biased if the participants improve their performance simply because they are observed. There are ways of avoiding some of the problems. By iterating the action research cycle within projects, and by conducting similar projects in similar settings, we can gain better and better understanding of what can be seen as functioning patterns or new knowledge. This is one way to increase the validity of the outcome. The usability design process (and its variants, paper 2 and paper 3), the usability designer role (paper 2, paper 3, paper 4 and paper 5), the key principles for UCSD (paper 1) have been developed and refined in several projects, iteratively. The design case papers (paper 7, paper 8 and paper 9) provide some examples from my practice where the usability design process, the key principles and the usability designer role have, in some way or another, been introduced, used and refined. Furthermore, by inviting other practitioners to use the principles, the processes and the role, we can to some extent triangulate the findings or observations increasing the confidence in our results. Moreover, most of my research has been put into practice in many situations and different contexts. We use the results and lessons learnt in our teaching of various courses on HCI and UCSD at Uppsala University. We have presented our research at numerous national and international seminaries and conferences, and given tutorials at international conferences: NordiCHI 2002, HCI International 2003, Interact 2003, CHI 2004, and a workshop at Interact 2001. Our Swedish textbook on UCSD (Gulliksen & Göransson, 2002) is also a result of our research. The key principles, the usability design process and the usability designer role have been the main topic in workshops, both within the research community and in the industry. All these different forums have offered, and still offer, opportunities to discuss, analyse, reflect on and develop our research findings and experiences.

As a practitioner I am typically involved in several parallel projects. This provides opportunities to try out research ideas and activities in several projects, which is an advantage. Where the researcher may have one project to study. I can take the results from one project, reflect on them, analyse them and apply them in the next project. The challenge is to get enough time to reflect on the actions and analyse the results. However, this is fundamental for me as researcher and practitioner.

Human-Computer Interaction

> *Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.*
>
> ACM Special Interest Group on Computer-Human Interaction (SIGCHI) Curriculum Development Group, 1992, section 2.1.

I would like to use the above quotation as a starting point for a discussion on the area or subject of HCI. This definition is, for instance, used and discussed in the two predominant course books in HCI classes in Sweden: Dix et al. *Human-Computer Interaction* (Dix, Finlay, Abowd & Beale, 1998) and Preece et al. *Human-Computer Interaction* (Preece, Rogers, Sharp, Benyon, Holland & Carey, 1994). The expression human-computer interaction was adopted in the mid 1980s, when HCI was recognised as a research area of its own. HCI had by then evolved from the more limited scopes of user interface and man-machine interface, commonly used in the 1970s. By introducing the HCI concept, the area was accepted as being broader than just the design of the interface. It was recognised to be concerned with all those aspects that relate to the interaction between users and computers. Delving deeper into the origins of HCI, we find that the areas of ergonomics and human factors can be considered to be the “mothers” of HCI. Ergonomics and human factors are more or less synonyms, where ergonomics is primarily used in Europe and human factors in North America (Dix et al., 1998, p. 2). Both areas concern studies of machines, systems, humans and the effects on the working environment and human performance. The Second World War boosted the interest in these fields as the war combatants worked on developing effective weapon systems. As computers became more used and widespread, researchers started to specialise in the interaction between people and computers, leading up to the current concept of HCI.

There is no single interpretation or understanding of the subject of HCI. But the definition provided by the ACM SIGCHI Curriculum Development Group (cited in the ingress of this section) is perhaps the most widely used definition. The diverse nature of HCI is elaborated as the curriculum group discusses the content of HCI:

> *Human-computer interaction is concerned with the joint performance of tasks by humans and machines; the structure of communication between human and machine; human capabilities to use machines (including the learnability of interfaces); algorithms and programming of the interface it-*

> *self; engineering concerns that arise in designing and building interfaces; the process of specification, design, and implementation of interfaces; and design trade-offs. Human-computer interaction thus has science, engineering, and design aspects.*

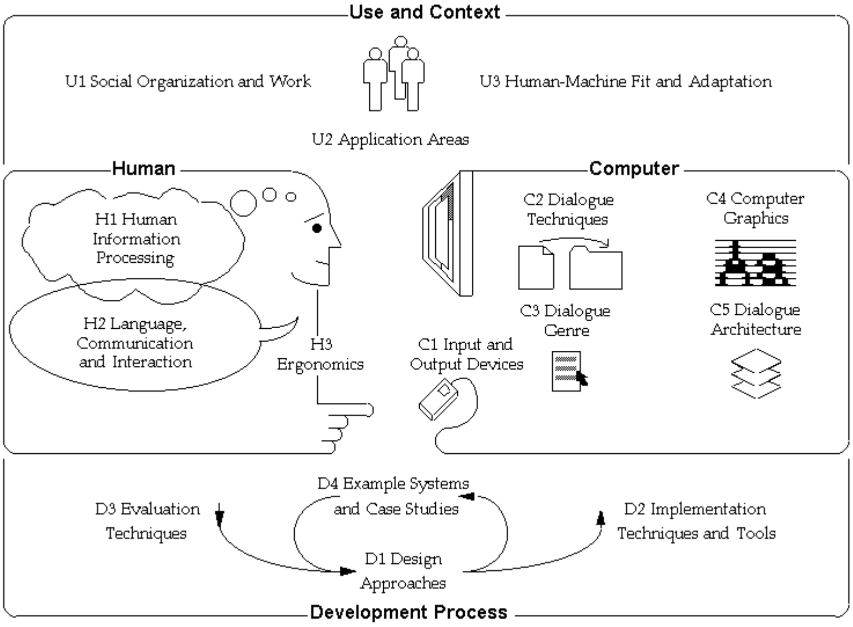

Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
Computer
C2 Dialogue Techniques
C4 Computer Graphics
C3 Dialogue Genre
C5 Dialogue Architecture
C1 Input and Output Devices
D4 Example Systems and Case Studies
D3 Evaluation Techniques
D2 Implementation Techniques and Tools
D1 Design Approaches
Development Process

Figure 4: The content of Human-Computer Interaction according to ACM SIGCHI Curriculum Development Group, 1992.

There are four major areas within HCI according to the ACM SIGCHI curriculum (Figure 4): Use and Context (U), Human characteristics (H), Computer (C) and Development Process (D). To successfully develop an interactive system we must learn to know and understand the intended users of the system: their background, capabilities and constraints, and also their work situation and work environment. Much can be learned by understanding human behaviour in general, but we must also learn about the specific users and their environment and tasks, etc. Furthermore, we must identify and describe the technical possibilities and limitations, development tools, etc., and we must have a process or a framework to guide our development. This sounds quite simple, but turns out to be a major challenge.

My research falls mainly into area D, the development process. I have a computer science background, and have, roughly speaking, moved from area C into area D. My research and practice has over the last decade focused on the UCSD process.

The ACM further discusses the disciplines involved in HCI as follows:

> *Because human-computer interaction studies a human and a machine in communication, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics,*

> *operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human performance are relevant. And, of course, engineering and design methods are relevant.*
>
> ACM Special Interest Group on Computer-Human Interaction (SIGCHI) Curriculum Development Group, 1992, section 2.1.

Evidently HCI involves many disciplines. One can, however, discuss how the different disciplines have influenced HCI. The ACM SIGCHI Curriculum Development Group and Dix et al. discuss HCI from a computer science perspective. They argue for HCI being taught in computer science classes, programmes, etc. From a computer science and systems design perspective, HCI can be seen as a subject that "[…] involves the design, implementation and evaluation of interactive systems in the context of the user's task and work." (Dix et al., 1998) Other researchers and authors promote a behavioural science approach, e.g. Preece et al. I believe that both computer science and behavioural sciences have had major impact on the HCI community.

HCI comprises a multitude of theories, coming from different disciplines. I am personally influenced by theories from the behaviour sciences (such as cognitive psychology, perception and mental models), computer science, software engineering and, of late, interaction design. I do not relate my research, or practice, to any one theory in particular, but to the diversity of theories in the disciplines of HCI. I have found the four approaches to design presented by Wallace & Andersson (1993) useful when discussing and elaborating on different theories in relation to HCI and systems design:

- The *craft* approach. A structured approach to interface design is impossible, as the aesthetics of the interface cannot be achieved by means of analytic techniques. It focuses on the designer's need for talent, not method. This approach draws heavily upon design theories described in Fallman (2003) as relying on "creative geniuses". Fällman labels these theories "the romantic account".
- The *enhanced software engineering* approach attempts to introduce HCI techniques into the repertoire of traditional software engineering. It suggests for example various methods for task analysis to make HCI aspects a concern of the software engineers. This approach is linked to theories in software engineering and systems analysis.
- The *cognitive engineering* approach aims at applying theories from cognitive psychology to the problems facing the designer. Cognitive metrics models, such as the keystroke-level model (Card, Moran, & Newell, 1983) measures the user's performance and thereby indirectly estimates the memory load for unit tasks to help predict the efficiency of different design solutions.

- The *technologist* approach tries to solve the problems of interface design by providing appropriate programming tools and development environments. This approach uses theories primarily from computer science.

I do not want to commit myself to a certain theory or approach. I firmly believe that HCI must be made tangible and "easy to apply" in order for it to be relevant and useful for practitioners, and systems development projects. Focusing on theories seldom generates work practices that are applicable in practice (Long & Dowell, 1989). Developing for example a process or a method that works in practice, may require an intersection or a mix between different theories.

HCI has a "scientific" bias. The creative disciplines and craft approaches, e.g. graphic design and industrial design, have not been fully recognised within HCI for the importance of the contributions they make to the design of usable interactive systems. Design approaches within HCI are traditionally closer to computer science and engineering than to traditional design areas, such as, architecture. The character of design and the role of the designer are underestimated. This has been discussed by, for instance, Winograd et al. (1996) in *Bringing Design to Software*. Additionally, I find the below quote by Terry Winograd useful in communicating the diverse nature of theories within HCI:

> *Since the earliest days of the field there have been attempts to apply theories of cognitive psychology to HCI design, with small successes and overall mixed results. Other theoretical frameworks dealing with human thought and action have been proposed as the basis for design, such as activity theory, speech act theory, and phenomenology. Each of these provides broad orientation to human-computer interaction, and in some cases they have been employed directly in the design of the interactions.*
>
> Terry Winograd in an invitation to a Bay CHI meeting January 14, 2003, accessed February 12, 2004:
> http://www.baychi.org/calendar/files/flyer200301/flyer200301.pdf

Further, HCI has traditionally been concerned with studies of how humans interact with computers or machines, and how this interaction can be improved by means of certain methods and techniques. These methods and techniques were typically the results of academic research in controlled laboratory experiments, and focused on behaviour aspects. Although such research is adequate, the HCI research also needs to focus on methods, processes and best practices that can be directly applied in real life situations and in commercial industrial development projects. This has been highlighted by others in a similar manner; see for example Bannon (1991):

> *Despite some advances in the area some serious criticism has been directed towards the field for its lack of relevance to the practitioners and the limitations of cognitive theory when applied to everyday design situations. A partial remedy for this is to go from controlled laboratory experiments to doing workplace studies.*

My point is that it is far too difficult to apply HCI theories and use methods exactly as they are described in practice. There are typically circumstances in a specific project that require adjustments and modification, and sometimes you also need to improvise a bit. There is rarely any room for "full scientific quality" when working in industrial projects. Is it significant that HCI and usability researchers have their own forum such as the annual ACM/SIGCHI conference on Human Factors in Computing Systems, whilst usability professionals are organised in the Usability Professionals' Association (UPA) organising their own conference?

Others criticising HCI try to establish research fields "beyond" HCI such as *interaction design* (more on this topic in the chapter *The Legacy of UCSD and Some Trends*).

However, to me, the HCI research field encompasses the very foundation of knowledge required in the development of usable interactive systems.

The Legacy of UCSD and Some Trends

> *But user-centred design emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system.*
>
> Donald Norman "Cognitive Engineering". In: Norman D. A. & Draper S. W. (Eds.), User Centred System Design, 1986, p. 61.

Is it important that an interactive system is usable? Most people would reply to that question with the rhetorical question: are not all systems supposed to be usable? The usability of a system is generally implicit and taken for granted. For example, when a customer orders a system, he or she assumes that the developing organisation will produce a usable system. This makes perfect sense from a customer perspective, but is unfortunately far from the truth. In many cases, the customer has to explicitly order a usable system by defining goals or requirements for the usability. This is neither easy nor something that customers are familiar with. The usability of an interactive system is utmost important in a context where professionals use interactive systems as tools in their daily work. The systems have to be effective, efficient and satisfactory to use in order to support the users in their work activities. The focus must be on the professionals accomplishing their work, and not on the technology itself. In a work situation, the professionals should be able to focus on their work-related activities and not be forced to struggle with a system that is not suitable for them. This is true for almost any user; struggling with a system instead of using it to perform the actual tasks is annoying.

In this chapter I discuss the concepts *users*, *usability* and *UCSD*. This is to provide a background before I finally discuss the outcome of my research and try to summarise lessons learnt.

What is in the Concept User?

We are all computer users these days, are we not? That does, however, not mean that we all need the same systems or that we have the same context of use. We need to recognise at least two things regarding the term *user*: developers are not the targeted users of the systems that they develop; and the customers who order a system are usually not the users. What I am referring to as users are those who will ultimately use the system to achieve some kind of task and goal. They are sometimes called "real users", end-users or simply users.

I use the terminology user and user-centred because they are known and, at least partly, acknowledged. Furthermore, I think that the term user implies that someone is actually using the system for a purpose. User, as in user-centred, is also to prefer compared to human or even people, as I believe that it takes more to design a usable interactive system than knowing a great deal about human behaviour in general. We also need to know about the *specific situations* in which humans are going to use the system—*the users and their context of use*.

In general I like to think that people want to accomplish something. It can be more or less trivial things, such as reading the latest news on the Internet, or more serious things such as paying your bills on the Internet or handling peoples' taxes as professionals. From my experience with professional users, I can tell that the majority of users are hard-working people who want to perform well and they benefit from systems that are supportive and *usable*.

I believe that humans are social and that our activities are social, situated and related to the context. The questions of how humans act and why we do certain things are subject to much research and theory building. From cognitive psychology we have learned about how individuals process information, often studied in an isolated situation, separated from their natural context. This is important knowledge, but it has been criticised and contrasted by other theories such as situated action (Suchman, 1987). Where cognitive sciences tell us that humans act according to a rather detailed well-defined action plan to achieve goals, situated action says that plans do not control actions, and that every course of action depends on the situation.

My view is that a user, or a group of users, has some reason, intention, goal or equivalent that triggers a set of actions. Exactly how they go about achieving this goal and what actions they take depends on many factors, where cognition, their earlier experiences, the context, the current situation, etc. are important. We do not act randomly, but exactly how we act may vary, and our actions evolve over time. You do not use a system in the same way when you are very familiar with it, as you did when you were introduced to it.

Interaction and communication between people are fundamental and must be considered in systems development. They cannot be neglected since interactive systems are not used in isolation. Furthermore, knowledge about human behaviour in general is important, primarily to provide rationales when discussing about and with users. We need to study and engage with users as they act in different situations. Then we need to use the knowledge we have gained as the basis for our system design. Grouping or categorising users is common and more or less inevitable in systems development since you cannot design for every single individual. Concepts and techniques, such as user profiling and personas, attempt to produce rich descriptions of the users, given the project context and limitations. Those techniques must not be interpreted or used in ways that turn individuals and groups into passive objects in the development process. That is, they should not be used as excuses to avoid active user involvement. On the contrary, they must be used in such a way that they help the development team actively engage users in the process. They are not substitutes for active user involvement, only enhancements.

Usability Defined

To make it possible to talk explicitly about how to accomplish usable systems we need to describe and understand what usability is. I prefer to use the term usability as defined in the ISO/IS 9241 standards on "Ergonomic requirements for office work with visual display terminals (VDTs), Part 11 – Guidance on usability":

> *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.*

ISO/IS 9241-11 Guidance on usability

The different parts of the definition are additionally defined as: *Effectiveness*: Accuracy and completeness with which users achieve specified goals. *Efficiency*: Resources expended in relation to the accuracy and completeness with which users achieve goals. *Satisfaction*: Freedom from discomfort, and positive attitudes towards the use of the product. *Context of use*: Users, task, equipment (hardware, software and materials), and the physical and social environment in which the product is used.

I, as well as others, have found the ISO/IS 9241-11 definition of usability valuable since it provides a common understanding of the concept. Furthermore, the definition emphasises that usability is objective and measurable, at least to some extent. If we want to, we can measure usability and say that for a certain user in a certain context, system X is more usable than system Y or the usability for system Z has improved by 50 %. The ISO definition covers a wider range of the aspects that are important to the users than most definitions and perceptions of usability do. I often hear people discuss usability in vague terms, such as "user friendly", "easy to use", or as something that is related to the user interface and graphics of an interactive system only. Even though the user interface is important, it is no more than a part of the system. We must make real efforts to learn about the potential users, their intentions/goals, their tasks, context of use, etc., in order to develop a usable system. This means taking into consideration all the aspects covered by the ISO definition.

However, I am somewhat ambivalent about usability being fully objective and measurable. Usability is neither completely objective nor completely subjective. Objective usability is closely connected to HCI and experimental psychology, focusing on for instance key-strokes and actions. Subjective usability points to peoples' feelings and subjective understanding and judgements. I firmly believe that the way to achieve usable interactive systems is neither just to specify and measure, nor just to create something elegant, based on the designer's own preferences. Instead, it is a process in which all aspects of the system evolve. The definition of usability specifies *what* we as developers need to know in order to develop usable systems, and how we can measure our progress. It does not concern *how to* develop usable systems—the actual process. To my opinion, this process must be user-centred and have the explicit purpose of supporting the development of usable systems.

It takes an effort to define usability goals, to use them as acceptance criteria and as a driving force for design solutions. For many projects and organisations this effort is too much or simply not applicable. This may be due to many reasons, including:

- Usability metrics are not easy to translate into design solutions. It is one thing to establish the goals and measure them, another thing to design accordingly, i.e. usability metrics are sometimes questioned as the most efficient way for producing "good" design (as opposed to an experienced designer doing the design).
- Not enough knowledge or experience in usability and metrics, or not possible for different reasons to define usability goals.
- Often there are not enough resources in a project dedicated to usability to make a thorough usability assessment.

Goals are essential, as argued in the ISO standard. But every project must identify and customise its own usability activities. If resources are restricted it may be wise to focus on what is most valuable for the users within that context. If there are "enough" resources available, go the whole way and establish a complete framework for usability. By defining usability goals we have the possibility to guide the design of the system and to ultimately measure the usability. The usability of the system can be described, documented and verified. Among other things, this gives us the opportunity to use the definition of usability as a tool for focusing on usability in projects. And further, to define methods, processes and practices to achieve usability. Perhaps it is not necessary, or possible, to have a complete set of usability goals, it may be enough to have a few goals only.

Recognising the difficulties in specifying usability goals, we have tested a concept that we call *design criteria*. Design criteria are discussed in several of my papers: paper 2, paper 3, paper 8 and paper 9. These criteria are complementary to usability goals. They are high-level criteria, partly derived from the usability goals, providing guidance for design decisions.

There are different views of the usability concept. Having a background in computer science and HCI, I have adopted the idea that usability can be defined and, at least to some extent, measured. Other concepts, such as *user experience*, have had an impact on the IT industry. The user experience concept implies that whenever the user is confronted with a product (such as an interactive system) the user shall experience a "positive" time together with that product. This includes all the way from seeing a software product on a shelf in the shop, through the installation of the product and the use of it. Partly in contrast to the standard definition of usability user experience focuses almost exclusively on the user's feelings and attitude towards the product. Furthermore, user experience focuses on the commercial aspects of usability—the customer is king. What the customer thinks and says rules, not what we can measure in terms of efficiency and effectiveness in use. I think it is a cause for worry that concepts like user experience take precedence over usability as defined by the international standard. I believe this is a step in the wrong direction, since it brings us back to the situation where usability is unclear and hidden in imprecise catch phrases. How can we measure concepts like user experience? How can we make sure that the users get usable interactive systems if managers base their decisions to buy systems on a fuzzy slogan like the user experience?

In other situations the term usability is referred to as the capability of a system or product to be used with ease. This corresponds to the definition of usability as a software quality. An example of this is the ISO/IEC 9126 (1991) (concerned with quality in software) definition of the term usability: "A set of attributes of software which bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users". However, the attributes required for usability depend on the nature of the user, task and environment. *A product has no intrinsic usability*, only a capability to be used in a particular context. Usability cannot be assessed by studying a product in isolation. I find this declaration, taken from the ISO/IS 9241-11 standard, important.

In Summary

Usability is not something that is either on or off; it is relative. I use the ISO 9241-11 definition of usability since it has some strong points: it makes the concept of usability concrete and tangible; it implies that usability is partly objective and measurable, but at the same time subjective, related to attractiveness. The ISO 9241-11 also includes important statements about such things as freedom from discomfort. It is fundamental that usability covers more aspects than just a nice-looking user interface. This is particularly important when developing interactive systems to be used by professional users in their work.

History, Variations and Trends in UCSD

There is no single and exact definition of UCSD. The expression is perceived to be vague and is interpreted differently even within the HCI community. Yet there is an ISO standard on the subject; ISO/IS 13407 *Human-centred design processes for interactive systems*, but the standard describes rather than defines UCSD. When searching for a definition I have found some statements that can serve as good illustrations of the lack of a common definition and understanding of UCSD. Preece et al. define UCSD as:

> *An approach which views knowledge about users and their involvement in the design process as a central concern.*
>
> Preece et al., 1994

Donald Norman wrote back in 1986:

> *But user-centred design emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system.*
>
> Norman, 1986

John Karat from IBM classifies UCSD as:

> *For me, UCD is an iterative process whose goal is the development of usable systems, achieved through involvement of potential users of a system in system design.*
>
> Karat, 1996

Karat continues:

> *I suggest we consider UCD an adequate label under which to continue to gather our knowledge of how to develop usable systems. It captures a commitment the usability community supports—that you must involve users in system design—while leaving fairly open how this is accomplished.*
>
> Karat, 1997

Dennis Wixon, formerly at Digital Equipment, now at Microsoft, has stated:

> *A user centred design process is one that sets users or data generated by users as the criteria by which a design is evaluated or as the generative source of design ideas".*
>
> Cited in: Karat, Atwood, Dray, Rantzer & Wixon, 1996

The book *User Centred System Design* by Norman & Draper in 1986 is usually acknowledged as a pioneer work on the subject. The book, still worth reading, introduced a couple of new concepts that were fairly innovative (it had the subtitle New Perspectives on Human-Computer Interaction). Donald Norman wrote one of the most cited book chapters in HCI: *Cognitive Engineering*. In this chapter he laid out some theories and premises that are fundamental, and today natural, when talking about UCSD. One example is how the theory of mental models (Johnson-Laird, 1983), affects systems design. In short, a mental model describes, for instance, the user's understanding of how a system is assembled. Our mental models are unique for each individual and we *cannot see* another person's mental model or our own. But, we can use methods to try to understand some aspects of the users' model of and expectations on the system.

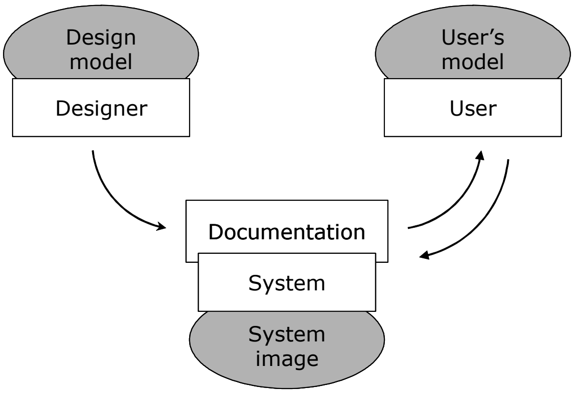


Figure 5: The designer should aim at having the User's model be compatible with the underlying conceptual model, the Design model (Norman, 1986).

The *Design model* is the conceptual model of the system, held by the designer. The user develops a mental model of the system—the *User's model*. The user model is not formed from the design model. It results from the way the user interprets the *System image*. The system image is the physical image[5] (i.e. user interface) of the computerised work situation (including the documentation and instructions). The design challenge is to design a system so that the user can develop a mental model of the system that is consistent with the design model. The design model is supposed to be based on the user's task, requirements and capabilities. It must consider the user's background, experience and the powers and limitations of the user's information processing mechanisms.

Further on, Norman made his "prescriptions for design"[6]. One of those prescriptions was to do user-centred system design:

> *Do user-centred system design: Start with the needs of the user. From the point of view of the user, the interface is the system. Concern for the nature of the interaction and for the user—these are the things that should force the design. Let the requirements for the interaction drive the design of the interface, let ideas about the interface drive the technology. The final design is a collaborative effort among many different disciplines, trading off the virtues and deficits of many different design approaches.*
>
> Norman, 1986, p. 61

Norman's theories are important as they help us understand that all users are different and that we all interact with technology in different ways. We all establish our own mental model for how the system works. In order to understand usability problems, we have to use special methods to study users in action. Norman has attracted some criticism for treating users as passive objects of study and not inviting them into the design process (e.g. cooperative design and the Scandinavian school; Greenbaum & Kyng, 1991). But, his theories and discussions about UCSD are important.

Donald Norman was not alone in approaching UCSD. People at the IBM Thomas J. Watson Research Center (John Gould is probably the most well-known) had for some time experimented with methodologies for developing usable systems. The most significant project was perhaps the 1984 Olympic Message System[7] (Gould, Boies, Levy, Richards & Schoonard, 1987). This was a significant development effort where the aim was to design a usable system, but also to test a user-centred methodology. They were successful on both

[5] Including whatever physical devices are available: knobs, dials, keyboards, mouse and displays.

[6] Norman talks about these prescriptions as: "The general ideas and global framework lead to a set of overriding design guidelines, not for guiding specific details of the design, but for structuring how the design process might proceed" (Norman, 1986, p. 59).

[7] The Olympic Message System provided a voice mail message service. Kiosks were placed around the Olympic village, allowing athletes to send and receive voice messages among themselves, but also to receive congratulations, etc. from around the world.

accounts. The system was truly usable and the UCSD principles that guided the process proved to work (more on these principles later in this chapter). At Digital Equipment, Dennis Wixon and Karen Holtzblatt started to work on something they called Contextual Inquiry and Contextual Design (Wixon & Holtzblatt, 1990), later evolving to Customer-Centred Design (Beyer & Holtzblatt, 1998). They were influenced by ideas from Scandinavia known as the Scandinavian school (or approach). This school was introduced to a broader audience by the book "Computers and Democracy: A Scandinavian Challenge" by Bjerknes, Ehn and Kyng (eds., 1987) and a doctoral thesis by Pelle Ehn: "Work-Oriented Design of Computer Artifacts" (Ehn, 1988). The Scandinavian school emphasises the importance of having users participate in the development on equal terms with the developers. It contains a set of perspectives and practices for emphasising the role of users as active participants in the process of designing computer artefacts that have effect on their lives, in and out of the workplace. It takes its stance in strong labour unions, acting as advocates for workers, and a history of socio-technical approaches arguing for the importance of the social dimension of work with technology. The legal and democratic rights of the users to influence and control the work situation are very important in the Scandinavian school. Users should be in control of the design process, not only involved. Greenbaum & Kyng coined the term *Cooperative design* as they summarised and elaborated on the Scandinavian approach (Greenbaum & Kyng, 1991). In North America the term *Participatory design* is commonly used for the theories based on the Scandinavian approach (see Schuler & Namioka, 1993). Yet others, such as Catterall et al., take a socio-technical standpoint: "User-centred design implies the active participation of users in the design process […] [it] should be comprehensive and not simply part of an end-process evaluation procedure" (Catterall, Taylor & Galer, 1991, as cited in Preece et al., 1994 p. 376).

None of these, very important, landmarks of UCSD approaches have become widely adopted and applied as systems development processes. Contextual design is maybe the most widespread and process-oriented one. But, it should be noticed that these approaches reflect a certain attitude or philosophy rather than a development process. In parallel with UCSD, *usability engineering* has evolved, offering a more pragmatic approach to developing usable systems. Usability engineering has been defined as "a process whereby the usability of a product is specified quantitatively, and in advance. Then, as the product is built, it can be demonstrated that it does or does not reach the required levels of usability" (Preece et al., 1994, p. 650). Usability engineering has adopted the general components of software engineering to provide an engineering-oriented process for the development of usable systems. The major book on the subject is probably Jakob Nielsen's "Usability Engineering" (1993). Usability engineering covers a broad range of techniques for analysing users, establishing usability goals, evaluating design, etc. Such techniques are part of any UCSD process. The best effort to define a usability engineering process that I have found is *The Usability Engineering Lifecycle* by Deborah Mayhew (1999). Mayhew frames usability activities into a lifecycle perspective. But, usability engineering is not the same as UCSD. While usability engineering specifies some hands-on activities for certain steps in a UCSD process, the framework for UCSD covers a much wider range of activities and approaches.

Today, major software companies promote UCSD as one of the most important success factors for their products. IBM continues on the UCSD theme with their Ease of Use approach (see, http://www.ibm.com/easy/). Even Microsoft actively promotes UCSD (see, http://msdn.microsoft.com/ui/) as evidenced in the following statement: "Effective interface design is more than just following a set of rules. It requires a user-centred attitude and design methodology. It also requires early planning of the interface and continued work throughout the development process". That is a rather interesting statement coming from a company infamous for their "function-centred" approach.

What can be learned from this short historical odyssey? Firstly, that there is no unambiguous definition of UCSD, and secondly that there is no *single* UCSD process. It would be easy if we could specify a simple step-by-step description of how to do UCSD. But, the fact is that designing a usable interactive system requires a sequence of activities, using appropriate methods for each of the activities. These activities and methods differ from organisation to organisation and may even differ from project to project. UCSD is a *process* and requires a user-centred *attitude*.

In my research and practice I have found two resources that set the scope for UCSD and are widely accepted as fairly indisputable: the international standard ISO/IS 13407 *Human-centred design processes for interactive systems* (1999) and the principles of designing for usability by Gould et al. (1997). The ISO/IS 13407 standard is a framework for those managing design processes and provides guidance on sources of information and standards relevant to the user-centred approach. It provides an overview of user-centred activities. It does not provide detailed coverage of the methods and techniques required for UCSD. The design principles by Gould et al. are on quite a "high level" and point to the most important ingredients in a user-centred approach. They do not describe a detailed design process. The principles are summarised in Gould et al. (1997) but have evolved over the years as documented in Gould et al. (1983, 1985 and 1987). This is the final set of four design principles:

- Early – and continual – focus on users.
- Empirical measurement.
- Iterative design.
- Integrated design – wherein all aspects of usability evolve together.

As John Karat from IBM puts it: “What best defines user-centred design? I do not think that I have found anything better in print than chapters by Gould and by Whiteside, Bennett and Holtzblatt, though neither paper has UCD as a label”. (Karat, 1996. Authors note: Karat is referring to chapters in Helander, Landauer & Prabhu, 1997 and Helander, 1988, respectively).

We have searched for a definition of UCSD, analogous to the ISO/IS 9241-11 definition of usability, but we have not found any one serving our purposes for communicating what UCSD means. Therefore, we have set out to describe the essence of UCSD by means of a definition and a set of key principles. They are based on ISO/IS 13407 and the principles by Gould et al. as well as the ideas of cooperative design in the Scandinavian school. We believe that by providing a precise definition of UCSD we can, for instance, avoid problems with ambiguity and vagueness and argue against the use of approaches that are not user-centred. The key principles are described in paper 1, and further discussed in the next chapter *Guidance to UCSD and Discussion on Lessons Learnt*.

Focusing on Design

There is an increasing interest in the field of *interaction design*. It is sometimes viewed as the next step after UCSD and even HCI. Terry Winograd’s edited book *Bringing Design to Software* (1996) is mentioned by many researchers and practitioners as a kind of manifesto recognising the importance of interaction design in systems development. Alan Cooper (Cooper, 1999), an influential consultant, has promoted interaction design for years. Furthermore, in *Designing interactive Systems*, Benyon et al. (2004) combine the human-centred (or user-centred) approach with the creative approaches of different design schools. They call their approach Human-Centred Interaction Design. In the book *Interaction Design, Beyond Human-Computer Interaction*, Preece et al. (2002) describe interaction design as: “designing interactive products to support people in their everyday and working lives” (ibid, p. v). They specify three characteristics of the interaction design process: user focus, measurable usability criteria and iteration. This fits well with the description of a UCSD process. Yet others call for a more design-oriented approach, moving away from formal processes towards an emphasis on exploring new territories. One example of this is expressed in the following quote:

> *In recent years, information and communication technology has taken on whole new meanings in Western society and everyday life. Computers were once productivity tools for industry and administration, clearly alien to the personal life-worlds of most people. The goal of so-called user-centred approaches was to find digital ways of supporting work and enterprise goals more efficiently. But then something happened. Electronic mail and web surfing became everyday household activities. Computer games grew in popularity to become one of the major sectors in commercial entertainment. Mobile digital services are transforming the way we communicate, share, exist together. Ubiquitous computing is gradually infusing a digital infrastruc-*

> *ture into our personal spaces. In this situation interaction design is emerging as a new and challenging design discipline. This discipline has a design-oriented focus on human interaction and communication mediated by artefacts.*
>
> Ehn & Löwgren (2003)

This is indeed an interesting statement from two influential researchers, having implications for the future. But I feel the need to make some remarks. The authors give the impression that computers as we know them are history; that is not true for a majority of people in working life. In Sweden, 66 % of the total workforce use computers in their work. About 30 % – 35 % use computers at least half of the working day. For office workers, the overall penetration of computers is about 90 % (figures from Åborg, 2002). Ehn and Löwgren also claim that user-centred approaches are not interesting any more, as if they can only be used in the development of "productivity tools for industry and administration". On the contrary, I argue that UCSD approaches are needed more than ever, but not solely for developing "productivity tools" but also for developing usable tools (systems), not isolating users from their social context.

The interesting part of their statement is the call for more design-oriented research and development, and the exploration of emerging techniques to facilitate digitally mediated human interaction. But rejecting the large number of professional users, using interactive systems at work, together with all user-centred approaches, does not seem fair or wise. My point is that interaction design is very interesting and should be explored, but that does not necessarily mean that all other approaches are obsolete.

Re-labelling UCSD to interaction design, which is what Preece et al. (2002) do, in my opinion, does not really add anything new to the field. I am not saying that there is nothing new about interaction design, only that putting forth interaction design must mean something inherently new and make a substantial difference. There are numerous good examples where interaction between people is enhanced through innovative designs and devices. This is for example the case in research and development targeting public meeting places, real as well as virtual, and in projects aiming at seamless integration of technology into the real world, making the technology "invisible" – ubiquitous computing and augmented reality.

Designs such as the device in Figure 6 could be the result from interaction design as well as from a UCSD process. A key question is how ideas for new styles of interaction and "gadgets" arise. An important research field is therefore to find techniques, tools, environments, etc. that foster and stimulate the generation of new ideas.

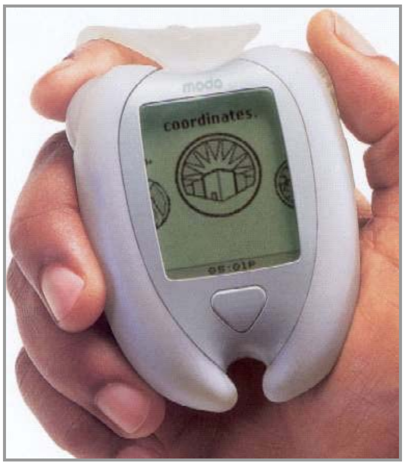


Figure 6: A handheld and wireless device that can be used to obtain accurate information about what is going on in for example a city[8].

New areas of design certainly call for other approaches than the ones found within UCSD. Structured processes and empirical methods may not be applicable when designing computer games or inventing the next generation of consumer electronics, or home appliances – inventing something beyond the existing.

From my rather subjective point of view, one major difference between UCSD and the emerging field of interaction design is the relationship between the process and the individual designer. I favour the process and explicit user involvement, but I also recognise the value of having skilled and experienced interaction designers.

To Sum It Up

The political and democratic issues that were an inherent part of the original Scandinavian school are nowadays toned down. Today, computers are everywhere and they are no longer considered solely as tools for improving productivity in the industry. People use computers as a tool at their workplace, and at the same time as a tool in their homes for various things such as paying bills, playing games, buying books, booking flight tickets and education.

Starting with the *objective* perspectives of human factors and ergonomics, via the approaches in cooperative design and the Scandinavian school emphasising

[8] This was a wireless device, called MODO, developed by Scout Electromedia in 2000. It was a simple yet stylish portable "lifestyle device". The Palm-sized and egg shaped device's graphically rich design offered a daily guide to bars, restaurants, clubs/DJs, live music, movies, sporting events, coffee shops, museums, shopping, etc. Unfortunately the company went out of business during the launch of the device. But, the MODO remains as an example of innovative design.

social and political awareness, we now see *subjective* design-oriented approaches emerge. The terms used to illustrate the different trends range from usability engineering to cooperative and participatory design, to contextual design to UCSD and interaction design. The focus seems to have shifted from processes and user involvement to the genius interaction designer. The individual has become more important than the team.

But, relating to the systems development process is essential to me in my research. Some of the approaches mentioned above do this, but most of them do not. They do not even set out to do so. They are attempts to focus on a specific aspect of a complex process or provide a single method or technique. Relating and integrating UCSD with systems development and software engineering is about weaving them together, not defining silos. Systems development is teamwork and involves large numbers of people from various disciplines. It must be built on processes that encourage teamwork and have a built-in ability to link activities together and make the hand-over between different people and phases as smooth and effective as possible. *Traceability* as well as *accuracy* are words of honour in systems development. Taking on UCSD is a step by step approach, presumably leading to a paradigm shift in moving from a technical-centred approach to a user-centred approach, and possibly beyond UCSD to something new and even more promising.

Guidance to UCSD and Discussion on Lessons Learnt

> *From Product to Process in Research and Design: By this I mean that more attention needs to be paid to the process of design, to working with users in all stages of design, to see the iterative nature of design, and the changing conception of what one is designing as a result of the process itself. This is in contrast to a view of design that proceeds from a set of fixed requirements without iteration, and without involvement of the users.*
>
> Liam Bannon "From Human Factors to Human Actors". In: Greenbaum & Kyng , "Design at work: Cooperative Design of Computer Systems", 1991, p. 34.

In this thesis I set out to give insight in and guidance on how to develop usable interactive systems. We cannot take usability for granted, but we can help in making it more likely for usable systems to emerge, and make it easier for organisations to understand the actions needed in order for this to happen. The definition of usability as discussed in the previous chapter is a good starting point, and so are the various user-centred approaches mentioned. But they must be disseminated, recognised, accepted and put into practice. My research and practice focuses on the challenge to take the field further and make UCSD and usability applicable in practice, and accessible to development organisations. This thesis provides a set of reasons and underlying principles for focusing on usability in systems design. It furthermore describes what it takes to work with UCSD. My conclusion after 20 years of practice and research in the field of UCSD and HCI is that the *attitudes* of the people involved and the development *process* make up the most essential pieces in the complex puzzle of systems development. This puzzle can, in the best of situations and settings, produce technology that truly supports people and business. The attitude affects the individuals' contribution to the whole and the process tells us how to put the individual contributions together to create effective and fruitful teamwork.

My main research question is (see the chapter *Research Objectives*): *how to develop usable interactive systems in practice?* This is a multifaceted question and it does not have a single answer. It takes a rather complex combination of factors to align a business and organisation to be ready to address this issue. It is my experience that the combination of factors differs from organisation to organisation and often even from project to project within the same organisation. Although I cannot give any guarantees or provide a full theory for how to develop usable interactive systems, I hope that this thesis gives an understanding of the underlying complexity and provides some new knowledge and guidance for organisations as well as the individual practitioner, looking for answers and new approaches.

My Main Contributions

I believe that my main contribution is *knowledge and insights about what UCSD is and how it can be put into practice, systematically gained from real-life situations*. The concrete results from my research are: the proposal of a clear definition of UCSD and a set of key principles encompassing UCSD; a process for usability design and the usability designer role. These are all results from my efforts to find new ways of framing UCSD and making it work in practice. The papers in the thesis contain examples and illustrations of how we have explored UCSD in different contexts. Other results from my research and practice are harder to articulate as they are less tangible, but nevertheless equally important. W*e have raised the awareness of usability and improved the practice of UCSD in a number of organisations*. Cooperative design workshops (see Figure 7) offer one way of introducing UCSD practices in an organisation. This type of UCSD activities are representative for the kind of changes that I introduce in the development process in my action research and in my own practice.

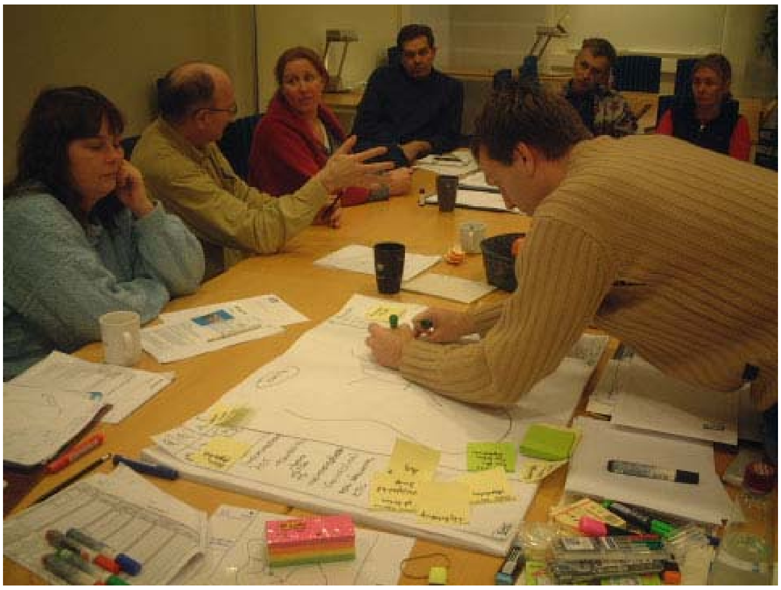


Figure 7: A user is illustrating his ideas of the system during a cooperative design workshop – directly in a paper prototype.

Another improvement is the introduction and establishment of the usability designer role in some development organisations. Yet another valuable result is the process of learning. When helping organisation to introduce and apply UCSD, I find that they learn a great deal. The knowledge and the experiences are specific for the organisation and will help them improve their own practice and business. As a side-effect, I have also been able to enhance my own work practices. I have learned a great deal by practicing in various contexts and in a number of different organisations. I believe that this has been very useful to me in my research.

Further spin-offs from my research and practice are the seminars, tutorials, workshops, university courses, etc. that we have developed and held over the years. Another spin-off is the Swedish textbook on UCSD written by myself and Jan Gulliksen (Gulliksen & Göransson, 2002). The fact that there is little literature targeting how to practice UCSD, lead us to write the textbook. We compiled our research results and lessons learnt, and tried to write a comprehensive textbook that would be easy to understand and appeal to students as well as practitioners. The textbook and this thesis are attempts to improve the situation described by Karat & Karat (2003): "We do believe that we have come a long way from the old days in understanding what human-factors practitioners can (and should) do to contribute to creating usable systems, but it is currently more in the heads of experienced practitioners than in the HCI literature" (ibid, pp. 538–539). Our textbook is used as a course book at several Swedish Universities and even in other countries (although it is currently only available in Swedish).

My research is also used as input in the development of new curricula for computer science and engineering students at our department. How to integrate HCI, usability and UCSD in the higher education is an interesting and vital question that is currently addressed at many universities (see for instance Seffah, 2003). By teaching the students about UCSD and the philosophy behind it fairly early in their education, we hope to raise their awareness of the importance of usability and that, once they find a job, they will have an impact on the attitudes in the industry.

In the next sections I briefly reflect on my own research and practice, with the hypotheses described in *Research Scope and My Aims* providing a background:

- Making people, project members and organisations aware of usability and UCSD will increase the chances of improved usability in interactive systems.
- Introducing methods and processes, based on a set of principles for UCSD that are comprehensible, familiar and make sense to a development organisation, will increase the possibility that the methods and processes will be used.
- Making usability and UCSD explicit and even appealing, i.e. through a set of key principles, a process and a development role, will make it more likely that these concepts receive proper attention and even become demanded.

I realise that the hypotheses cannot simply be confirmed or rejected. They have to be discussed and reflected upon.

Reflections: the Key Principles for UCSD

A key question for an organisation focusing on usability is how to adopt a user-centred approach that fits their organisational context and their development environment. This topic is discussed in several papers in this thesis. I propose our definition of UCSD and the key principles as a common ground and a kind of general plan for how to introduce and apply UCSD. The key principles are outlined, discussed and elaborated on in paper 1. I do think that they are important for all the reasons put forth in the paper. They have the potential to guide the development process and to influence and support an organisation in establishing a user-centred attitude.

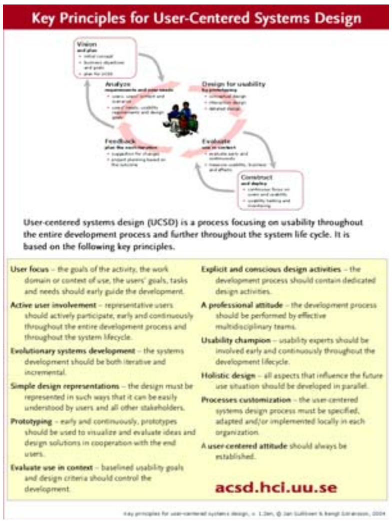
In an attempt to promote the key principles we do encourage people to take our A4-sized poster[9] and put it where everybody can see it and have the time to read it. Here is the short version of the principles as they appear on the poster:

User-centered systems design (UCSD) is a process focusing on usability throughout the entire development process and further throughout the system life cycle. It is based on the following key principles.

- *User focus* – the goals of the activity, the work domain or context of use, the users' goals, tasks and needs should early guide the development.
- *Active user involvement* – representative users should actively participate, early and continuously throughout the entire development process and throughout the system lifecycle.
- *Evolutionary systems development* – the systems development should be both iterative and incremental.
- *Simple design representations* – the design must be represented in such ways that it can be easily understood by users and all other stakeholders.
- *Prototyping* – early and continuously, prototypes should be used to visualize and evaluate ideas and design solutions in cooperation with the end users.
- *Evaluate use in context* – baselined usability goals and design criteria should control the development.
- *Explicit and conscious design activities* – the development process should contain dedicated design activities.
- *A professional attitude* – the development process should be performed by effective multidisciplinary teams.
- *Usability champion* – usability experts should be involved early and continuously throughout the development lifecycle.
- *Holistic design* – all aspects that influence the future use situation should be developed in parallel.
- *Processes customization* – the user-centered systems design process must be specified, adapted and/or implemented locally in each organization.
- *A user-centered attitude* should always be established.

[9] A version of the poster is in an appendix to paper 1.

Packaging your message in an appealing way is important and often neglected by researchers. Our poster and the short description of each principle are attempts to do that (see the thumbnail to the left). A visually appealing presentation format and thorough information design can attract those who would not bother to listen to your well-prepared, but long and tedious talk with dozens of slides. Even though the content is the main thing, it can be enhanced and benefit from marketing.



Key Principles for User-Centered Systems Design

User-centered systems design (UCSD) is a process focusing on usability throughout the entire development process and further throughout the system life cycle. It is based on the following key principles.

User focus – the goals of the activity, the work domain or context of use, the users' goals, tasks and needs should early guide the development.

Active user involvement – representative users should actively participate, early and continuously throughout the entire development process and throughout the system lifecycle.

Evolutionary systems development – the systems development should be both iterative and incremental.

Simple design representations – the design must be represented in such ways that it can be easily understood by users and all other stakeholders.

Prototyping – early and continuously, prototypes should be used to visualize and evaluate ideas and design solutions in cooperation with the end users.

Evaluate use in context – baselined usability goals and design criteria should control the development.

Explicit and conscious design activities – the development process should contain dedicated design activities.

A professional attitude – the development process should be performed by effective multidisciplinary teams.

Usability champion – usability experts should be involved early and continuously throughout the development lifecycle.

Holistic design – all aspects that influence the future use situation should be developed in parallel.

Processes customization – the user-centered systems design process must be specified, adapted and/or implemented locally in each organization.

A user-centered attitude should always be established.

acsd.hci.uu.se

Key principles for user-centered systems design, v. 1.2en, © Jan Gulliksen & Bengt Göransson, 2004

Reflecting on the impact of the key principles, my main impression is that of positive feedback, but there remains also a reluctance to bother with UCSD. Moreover, my experience is that the key principles are primarily valuable on an organisational level; defining and promoting UCSD, and giving an insight into the implications for the development processes and organisations. I believe that the key principles have to be translated into a process to make them applicable in practice. It is impossible for me to tell how many organisations or projects that use the key principles as guidance, but from contacts with a large number of organisations I can tell that they are much appreciated. It is more difficult to approach organisations and individuals that do not care about UCSD at all. But, whenever possible, I try to point to some of the UCSD features that can be applied without too much trouble.

I cannot guarantee that the current set of principles is correct or complete, or even the best possible. But, they are derived from trustworthy sources (such as ISO/IS 13407 and Gould et al.) and thoroughly grounded. The current set has been around for about two years and it seems fairly stable. Practitioners sometimes tell us that they think that twelve key principles are too many. It is difficult to communicate such a large number of principles. But, my experience is that our twelve principles offer better support than for instance the principles in ISO/IS 13407 or Gould et al. I could even consider including more principles, as long as they contribute to the whole and make sense. But then, we would perhaps need new ways of communicating them. I also think that it is essential to create activity lists, complemented with tools and techniques, for each principle, as described in paper 1. The activity lists will improve and facilitate the translation of the key principles into a process.

Reflections: the Process for Usability Design and the Usability Designer Role

> *Although there is a substantial body of human factors, and ergonomics knowledge about how such design processes can be organised and used effectively, much of this information is only well known by specialists in those fields.*
>
> ISO/IS 13407, 1999

Many of my papers are concerned with the usability design process and the usability designer role, primarily papers 2, 3, 4, 5 but also the three design cases. Early on in our research, we realised that UCSD must be made concrete and tangible in order to be applicable in practice. But we were also well aware that UCSD is difficult to formalise and visualise. We hypothesised that defining a process and a role would make it easier to discuss and argue for UCSD, even if they are only parts of the big picture. The process and the role have been integrated parts of the work practices at Enea Redina AB for some years now. And we know that they are applied regularly in other organisations and by individual practitioners. But I cannot tell exactly to what extent. They are described in our textbook on UCSD, and they are significant parts in our seminars, courses and tutorials.

Promoting UCSD as a process does not mean promoting one single process. There are several processes, depending on the context. We use a standalone figurative process in order to illustrate the principles (as in the UCSD poster). We use the usability design process in organisations and projects, either as standalone or integrated with existing processes. And we use the usability design discipline as an integrated part, a so called plug-in, of the Rational Unified Process (RUP)[10]. The outline of the processes in these three contexts is almost the same, but they differ in details. Describing the development process at the level of “boxes and arrows” is important, in particular when dealing with development organisations and software engineers. But when reaching that level of detail, you often lose the “big picture” and unfortunately also some of the advantages with UCSD.

There are several benefits with the usability design process (see paper 2 for a detailed discussion). Perhaps the main advantage is that the process is communicative; it is not too detailed, but at the same time not too abstract and it can serve well as an explanation model. However, the process does not cover all aspects of UCSD or our key principles. It provides a simplified view of

[10] “The Rational Unified Process or RUP product is a software engineering process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.” (Rational Software Corporation, 2003, the Overview page)

UCSD and organisations and projects need additional support to become fully user-centred. However, it is my experience that the process effectively communicates how to practice UCSD and also works well in the hands of a usability designer in a project. It is more difficult to apply the process if you do not have a background in usability and some experiences with usability methods. The usability design process and the role are closely connected, even if they do not rely on each other. My observation is that the usability design process is generally perceived as being straight forward and easy to understand and practice.

How does our usability design process compare to other UCSD processes? I have studied four user-centred approaches (even though none of them is labelled user-centred) used by development organisations and individual practitioners: customer-centred design (Beyer & Holtzblatt, 1998), usage-centred design (Constantine & Lockwood, 1999), goal-directed design (Cooper, 1999) and Mayhew's (1999) usability engineering lifecycle. They are all proprietary processes used by their inventors as consultants. They are available as textbooks and courses, but are not off-the-shelf products. Personally, I have never seen these approaches fully used in any organisation, only parts of them. But, they do have an impact on organisations and practitioners in that they have contributed some extensively used UCSD methods: contextual inquiries from customer-centred design; essential use cases from usage-centred design and personas from goal-directed design. Mayhew's process, on the other hand, does not include any new methods, but provides a collection of existing methods. It is hard to compare these processes with our usability design process as they all have their pros and cons. To a great extent, we see that they reflect the inventors' views on the systems development process and what usability is about. We have tried to create the usability design process so that it reflects our view on UCSD. It does not contain any new activities or methods, but a combination of activities that we believe support a UCSD approach. We have been influenced and inspired by Mayhew's process (see paper 2), but we have also integrated parts from customer-centred design and goal-directed design. One major advantage with the usability design process is that it is defined on such a level that it can fit into other process frameworks quite easily and with minor modifications only. This makes it relatively easy to get acceptance for the process. One example of this is the plug-in to the RUP described in paper 2 and 3.

The RUP is the predominant development process in Sweden. Actually, it is the only off-the-shelf development process that is widely used by development organisations. It seems that those who do not use the RUP use some in-house development process or no explicit process at all. We therefore believe it is important to relate our research to the RUP. Our plug-in to the RUP has attracted a great deal of interest in many organisations in Sweden but also in other countries within Europe and in the US. However, making the RUP more user-centred is not without its problems. The RUP is inherently engineering-oriented and I regard our plug-in as a compromise and an effort for HCI to meet software engineering on its own turf. However, to be successful you sometimes need to find allies.

The usability designer role is less precise and need specific adjustments and refinement to fit into organisations and development projects. Even though the role is perceived as somewhat fuzzy, it seems that those organisations adopting it take it seriously. The role includes championing usability, which seems to be well accepted by usability practitioners. They usually take it on with enthusiasm.

As the results in paper 4 indicate, the usability designers do make a change in favour of UCSD and are generally accepted and appreciated, even though they are not yet a natural part of the development process. The usability design discipline in the RUP (paper 3) is an attempt to solve this problem.

Moreover, paper 4 describes one important finding; the usability designer sometimes finds herself in the position of being seen as a substitute for the users. Whenever managers and developers need to know something about the users or the use situation they go to the usability designer instead of contacting the users directly – the usability designer becomes a mediator. But, one of the rationales for the usability designer is to act as a facilitator rather than as a mediator. It would be better if the usability designer could act as a facilitator instead and “arrange” facilitated meetings, see Figure 8.

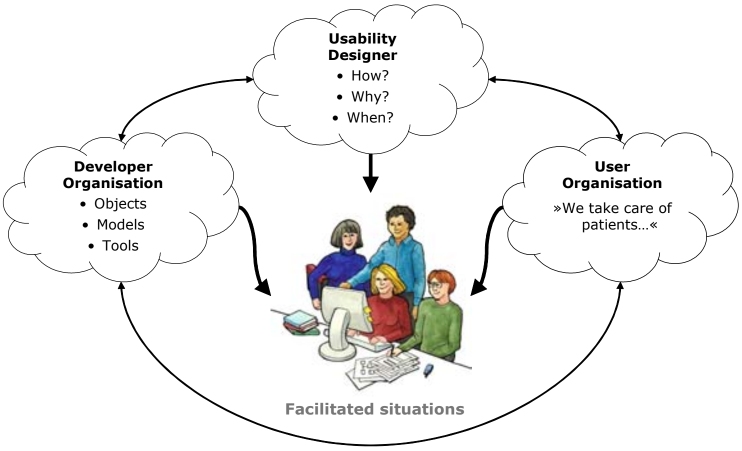


Figure 8: The usability designer as a facilitator rather than a mediator.

I want to avoid the role becoming too much of a mediator. Therefore I will investigate how to encourage the facilitation aspects and discourage the mediation aspects in the role. Acting as a mediator may be inevitable in certain circumstances. But in general, as expressed in the key principles, all participants should in one way or another be involved with users.

I am very pleased to see that the usability designers in our studies have succeeded in shifting their focus and attention to explicit and deliberate design for usability (paper 4). Since the design focus is one of the rationales underpinning the process as well as the role, I regard this shift of focus as a success. I believe that replacing usability engineering with our more design-oriented approach is a good thing.

Reflections: UCSD and Software Engineering

Working for UCSD to become the "standard operating procedure" and *the way* of developing systems in an organisation has almost become a mission for me. This is something I promote or even "preach" as often as I am given the opportunity. One of the biggest challenges is to promote and practice UCSD in organisations that look upon themselves as software engineering organisations, in contrast to taking on the full scope of systems design.

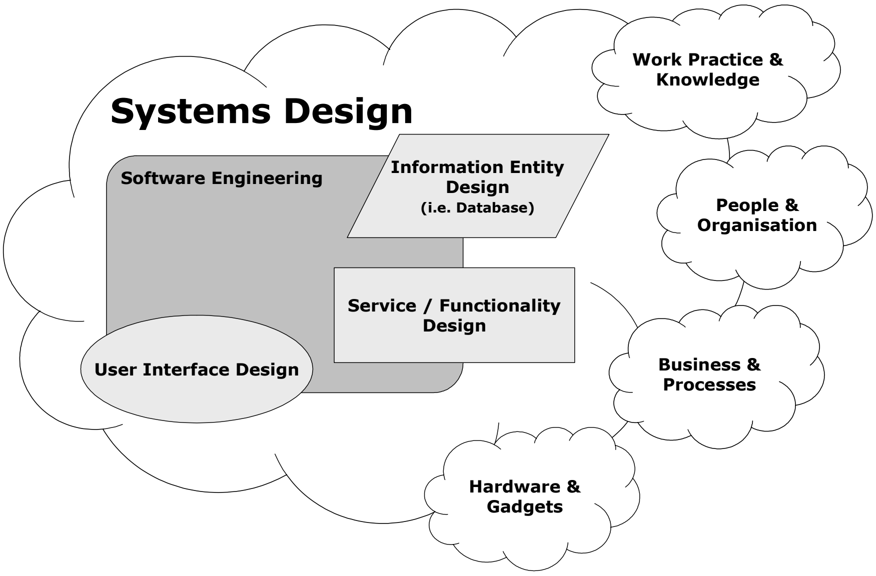


Figure 9: Systems design covers more than software engineering.

Figure 9 is an attempt to illustrate the relationship between systems design (i.e. in my case UCSD) and software engineering. The big cloud illustrates the systems design space. Within that space, software engineering is usually a significant part, but systems design also covers, and must take into account, other aspects (the smaller clouds). UCSD typically focuses on people, work practices, business processes, etc., as well as on software engineering. This is an advantage, and one of the reasons for UCSD as described in *Reasons for UCSD*. But, it can also be frightening for a development organisation. A mature and progressive development organisation can take advantage of the breadth of UCSD and recognise its potential, whereas a more conservative and less dynamic organisation may reject UCSD. Such organisations consider most parts of UCSD someone else's responsibility, i.e. it is up to the business, management and the user organisation to order the system they want to have. Another frightening obstacle is the insight that the development organisation has to incorporate the user's view into the development process and cannot rely on engineering and technical assumptions alone. These organisations prefer to address usability and UCSD in terms of user interface design (see the boxes in Figure 9). This seems to make their lives easier and they strive for clear boundaries and responsibilities. What they fail to realise is that the software of a system cannot be treated as an isolated part; it has to be developed in conjunction with all the other parts that make up a dynamic, professional and successful organisation or business.

Bridging the gap between UCSD and software engineering is a challenge and an interesting topic for further research. It is debated at conferences and addressed in journal articles. In my research and practice I have tried to adhere to my beliefs that this is not only about adding some usability activities (i.e. usability engineering) to the process. It is about taking on the full scope of UCSD as defined and described in our key principles. Papers 2, 3, 5 and 6 describe and explain our criticism of many development organisations and processes for having too narrow a view on systems development. In paper 2 we outline some of the activities that we consider important or even mandatory when changing organisations, processes and attitudes towards a user-centred philosophy. In short they cover: usability must be integrated in the development process; usability professionals must learn about software engineering; software engineers must learn about UCSD and acknowledge the power of involving users. Too often, in my opinion, the gap between UCSD and software engineering is supposed to be bridged by simply treating UCSD as usability engineering. As mentioned earlier, our work on UCSD and the RUP is an attempt to do something that is applicable and relevant to development organisations but that goes beyond usability engineering. Thus, I have gradually come to the conclusion that this is not a question of UCSD meeting software engineering on their turf. To me, it has become clear that ”bridging the gap” is about meeting on a new ground rather than building bridges between points – i.e. to share a mutual understanding and practice. I have, through my research, tried to clarify the concept of UCSD and how it can be practiced, but we need to further explore the new meeting ground. The key principles for UCSD are our main contribution to constituting a common ground and also providing a starting point to build from.

At Enea Redina AB, we have made concrete attempts to explore how the mutual understanding between software engineering and UCSD can be enhanced. The company is an IT consultancy (about 35 consultants) focusing on contract IT development and UCSD (primarily complete development projects, building administrative IT systems for specific organisations). There is management support for UCSD and one of the major success factors has been the key principles for UCSD providing a common ground. Two years ago all consultants attended an internal course in UCSD. The objective for this initiative was to get a broad understanding and acceptance of UCSD among all developers. By making the course mandatory, the company made explicit its ambition to invest in promoting, establishing and institutionalising UCSD. The objectives for the course were: "To give the consultants a general knowledge in user-centred systems design, how it can be applied and what the objectives are. Furthermore, the course aims to give the consultants insight in the concepts behind Human-Computer Interaction; how to design an effective user interface and introduce the basic theories for human perception and cognition". The second phase, supporting the mutual understanding and practice, is in progress right now, under the name "a thematic year on system architecture". It includes courses in object-orientation, design patterns, etc. All the consultants participate in this phase, including our usability designers. Our hope is that these measures along with extensive practice will help us in eventually defining the new meeting ground. Karat and Dayton (1995) argue for educating all developers in usability. This is well in line with my experiences, but I also argue that usability designers need to know more about software engineering.

Reflections: User Involvement

To my mind, reflections on how to *actively involve users* merit their own section. User involvement seems to be controversial and this is never more evident than when comparing a UCSD approach to software engineering processes. The reasons for involving users can vary, see the earlier chapter about reasons (*Reasons for UCSD*), but it is a must to involve users in the development of interactive systems. The degree to which users should be involved and how are subjects for theories as well as practical considerations. Users can be treated as anything from passive objects of study, to fully empowered decision makers, actively designing their future system as well as their work practices. It is hard to maintain a consistent and unambiguous view on how users should be involved in practice. The *design ideals* that emerged in the Scandinavian school during the 1970s and 1980s with roots in the democratisation of working life and with clear political implications (see Greenbaum & Kyng, 1991, pp. 1–2) are not that easy to accomplish:

- Full participation of the users.
- Computer systems should enhance workplace skills.
- Computer systems are tools, and need to be designed so that they are under the control of the people using them.
- Computer systems must support both productivity and quality.

- The design process includes conflicts.
- The use situation is a fundamental starting point for the design process.

While being firm in theory, we need to be flexible in practice. In practice, the line of actions by which users are involved must be modified from project to project. My view, as communicated in our key principles, is that active user involvement should be preferred in all situations. But, on the other hand, user participation in a project is never, in itself, a guarantee for a usable system. Users must become more than just passive members of the project team. There must be activities, methods and skilled usability professionals in the project to ensure that the users and their contributions are allowed to make an impact on the development, and that the skills and knowledge of the users' are taken care of in the best possible way. Software engineering processes (such as the RUP) tend to discourage active user involvement and be insensitive to the richness of information revealed in cooperation with users and studies of their work situations. Most software engineering processes capture knowledge about the system to build in formal models with a well-specified narrow scope. The Unified Modelling Language (UML) (Fowler, 1997) and other formal model languages can only represent what they are intended to represent; logical relations and flows of events. They cannot accommodate descriptions of, for instance, the context of use, or knowledge about the design space. The development processes furthermore prescribe that all documentation should be modelled on such formal notations, which means that the activities and methods for capturing information are influenced and constrained by them. Rich descriptions of users are substituted by stereotypes such as actors[11]. The users' work context and needs are deliberately stripped down to requirements in the shape of use cases[12]. The idea is to make abstractions with the purpose of being precise and unambiguous, but the information and knowledge that is essential for the design of the system is never captured owing to the process and the formalism in documenting the results. One striking example is the term actor that has changed from being a potentially useful concept (see for example Bannon, 1991: "From Human Factors to Human Actors") to a "skinny" stick figure in UML diagrams that can either represent another system (or just a database) or a "human" role in the organisation. This is pretty far from representing the *active user* approach that UCSD strives for.

[11] "Defines a coherent set of roles that users of the system can play when interacting with it. An actor instance can be played by either an individual or an external system." (Rational Software Corporation, 2003)

[12] "A use case defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor." (Rational Software Corporation, 2003)

I have so far never seen projects with too much user involvement. It is my experience that there can always be additional, and more active, user involvement. This is one of the major responsibilities for usability designers to take on, not on their own, but this role is the foremost advocate for the users' interests. Arguing and creating space for the facilitated situations described earlier is crucial in promoting and practicing UCSD. I realise that there sometimes are practical problems in involving users, e.g. finding users that have time to spare. But, I firmly believe that the reluctance to involve users is primarily related to the attitude towards users and their contributions. The attitudes must change in favour of UCSD and active user involvement has to become a natural and prominent ingredient in systems development.

Concluding Remarks

It is difficult to draw an unambiguous picture of the effects our research has had outside our research projects. However, I still think that my three hypotheses for increasing the practice and impact of UCSD make sense on the whole. So far, I have not found any evidence in my research that contradicts the hypotheses. It might be so, that the hypotheses are self-evident, but they are nevertheless important issues to raise within systems development. Making my research applicable in real life settings is important to me, and I can just hope that other practitioners will take the opportunity to apply our research results. Furthermore, I know for certain that, through action research and practice, we have made a difference in the projects and organisations where we have been involved. Project members and organisations have learned from our presence and improved their own practice. It seems that the "trinity" of the key principles, the usability design process and the usability designer role makes a strong case for UCSD. And most likely this has led to more usable interactive systems. However, despite the constructive examples, positive experiences and changes that my research have made there are things that never seem to change. I am primarily thinking about two major obstacles that come back, again and again:

- If the user interface "looks nice", the system is automatically perceived as being usable. This is closely connected to the everlasting problem of usability being seen as a user interface matter alone.
- Developing a system cannot be treated as an isolated and non-social activity. Reducing systems development to a software process, heavily based on engineering principles, seems unfortunately to be in the interest of too many development organisations. This is further supported by development processes such as the RUP, that focus on engineering aspects.

Even though I am generally optimistic about the future, these two obstacles disturb the picture somewhat. We need to further develop and practice UCSD.

But, in addition we need to work on parallel approaches with other theoretical backgrounds and angles.

My advice to those practitioners that are familiar with the situation that I have described here is to continue to promote usability and UCSD – possibly using our key principles, the usability design process and the usability designer role – but to do it slowly and step by step. They also have to take organisational politics into account and find allies on all organisational levels.

More Research Needed

Continuing with the UCSD theme, I think that there are at least two main research topics related to my work that are essential and interesting to further investigate:

- How do project members' attitudes affect the development of usable interactive systems?
- How can we facilitate and increase the practice of cooperative design and active user involvement?

It is obvious that there is still quite a long way to go before active user involvement happens per se. It seems to be difficult to find effective and practical solutions to this problem. The design ideals in cooperative design and the Scandinavian school seem to be far away. I believe that one explanation to these obstacles can be found in the attitudes within development organisations. I have so far focused mostly on UCSD as a process, well aware that it is a matter of both process and attitude. In the future, I will probably focus my research on attitudes. How can we communicate the need for a change in attitude towards users, their role in development and the dominating position of technology? That will be one of my research questions in the future.

The Papers in the Thesis

This chapter briefly describes the papers included in the thesis; their origin and my contribution.

When I wrote my licentiate thesis in 2001 (Göransson, 2001), I tried to cover and put forward as much as possible of what I had explored and experienced so far. As a result it was rather lengthy and contained a lot of material that was not published anywhere else. Since then, my research and practice have focused on three of the most significant issues in that thesis. They are: what it takes to develop according to UCSD; the process of usability design and the usability designer role. This research has resulted in new publications that are part of this thesis (see Figure 10).

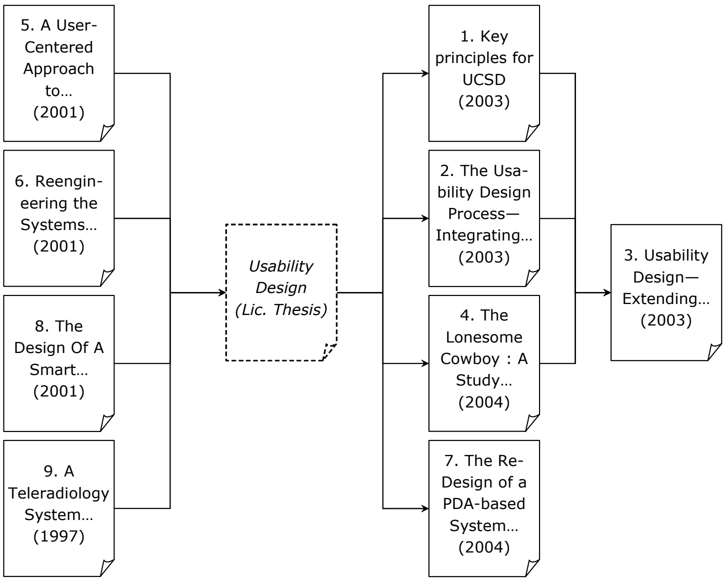

5. A User-Centered Approach to… (2001)
6. Reengineering the Systems… (2001)
8. The Design Of A Smart… (2001)
9. A Teleradiology System… (1997)
Usability Design (Lic. Thesis)
1. Key principles for UCSD (2003)
2. The Usability Design Process—Integrating… (2003)
4. The Lonesome Cowboy : A Study… (2004)
7. The Re-Design of a PDA-based System… (2004)
3. Usability Design—Extending… (2003)

Figure 10: The papers in the thesis.

There is a general plan and rationale behind the paper topics. The papers are a collection of design cases, and studies of organisations and the development process. Furthermore, we have recently turned our attention towards the software engineering community. Some of the latest publications have been published in forums mainly intended for software developers. We have realised that to make a change in development practices we need to directly address those with the most “skin in the game”.

Paper 1

Key Principles for User-Centred Systems Design

This paper is our manifesto and rationalisation for UCSD. In this paper, we try to clarify what UCSD is about and define it. The key principles for UCSD have grown out of the need for being more exact on what constitutes UCSD. An important source was the modelling of the systems development process that we did at the Swedish Tax Agency, described in paper 6. A first draft of the principles was presented in my licentiate thesis (Göransson, 2001) as “heuristics for user-centred design”. The principles are grounded (Goldkuhl, 1994) in relation to existing theories on UCSD and empirically grounded in projects and organisations.

My contribution. I and Jan Gulliksen outlined the first set of principles and I have been active in the refinement leading to the current set of the key principles.

Paper 2

The Usability Design Process—Integrating User-Centred Systems Design in the Software Development Process

The usability designer role (Göransson & Sandbäck, 1999) was the starting point of my work with the usability design process. The process was modelled on the work practices of the usability designer, and is an attempt to compile the best practices of that role. When I reflected on how I worked as a practitioner, I started to model what I was doing and the usability design process was the result. It is related to the Usability Engineering Lifecycle process proposed by Deborah Mayhew (1999), which is no coincidence. I have found Mayhew's process valuable and been inspired by it, but one main difference is a more explicit focus on design in our process.

My contribution. The first outline of the process was presented in my licentiate thesis and I have thoroughly detailed the process through my practice at Enea Redina.

Paper 3

Usability Design—Extending Rational Unified Process with a New Discipline

This paper describes the usability design process as a plug-in to the RUP. To be successful you sometimes need to find allies. In Sweden the RUP is the most popular and widely used commercial development process. We therefore believe that it is important to relate our research to the RUP in our mission to increase the applicability and practice of UCSD. This paper summarises our work on integrating the usability design process in the RUP.

My contribution. The research and practice described in this paper is primarily the outcome of my work at Enea Redina. My colleague at Enea Redina Magnus Lif and I have worked together on this paper.

Paper 4

The Lonesome Cowboy – A Study of the Usability Designer Role in Systems Development

This is a study on how the usability designer role is applied in two development organisations. The role as usability designer was first described in a paper by Göransson & Sandbäck (1999).

Since then the role has been introduced and practiced in several organisations, two of them were in focus during this study.

My contribution. I have contributed to the definition of the role and introduced it in several organisations. I took part in all activities in the study, except the interviews. They were conducted by Inger Boivie and Jan Gulliksen.

Paper 5

A User-Centred Approach to Object-Oriented User Interface Design

This paper is the result of research and practice in many development projects, presented as a chapter in a book. It reviews what we had learned so far, focusing on the UCSD process.

The part concerning user interface modelling (UIM) was prepared by Magnus Lif.

My contribution. My main contributions are: the examination of the RUP and the Dynamic Systems Develop Method (DSDM); the usability designer role and the experiences in promoting UCSD at the Swedish Tax Agency.

Paper 6

Reengineering the Systems Development Process for User-Centred Design

This paper is the result of the modelling of the systems development process at the Swedish Tax Agency. It describes a method for modelling development processes and a case study from the Swedish Tax Agency.

My contribution. Jan Gulliksen and I did the modelling together with people at the Swedish Tax Agency. Jan Gulliksen came up with the idea of doing the modelling using the organisation’s own techniques and I contributed with my knowledge in systems development in general and UCSD in particular.

Paper 7

The Re-Design of a PDA-based System for Supporting People with Parkinson’s Disease

This is a design case from my practice. It contains details of the customised UCSD process that we used in a product development project, as well as details of the interaction design. It also includes reflections on what we succeeded in when applying UCSD, but also what we were less successful with.

My contribution. It is essentially my own practice that is described in this paper.

Paper 8

The Design of a Smart Card-Based Home-Help System

This is a design case from my practice together with some colleagues. It describes how we conducted UCSD in a project with a major focus on visual design. Special attention is paid to the idea of multiple and parallel prototypes.

My contribution. I did most of the planning of the UCSD process. I also participated in all phases: analysis, design and evaluation. Furthermore, I acted as principal usability expert and designer in the development team. I introduced the concept of design criteria, as explained in paper 9.

Paper 9

A Teleradiology System Design Case

This paper describes the design case of the teleradiology application CHILI. This product supports the transmission of radiological images or image sequences between different sites for the purpose of interpretation and consultation. Digitally transmitted images can be presented, analysed and discussed simultaneously at different sites. The paper introduces the concept of design criteria as a method for design. Further it describes how criteria and requirements were transformed into user interface design. The concept of a work task metaphor is central as well as detailed user interface design. CHILI won the European IT Prize Award in 1997.

My contribution. Erik Borälv and I produced the prototype of the teleradiology system CHILI together in Heidelberg, Germany. We put much of our common ideas and experiences into the design of the prototype.

Acknowledgements

Firstly, I would like to express my deep gratitude to my two friends; my supervisor and Associate Professor Jan Gulliksen, and my Professor and the head of the HCI department Bengt Sandblad. Thank you for helping me with this thesis and guiding me in my research. Anders Hektor, my assistant supervisor for the final part of the thesis, has most kindly helped me with new perspectives and insights.

I would also like to express my heartfelt thanks to all my co-writers. I have known most of you for a long time now and I really appreciate your professionalism as well as your warm friendship.

To all my colleagues and friends at the Division of Human-Computer Interaction, Department of Information Technology at Uppsala University: I am sincerely and greatly thankful for your support and help. I particularly would like to thank Inger Boivie for helping me with editing and commenting on many papers, not least on this summary. I am deeply in dept to her and appreciate her expertise as well as her warm personality. I would also like to mention Erik Borälv and Eva Olsson for a long and rewarding professional relationship as well as friendship.

I owe all the people at Enea Redina, my friends and colleagues there, deep and honest thanks for support and encouragement, as well as for interesting and fruitful discussions. I particularly want to thank Björn Eriksson, Managing Director at Enea Redina. He has always supported and encouraged me in my research.

I would like to express my respect for all those individuals whom I have collaborated with in various research and development projects. In particular, all users and project members who have acknowledged and understood the full potential of usability and UCSD.

Parts of my work have been financed by VINNOVA – the Swedish Agency for Innovation Systems, FAS – the Swedish Council for Work Life and Social Research and Skatteverket – the Swedish Tax Agency.

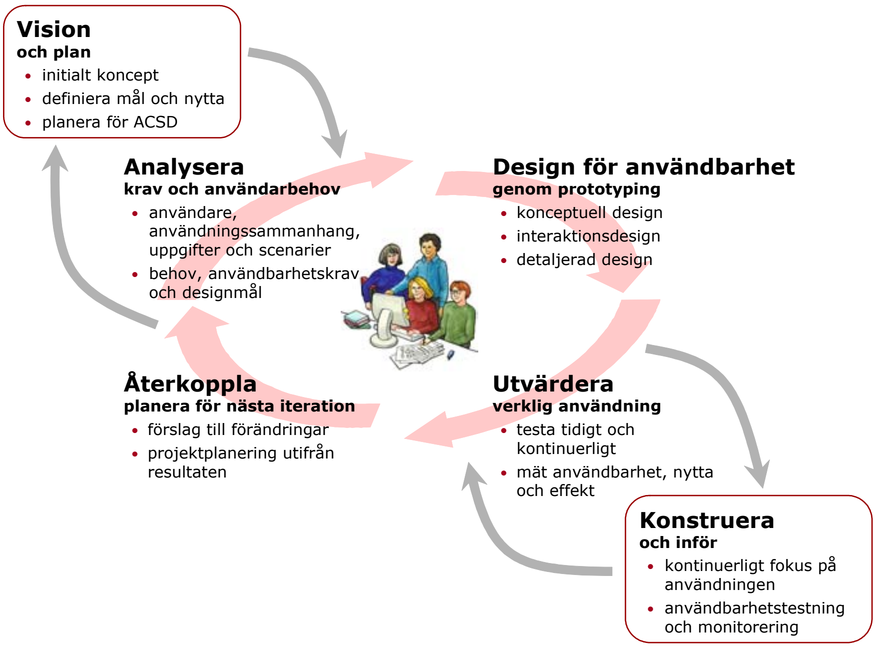
Finally I would like to honour and express my deep love for those who are the most important persons in my life; *Ellinor*, *Carl* and *Emma*. You have all supported me in your own special ways and you are my true spring of joy and energy.

Summary in Swedish

Användarcentrerad systemdesign
– Design av användbara interaktiva system i praktiken

Den här avhandlingen inom området människa-datorinteraktion (MDI) handlar om hur man kan gå tillväga för att utveckla *användbara interaktiva system* – system som stöder och engagerar människor, istället för att vara ett hinder. Utformningen av dessa system är inte bara något som ”uppstår”, utan de har oftast utvecklats med hjälp av någon typ av *användarcentrerad utvecklingsprocess* och med en *användarcentrerad attityd* under hela utvecklingen. I korthet innebär det att ”riktiga” användare och deras; behov, mål, användningssammanhang, förmågor och begränsningar styr utvecklingsarbetet. Detta i kontrast till utveckling som enbart styrs av tekniken. Det handlar mer om att utveckla för det som behövs än för vad som är tekniskt möjligt.

Genom mitt arbete, både som forskare och konsult, har jag studerat, analyserat och jobbat i och med en rad olika typer av utvecklarorganisationer. Det har varit interna IT-avdelningar såväl som produktutvecklande. Många av dessa vill jobba användarcentrerat och har insett viktigt av användbara system, men har haft svårigheter att tillämpa användarcentrerad systemdesign (ACSD) i praktiken. I avhandlingen tar jag upp, diskuterar och visar på hur organisationer kan gå tillväga för att jobba på ett användarcentrerat sätt. Nedanstående principer för ACSD summerar våra forskningsresultat och erfarenheter av att integrera en användarcentrerad designfilosofi i ett stort antal organisationer och projekt. Principerna baseras på bland annat ISO-standarden 13407 (Human-centred design processes for interactive systems), Gould et al.:s principer (Gould et al., 1997), samt våra egna erfarenheter av systemutvecklingsprocesser. Andemeningen i principerna har vi försökt att fånga i figuren nedan.


Vision
och plan
• initialt koncept
• definiera mål och nytta
• planera för ACSD
Analysera
krav och användarbehov
• användare, användningssammanhang, uppgifter och scenarier
• behov, användbarhetskrav och designmål
Design för användbarhet
genom prototyping
• konceptuell design
• interaktionsdesign
• detaljerad design
Återkoppla
planera för nästa iteration
• förslag till förändringar
• projektplanering utifrån resultaten
Utvärdera
verklig användning
• testa tidigt och kontinuerligt
• mät användbarhet, nytta och effekt
Konstruera
och inför
• kontinuerligt fokus på användningen
• användbarhetstestning och monitorering

Definition: *Användarcentrerad systemdesign är en process som fokuserar på användare och användbarhet genom hela utvecklingsprocessen och vidare genom hela livscykeln. Denna baseras på följande nyckelprinciper:*

- *Användarfokus – verksamhetens mål, användarnas arbetsuppgifter och behov skall tidigt styra utvecklingen.* Alla i projektet måste förstå verksamhetens mål, grunden i användarnas situation, deras uppgifter, varför och hur de utför sina uppgifter, etc. Genom detta blir det viktigt att man prioriterar vad som är bra för användarna framför vad som är tekniskt möjligt.

- *Aktiv användarmedverkan – representativa användare bör aktivt delta tidigt och kontinuerligt genom hela utvecklingsprocessen och vidare genom systemets livscykel.* Användarna bör involveras direkt i utvecklingsprojektet men även utanför själva projektets ramar. Det är viktigt att se skillnaden mellan domänexperter och verkliga användare; domänexperterna (väl insatta i verksamheten, men ej representativa som slutanvändare) kan involveras kontinuerligt under hela utvecklingsprojektets gång; slutanvändare däremot bör involveras för mer tillfälliga aktiviteter under analys, design och utvärderingar.
- *Evolutionär utveckling – systemutvecklingen bör vara både iterativ och inkrementell.* Designlösningarna bör kontinuerligt itereras med användarna. En iteration måste innehålla följande aktiviteter: en ordentlig analys av användarnas krav och användningssammanhanget: en designfas och en dokumenterad utvärdering med konkreta förslag till förändringar. Inkrementell utveckling innebär att systemet stegvis utvecklas uppdelat i inkrement som vart och ett levereras till verklig användning.
- *Enkla designrepresentationer – designen skall representeras så att den lätt kan förstås av användare och alla andra inblandade intressenter.* Använd en för användarna bekant terminologi. Så långt som möjligt bör konkreta designrepresentationer som prototyper (alltifrån enkla skisser till mer avancerade mock-up:er) och simuleringar användas för att åskådliggöra den framtida användningssituationen.
- *Prototyping – tidigt och kontinuerligt skall prototyper användas för att visualisera och utvärdera idéer och designlösningar i samverkan med användarna.* Använd pappersskisser, mock-up:er och prototyper för att stödja den kreativa processen; att visualisera idéer och lösningar. Materialet som man använder är viktigt. Börja på låg nivå med t.ex. pappersskisser innan någonting över huvudtaget kodas.
- *Utvärdera användningen i sitt sammanhang – avstämda mål för användbarheten och designkriterier skall styra utvecklingen.* Specificera alltid användbarhetsmål och basera designen på speciella designkriterier. Utvärdera designen gentemot användbarhetsmålen och designkriterierna tillsammans med användare, så långt möjligt.
- *Explicita och medvetna designaktiviteter – utvecklingsprocessen skall innehålla dedikerade designaktiviteter.* Gränssnitts- och interaktionsdesignen är avgörande för ett interaktivt systems användbarhet. Alltför

ofta uppstår designen snarare än att den är resultatet av en medveten och strukturerad aktivitet.

- *Professionell attityd – utvecklingsprocessen skall genomföras av effektiva tvärdisciplinära team.* Olika kompetenser bidrar till helheten, t.ex. systemarkitekter, databasexperter, programmerare, informationsarkitekter, användbarhetsdesigners, interaktionsdesigners, experter på fältstudier, etc.
- *Användbarhetsförespråkare – erfarna användbarhetsexperter skall involveras tidigt och kontinuerligt genom hela systemets livscykel.* Se till att en erfaren användbarhetsexpert (t.ex. en användbarhetsdesigner) är hängiven projektet som en motor för den användarcentrerade designprocessen från projektets början och genom hela utvecklingen.
- *Holistisk design – alla aspekter som påverkar den framtida användningssituationen skall utvecklas parallellt.* Alla delar i systemet skall utvecklas parallellt, kontinuerligt och beroende av varandra: användargränssnitt och interaktion; on-line hjälp, handböcker, utbildning, arbetsmiljöaspekter, etc. genom hela livscykeln.
- *Lokalanpassade processer – den användarcentrerade designprocessen skall specificeras, anpassas och/eller införas lokalt i varje organisation.* Att införa och bedriva användarcentrerad systemdesign är komplicerat. Det krävs att organisationen själv tar ansvar för hur detta görs för att möta organisationens eller rent av de enskilda projektens behov.
- *En användarcentrerad attityd skall alltid etableras.* En hög ”lägsta nivå” av användbarhetsmognad är nödvändig bland organisationens/projektets/utvecklingsteamets medlemmar. Alla utvecklingsprojektets medlemmar måste träffa verkliga eller potentiella användare.

Genom att tydligt definiera ACSD och specificera principer kan vi undvika att begreppet ”missbrukas”. Vi kan dessutom vägleda de utvecklare som vill arbeta användarcentrerat i hur de skall genomföra detta i processen. *Processen* vägleder utvecklarna i vad de skall göra, hur och när, samt vilket resultat som kan förväntas. I avhandlingen presenteras förslag på en sådan användarcentrerad process – *användbarhetsdesign*. Den kan användas inom ramen för en kommersiell utvecklingsprocess, såsom Rational Unified Process (RUP), eller inom en egen utvecklingsprocess. Vidare för vi fram och diskuterar en utvecklarroll med speciellt fokus på användbarhet och användarcentrering – *användbarhetsdesignern*.

Slutligen är *attityden* till användarcentrering avgörande för framgången med att utveckla användbara system. Vi har dock sett att många organisationer har blivit slavar under utvecklingsprocessen, i det att den valda utvecklingsprocessen, blir händelsernas centrum. Det verkar som att många tycker att det är viktigare att följa t.ex. RUP än att reflektera över hur användarcentrerad och effektiv processen är. För att till fullo dra fördel av processen är man angelägen att fullfölja allt vad processen anger, och inte på allvar ifrågasätta vad som är bäst för nuvarande projekt i sitt sammanhang.

References

ACM Special Interest Group on Computer-Human Interaction (SIGCHI) Curriculum Development Group (1992). Technical Report, ACM, New York.

Allard, R., Lind, M., Sandblad, B. & Schneider, W. (1984). Sjukvårdsarbetet och datorn (The work in medical care and the computer). Available only in Swedish, *Arbetslivscentrum, forskningsrapport 45*, ISSN 0348-7210 rapport.

Bannon, L. J. (1991). From Human Factors to Human Actors: The Role of Psychology and Human-Computer Interacting Studies in Systems Design. In: Greenbaum J. & Kyng M. (eds.), *Design at work: Cooperative Design of Computer Systems*, pp. 25–44, Hillsdale: Lawrence Erlbaum Associates.

Benyon, D. R., Turner, P. & Turner, S. (2004). *Designing Interactive Systems*, Addison-Wesley.

Bevan, N. (1991). Standards relevant to European Directives for display terminals. In: Bullinger HJ (ed.): *Proceedings of the 4th International Conference on Human Computer Interaction*, Stuttgart, September 1991, Elsevier.

Beyer, H. & Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centred Systems*. Morgan Kaufman, San Francisco.

Bjerknes, G., Ehn, P. & Kyng, M. (eds.) (1987). *Computers and Democracy: A Scandinavian Challenge*. Aldershot, UK, Avebury.

Bourner, T. & O'Hara, S. (1999). Practitioner Centred Research. Paper presented at: *The SEDA/SRHE conference: Research and Practice in Educational Development(s): exploring the links*, 14-16 April 1999. Retrieved January 16, 2001 from http://www.lmu.ac.uk/lss/staffsup/confreps/bourne~.htm. Bourner and O'Hara are at the University of Brighton, England, http://www.brighton.ac.uk/.

Bynum, T.W. & Rogerson, S. (eds.) (2003). *Computer ethics and professional responsibility*. Blackwell Publishing, ISBN 1-85554-845-3.

Card, S.K., Moran, T.P. & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates Ltd., Publishers, Hillsdale, New Jersey.

Catterall, B.J., Taylor, B.C. & Galer, M.D. (1991). The hufit planning, analysis and specification toolset: Human factors as a normal part of the it product design processing. In: *Taking Software Design Seriously* (Karat J., ed.), London, Academic Press.

Constantine, L.L. & Lockwood, L.A.D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design*. Boston, Addison-Wesley.

Cooper, A. (1999). *The Inmates are Running the Asylum: Why High-tech Products Drive us Crazy and How to Restore the Sanity*, SAMS, Indianapolis, Indiana, ISBN-0-672-31649-8.

Creative Good (2000). *The dotcom survival guide.* Retrieved October 10, 2001 from http://www.creativegood.com.

Dagens Medicin (04/01). *Datorisering går i snigelfart.* Available only in Swedish. Also accessible on-line at http://www.dagensmedicin.se/.

Dick, B. (1993). *You want to do an action research thesis?* Accessed on-line March 17, 2004 at http://www.scu.edu.au/schools/gcm/ar/art/arthesis.html.

Dix, A., Finlay, J., Abowd, G. & Beale, R. (1998). *Human-Computer Interaction.* Second edition, Prentice Hall Europe.

Donahue, G. M. (2001). Usability and the Bottom line. *IEEE Software, Vol 18, No 1,* pp. 31-37, available at http://www.ieee.org.

Dray, S. & Siegel, D. (1998). User-Centred Design and the "Vision Thing", in the ACM/SIGCHI magazine: *interactions march+april 1998.*

Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts.* Arbetslivscentrum, Stockholm, Sweden.

Ehn, P. & Löwgren, J. (2003). Introduction. In: Ehn P. & Löwgren J. (Eds.) *Searching Voices. Towards a canon for interaction design, Studies in Arts and Communication #01,* published by the School of Arts and Communication, Malmö University, Sweden, ISBN 91–7104–010–2.

Fallman, D. (2003). Design-oriented Human-Computer Interaction. In: *The Proceedings of CHI,* Ft Lauderdale, the US, April 5-10, 2003. Vol 5, Issue 1. pp 225-232.

Fowler, M. (1997). UML Distilled. *Applying the Standard Object Modeling Language.* Addison Wesley Longman Inc., Reading, Mass., U.S.

Goldkuhl, G. (1994). *Välgrundad metodutveckling* (Well-grounded methods development). Technical report in Swedish. LiTH-IDA-R-94-04, Department of Computer and Information Science, Linköping University, Sweden.

Gould, J.D. & Lewis, C. (1983). Designing for usability. In: *The proceedings of the CHI'83* conference Boston, Mass, Human factors in computing systems edited by Ann Janda, Amsterdam North-Holland.

Gould, J.D. & Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM, Vol. 28, No. 3,* pp. 300-311.

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.

Gould, J.D., Boies, S.J. & Ukelson, J. (1997). How to Design Usable Systems. In: Helander, Landauer & Prabhu (eds.), *Handbook of Human-Computer Interaction,* Elsevier Science B.V.

Greenbaum, J. & Kyng, M. (eds.) (1991). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, NJ, Lawrence Erlbaum Associates.

Gulliksen, J. & Göransson, B. (2002). *Användarcentrerad systemdesign – en process med fokus på användare och användbarhet* (User-Centred Systems Design – A process with focus on users and usability, textbook only in Swedish). Studentlitteratur, Lund, ISBN 91-44-02029-5.

Göransson, B. (2001). Usability Design: A Framework for Designing Usable Interactive Systems in Practice. *Licentiate thesis in Human-Computer Interaction, Uppsala University*, June 2001. Available at: http://www.it.uu.se/research/reports/lic/2001-006/.

Göransson, B. & Sandbäck, T. (1999). Usability designers improve the user-centred design process. In: *The proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction: INTERACT99*, Edinburgh, UK, vol. II, pp. 163–167.

Helander, M. (ed.) (1988). *Handbook of Human-Computer Interaction*. North-Holland, Amsterdam.

Helander, M., Landauer, T. & Prabhu, P. (eds.) (1997). *Handbook of Human-Computer Interaction*. Second completely revised edition, Elsevier Science B.V.

ISO/IEC 9126 (1991). *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*, First edition 1991-12-15, ref. number ISO/IEC 9126:1991(E), International Organization for Standardization, Geneve.

ISO/IS 9241-10 (1996). *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 10: Dialogue principles*. First edition 1996-05-01, ref. number ISO 9241–10:1996(E), International Organization for Standardization, Geneve.

ISO/IS 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.

ISO/IS 13407 (1999). *Human-centred design processes for interactive systems*. International Organization for Standardization, Geneve.

ISO/TR 18529 (2000). *Ergonomics—Ergonomics of human-system inter-action—Human-centred lifecycle process description*. First edition 2000-06-15, ref. number ISO/TR 18529:2000(E), International Organization for Standardization, Geneve.

Johnson-Laird, P.N. (1983). *Mental models.* Cambridge Univeristy Press, ISBN 0 521 24123 5.

Karat, C. (1993). Usability Engineering in Dollars and Cents, *IEEE Software, May 1993*, pp 88–89.

Karat J. (1996). User Centred Design: Quality or Quackery? In: The ACM/SIGCHI magazine, *interactions july+august 1996*.

Karat, J. (1997). Evolving the Scope of User-Centred Design. In: *Communications of the ACM, Vol. 40, No 7*, pp. 33-38.

Karat, J., Atwood, M., Dray, S., Rantzer, M. & Wixon, D. (1996). User Centred Design: Quality or Quackery? Panel session in: *The proceedings of the CHI'96 conference*.

Karat, J. & Dayton, T. (1995). Practical Education for Improving Software Usability. In: *The proceedings of the CHI'95 conference*.

Karat, J. & Karat, C. M. (2003). The evolution of user-centred focus in the human-computer interaction field. In: *IBM Systems Journal, Vol. 42, No. 4*, 2003.

Katzeff, C. & Svärd, P.O. (1995). Användbarhet i Praktiken: En enkätstudie (Usability in Practice: A Survey). *SISU publication no. 95:20*, Stockholm, Sweden: Swedish Institute for Systems Development, available only in Swedish.

Lind, M. (2000). Perspektiv på Användarcentrerad Systemutveckling (Perspective on User-Centred Systems Development). *Compendium only in Swedish. Department of Information Science, Uppsala University, Sweden*. Copyright © Mats Lind.

Long, J. & Dowell, J. (1989). Conceptions of the discipline of HCI: craft, applied science and engineering. In: Sutcliffe, A. & Macauley, L. (Eds.) *People & Computers V*. Cambridge University Press, pp. 9–32.

Marcus, A. (2002). *Return on Investment for Usable User-Interface Design: Examples and Statistics*. Version: 28 February 2002. Retrieved February 2, 2004 from http://www.amanda.com/resources/ROI/AMA_ROIWhitePaper_28Feb02.pdf.

Masters, J. (1995). The history of Action Research. In: I. Hughes (ed.) *Action Research Electronic Reader*, The University of Sydney. Accessed on-line, March 17, 2004 at http://www.scu.edu.au/schools/gcm/ar/arr/arow/default.html.

Mayhew, D. (1999). *The usability engineering lifecycle, a practitioner's handbook for user interface design,* Morgan Kaufmann Publishers Inc., San Francisco, CA.

Mayhew, D. (2002). Promoting, Establishing and Institutionalizing Usability Engineering. *Tutorial at the CHI 2002 conference*, Pittsburgh, USA.

Nielsen, J. (1993). *Usability Engineering*, Academic Press Inc., San Diego.

Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Indianapolis, ISBN 1-56205-810-X.

Norman, D.A. (1986). Cognitive Engineering. In: D. A. Norman & S. W. Draper (eds.), *User Centred System Design*, Lawrence Erlbaum Associates Inc., Hillsdale, New Jersey.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T. (1994). *Human-Computer Interaction*. Addison Wesley Publishing Company, Wokingham, England.

Preece, J., Rogers, Y. & Sharp, H. (2002). *Interaction Design. Beyond Human-Computer Interaction*. Wiley & Sons Inc., ISBN 0-471-40249-4.

PRUE (Providing Reports of Usability Evaluation) (2002). *Guide to specifying and evaluating usability as part of a contract*. Final Report, March 15, 2002. Accessed January 15, 2004 at http://www.usability.serco.com/prue/project.htm.

Rational Software Corporation (2003). *Rational Unified Process.* On-line version 2003.06.00.65.

Rogers, E.M. & Scott, K.L. (1997). *The Diffusion of Innovations Model and Outreach from the National Network of Libraries of Medicine to Native American Communities.* Department of Communication and Journalism University of New Mexico, Albuquerque, USA. Retrieved January 27, 2004 from http://nnlm.gov/pnr/eval/rogers.html.

Schuler, D. & Namioka, A. (1993). *Participatory design: Principles and practices.* Erlbaum, Hillsdale, N.J.

Schön, D. (1983). *The Reflective Practitioner.* New York: Basic Books.

Schön, D. (1987). *Educating the reflective practitioner: Toward a new design for teaching and learning in the professions.* San Francisco: Jossey-Bass.

Seffah, A. (2003). Human-Centered Design Skills and Patterns for Software Engineers Education. In: *ACM Interactions, vol. X.5*, september + october 2003, pp. 36–45.

Smith, H. J. & Keil, M. (2003). The reluctance to report bad news on troubled software projects: a theoretical model. *Information Systems Journal 13 (1)*, pp. 69-95.

Standish Group (1995). *The CHAOS report*. Available at http://www.scs.carleton.ca/~beau/PM/Standish-Report.html.

Suchman, L.A. (1987). *Plans and situated actions: The problem of human-machine communications.* Cambridge, UK: Cambridge University Press.

Swedish Agency for Public Management (2000). *The 24/7 Agency. Statskontoret 2000:41*, available at http://www.statskontoret.se/pdf/200041.pdf

Swedish Agency for Public Management (2002). *24-timmarswebben – Rekommendationer och råd för 24-timmarsmyndighetens webbplats.* Statskontoret 2002:13, only in Swedish, available at http://www.statskontoret.se/pdf/200213.pdf.

Swedish Work Environment Authority (1998). *Arbetarskyddsstyrelsens föreskrifter om arbete vid bildskärm*. AFS 1998:5, 10§, only in Swedish, available at http://www.av.se/regler/afs/1998_05.pdf.

Swedish Work Environment Authority (2001). *The Work Environment Act*. Available at http://www.av.se/english/legislation/legislation.shtm.

UI Engineering (2001*). Are the Product Lists on Your Site Reducing Sales?* Retrieved January 27, 2004 from http://www.uiconf.com/whitepapers/WhitePaperPogoSticking.pdf.

U.S. Department Of Health and Human Services, Food and Drug Administration (2002). *General Principles of Software Validation; Final Guidance for Industry and FDA Staff*. Document issued on: January 11, 2002, accessed January 27, 2004 at http://www.fda.gov/cdrh/comp/guidance/938.html.

UsersAward (2002). *IT-kartan – Användare och IT-system i svenskt näringsliv* (The IT-map – Users and IT-systems in Swedish industry). Only in Swedish. Can be ordered at http://www.usersaward.se.

Vredenburg, K., Mao, Ji-Ye, Smith, P. & Carey, T. (2002). A Survey of User-Centred Design Practice. In: *Proceedings of the CHI 2002 Conference*, April 20–25, 2002, Minneapolis, Minnesota, USA.

Wallace, M.D. & Anderson, T.J. (1993). Approaches to Interface Design. *Interacting with Computers, Vol. 5, No. 3*, pp. 259-278.

Webb, J. (1995). *Extending the Theory-Practice Spiral: Action Research as Mechanism for crossing the Academic/Professional Divide*. First published in: Web Journal of Current Legal Issues in association with Blackstone Press Ltd. Accessed March 17, 2004 at http://webjcli.ncl.ac.uk/articles2/webb2.html.

Winograd, T., Bennett, J., De Young, L. & Hartfield, B. (eds.) (1996). *Bringing Design to Software*. Addison-Wesley.

Wixon, D. & Holtzblatt, K. (1990). Contextual Design: An Emergent View of System Design. In: *The proceedings of the CHI'90 conference*.

Åborg, C. (2002). How Does IT Feel @ Work—And How to Make IT Better. *Ph.D. thesis, Uppsala University, Sweden*.

Acta Universitatis Upsaliensis

Comprehensive Summaries of Uppsala Dissertations
from the Faculty of Science and Technology

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology*. (Prior to October, 1993, the series was published under the title "Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science".)

Distribution:
Uppsala University Library
Box 510, SE-751 20 Uppsala, Sweden
www.uu.se, acta@ub.uu.se

ISSN 1104-232X
ISBN 91-554-5975-7