# Key principles for user-centred systems design

JAN GULLIKSEN, BENGT GÖRANSSON, INGER BOIVIE, STEFAN BLOMKVIST, JENNY PERSSON and ÅSA CAJANDER

Uppsala University, Department of Information Technology, Human-Computer Interaction, PO Box 337, S-751 05 Uppsala, Sweden; e-mail: jan.gulliksen@it.uu.se

**Abstract.** The concept of user-centred systems design (UCSD) has no agreed upon definition. Consequently, there is a great variety in the ways it is applied, which may lead to poor quality and poor usability in the resulting systems, as well as misconceptions about the effectiveness of UCSD. The purpose of this paper is to propose a definition of UCSD. We have identified 12 key principles for the adoption of a user-centred development process, principles that are based on existing theory, as well as research in and experiences from a large number of software development projects. The initial set of principles were applied and evaluated in a case study and modified accordingly. These principles can be used to communicate the nature of UCSD, evaluate a development process or develop systems development processes that support a user-centred approach. We also suggest activity lists and some tools for applying UCSD.

## 1. Purpose and justification

This paper describes the results of our current research on UCSD and our experiences of applying UCSD in software development projects. Our purpose has been to compile knowledge and experiences of UCSD, in order to give the concept a more precise meaning and to increase its power. The main point in our paper is that applying UCSD requires a profound shift of attitudes in systems development, and our main goal is to promote that attitude shift.

## 2. Background

Our main concern has been the lack of an agreed upon definition of UCSD, turning it into a concept with no real meaning. UCSD was originally coined by Norman and Draper (1986). They emphasized the importance of having a good understanding of the users (but without necessarily involving them actively in the process):

> '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)

Several other definitions and understandings have been proposed over the years. The lack of a shared understanding of the meaning of UCSD (or User-Centred Design, UCD) has actually been pointed out as a quality in its own right by Karat:

> '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)

> '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)

The consequence of such general and non-specific definitions of user-centred design is that it, in practice, becomes a concept with no real meaning.

We have therefore identified a set of key principles[1] for UCSD. The principles summarize our research

*Behaviour & Information Technology*
ISSN 0144-929X print/ISSN 1362-3001 online © 2003 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/01449290310001624329

results and experiences from software development projects in a large number of organizations and projects. They are based on principles specified elsewhere (Gould *et al*. 1997, ISO 13407, 1999) and on our experiences made from trying to apply UCSD in systems development projects using processes such as the Rational Unified Process (Kruchten 1998). Our principles also take into account the Scandinavian tradition of extensive user involvement in the development process (Greenbaum and Kyng 1991) in some communities known as participatory design. Other well-known approaches such as contextual design (Beyer and Holzblatt 1998), goal-directed design (Cooper 1999), usability engineering (Nielsen 1993, Mayhew 1999) have also contributed to the result.

Below we describe one of the projects that had particular impact on the principles in that it was conducted with the explicit goal to capture critical success factors for UCSD.

## 3. The project

The pilot project was an in-house development project within the Swedish National Tax Board with the purpose to develop a new computerized case-handling tool for administrators working with national registration. We were able to follow the project from the very start. In the first project meeting we emphasized the importance of following a UCSD approach and introduced our set of principles to the project team.

These principles were specific for the organization and had been identified in an earlier research effort (Gulliksen and Göransson 2001). They were:

- *The work practices of the users control the development*. Early focus on users and tasks. The designer must understand the users, their cognitive behaviour, attitudes and the characteristics of their work tasks. Appropriate allocation of function between the user and the system is also important to prevent unnecessary control;
- *Active user participation* throughout the project, in analysis, design, development and evaluation. This requires a careful user selection process emphasizing the skills of typical users, including both:
    - work domain experts (continuously through the development project);
    - and actual end-users (for interviews and observations as well as evaluation of design results).
- *Early prototyping* to evaluate and develop design solutions and to gradually build a shared understanding of the needs of the users as well as their future work practices;
- *Continuous iteration* of design solutions. A cyclic process of design, evaluation and redesign should be repeated as often as necessary. The evaluation process should include empirical measurement in which tests are conducted where users perform real tasks on prototypes. The users' reactions and attitudes should be observed and analysed;
- *Multidisciplinary design teams*. Mainly achieved by including a usability designer (Göransson and Sandbäck 1999) in the process;
- *Integrated design*. The system, the work practices, on-line help, training, organization, etc. should be developed in parallel.

The project decided to act in accordance with the above principles.

### 3.1. *Research methods*

We used an action research approach in the project, i.e. our aim was to introduce changes in the development process as regards user involvement and usability issues, and to observe and record the outcomes of these changes. Our activities included introducing a set of UCSD principles as described above, and facilitating the project team's commitment to these principles. We also facilitated collaborative prototyping activities with users.

To observe the outcomes of the activities and actions, we used qualitative data collection methods as described below.

- observations of the work of the development team, for instance, by continuously participating in the project meetings of the software development team;
- observations of the current work practices (mainly paper-based) of the administrators working with national registration;
- semi-structured interviews based on open-ended questions with software developers and user representatives about their attitudes to and experiences with working with users and usability;
- semi-structured interviews based on open-ended questions with users about their work;
- continuous discussions with members of the software development team and representatives for the current work practices to check possible discrepancies in our interpretation of the observed activities and actions.

Meanwhile, we continued working with the principles. As a result of intermediate findings in the pilot project and findings in other, parallel, research efforts we modified the set of principles to cover the 12 key principles described in this paper. The applicability of these principles was then assessed in a number of workshops with researchers and practitioners.

### 3.2. *Results*

As a result of the introductory meeting, the project group decided to apply UCSD as defined by the initial set of principles.

We could not influence the choice and customization of the development process – the organization had recently shifted to using the Rational Unified Process (RUP) (Kruchten 1998). We were, however, able to introduce additional activities to complement the process as needed, e.g. activities for performing a thorough user and task analysis, for developing design solutions iteratively and in cooperation with the users, and for including a usability designer throughout the project.

One of the more successful events was a collaborative prototyping activity in which the users could develop their vision of the future system and work situation, integrating a future system and future work practices (figure 1).

These collaborative prototyping sessions were facilitated by a usability designer in cooperation with a researcher. The users brought sketches illustrating their own view of the future system as a basis for a negotiation on the most appropriate design of the system.

Low-level prototyping tools were used since the users regarded them as the most flexible tool for their purpose (figure 2).

Prior to the collaborative design sessions the usability designer had conducted a user analysis and created personas. According to Calde *et al*. (2002) user models, or personas, are fictional, detailed archetypical characters that represent distinct groupings of behaviours, goals and motivations observed and identified during the research phase. Cooper (1999) describes personas as a tool for communication and design within the group of designers, software developers, managers, customers and other stakeholders. The purpose is not to give a precise description or a complete theoretical model of a user. Instead, it is aiming at a simple, but good enough description of the user to make it possible to design the system (figure 3).

From the software engineering side they had been performing use case modelling to specify the detailed requirements on the system. A use case specifies the sequence of actions, including alternatives of the sequence, that the system can perform, interacting with actors of the system (Jacobsson *et al*. 1999). Use case modelling is today one of the most widely used software engineering techniques to specify user requirements. Unified Modelling Language (UML) is one of the most common formal notations to describe use cases (Fowler 1997). Rational Unified Process (RUP) (Kruchten 1998) builds heavily upon these techniques.

According to the users, the personas gave a much more concrete picture of typical users than what came out of the use case modelling sessions running in parallel with the collaborative prototyping activities.

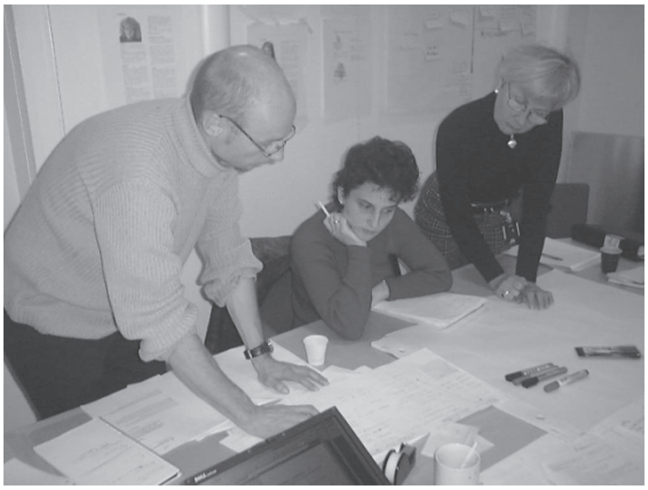


Figure 1. Collaborative prototyping in which the usability designer facilitates the users' production of mock-ups.

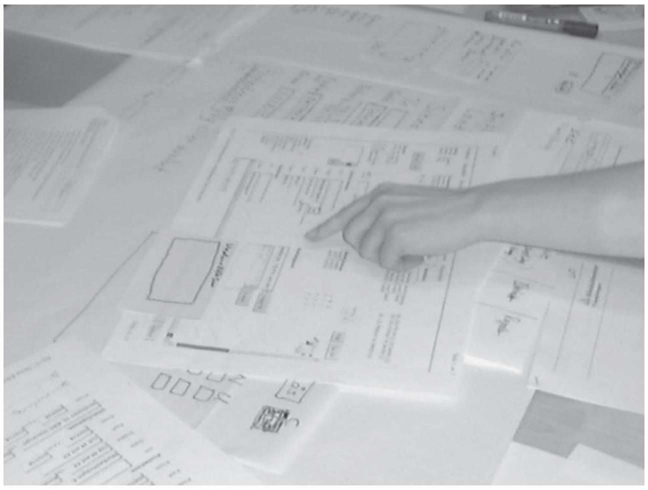
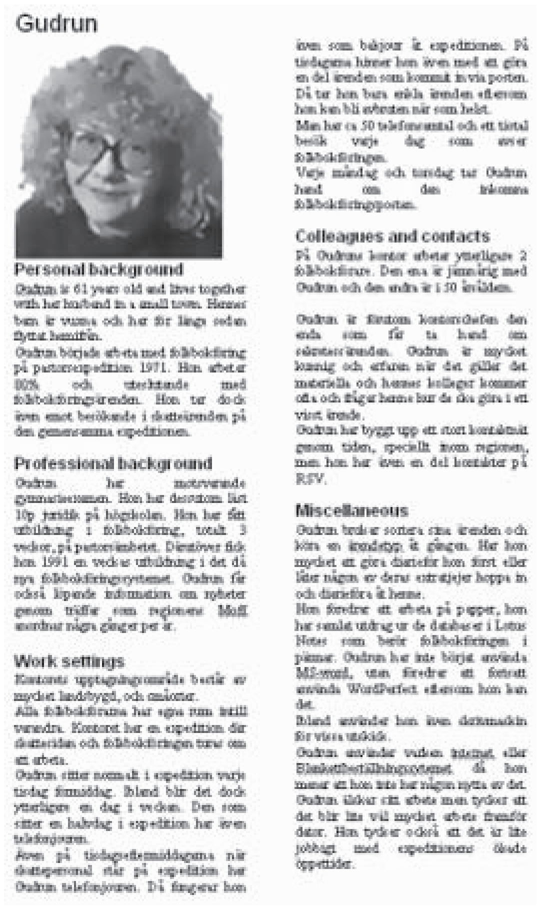


Figure 2. Low-fidelity prototyping tools were used as these were the most convenient for visualizing the future use situation without limiting the design space.



Gudrun

Personal background

Gudrun är 61 years old and lives together with her husband in a small town. Hennes barn är vuxna och har för länge sedan flyttat hemifrån.

Gudrun började arbeta med folkbokföring på pastorsexpedition 1971. Hon arbetar 80% och uteslutande med folkbokföringsärenden. Hon tar dock även emot besökande i dödsärenden på den gemensamma expeditionen.

Professional background

Gudrun har motsvarande gymnasieexamen. Hon har dessutom läst 10p juridik på högskolan. Hon har fått utbildning i folkbokföring, totalt 3 veckor, på pastorsämbetet. Därutöver fick hon 1991 en veckas utbildning i det då nya folkbokföringssystemet. Gudrun får också löpande information om nyheter genom träffar som regionens Moff anordnar några gånger per år.

Work settings

Kontorets upptagningsområde består av mycket landsbygd, och småorter.

Alla folkbokförarna har egna rum intill varandra. Kontoret har en expedition där dödsärenden och folkbokföringen turas om att arbeta.

Gudrun sitter normalt i expedition varje tisdag förmiddag. Ibland blir det dock ytterligare en dag i veckan. Den som sitter en halvdag i expedition har även telefonjouren.

Även på tisdagseftermiddagarna när dödsfallspersonal står på expedition har Gudrun telefonjouren. Då fungerar hon även som bakjour åt expeditionen. På tisdagarna hinner hon även med att göra en del ärenden som kommit in via posten. Då tar hon bara enkla ärenden eftersom hon kan bli avbruten när som helst.

Man har ca 50 telefonsamtal och ett tiotal besök varje dag som avser folkbokföringen.

Varje måndag och torsdag tar Gudrun hand om den inkomna folkbokföringsposten.

Colleagues and contacts

På Gudruns kontor arbetar ytterligare 2 folkbokförare. Den ena är jämnårig med Gudrun och den andra är i 30 årsåldern.

Gudrun är förutom kontorschefen den enda som får ta hand om sekretessärenden. Gudrun är mycket kunnig och erfaren när det gäller det materiella och hennes kollegor kommer ofta och frågar henne hur de ska göra i ett visst ärende.

Gudrun har byggt upp ett stort kontaktnät genom tiden, speciellt inom regionen, men hon har även en del kontakter på RSV.

Miscellaneous

Gudrun brukar sortera sina ärenden och köra en ärendetyp åt gången. Har hon mycket att göra diarieför hon först eller låter någon av deras extratjejer hoppa in och diarieföra åt henne.

Hon föredrar att arbeta på papper, hon har samlat utdrag ur de databaser i Lotus Notes som berör folkbokföringen i pärmar. Gudrun har inte börjat använda MS-word, utan föredrar att fortsätt använda WordPerfect eftersom hon kan det.

Ibland använder hon även skrivmaskin för vissa utskick.

Gudrun använder varken Internet eller Blankettbeställningssystemet då hon menar att hon inte har någon nytta av det.

Gudrun tycker om sitt arbete men tycker att det blir lite väl mycket arbete framför datorn. Hon tycker också att det är lite jobbigt med expeditionens ökade öppettider.

Figure 3. Personas were used to describe typical users. In this example, the persona 'Gudrun' is described based on personal background, the work setting, colleagues and contacts.

Halfway through the project all participants were very satisfied with the activities so far and the results achieved. The project was committed on all levels to UCSD. The principles communicated the essentials of UCSD very well.

From then on, however, there was a gradual increase of problems and obstacles to the user-centred approach. Despite efforts from our side and from the project, the problems were never really resolved. Some of them were outside the control of the project.

The major problems in the project are briefly described below. The problems reflect why the initial principles were not sufficient, and therefore each of the problems is related to the subsequent definition and 12 key principles of User Centred Systems Design. The outcome of the project can be compared with the consolidated list of 12 key principles, and each problem in the project maps well against one or more of the principles.

- *No lifecycle perspective on UCSD.* The developers focused on short-term goals, such as, producing models and specifications prescribed by RUP. The long-term goals and needs of the users regarding their future work situation were ignored or forgotten. Moreover, towards the end of the project, meeting the project goals and deadlines became much more important than achieving some sort of minimum level of usability. We believe, that had the project decided to give the usability activities higher priority than, for example, to develop absolutely all the functionality the end result could have been a lot better, without any of the missing functionality causing any big problems in the long run. We emphasize the importance of a lifecycle perspective in our definition of UCSD in the next chapter as well as in a number of the principles, for instance, the user involvement principle and the usability champion principle. The lack of lifecycle perspective also indicates that there was no real commitment to UCSD in the project which points to an attitude problem;
- *Usability designers were ignored.* Despite the skilful and experienced work that the usability designers performed, their results and their opinions were ignored in the later phases of the project. The usability champion principle points out that the usability champion/designer[2] should have the mandate to decide on usability matters. The project ignoring the input of the usability designer clearly indicates that this was not the case;
- *Use case mania.* When the project started, the organization did not have enough experience with use case modelling. The modelling went out of hand and the results could not be used efficiently in the development process. The project got literally bogged down in use cases, but did not really know what to do with them. The use case mania indicates that there was a problem with user focus in the project. Despite the confusion regarding the use of the use cases, producing them became more important than understanding the users' real needs;
- *Poor understanding of the design documentation.* The design was documented in UML and the users were invited to evaluate it. The users had severe difficulties predicting their future use situation based on the UML notation. One of the users said that after having worked with use case modelling, the collaborative prototyping was like 'coming out of a long dark tunnel'. The design representation principle emphasizes the impor-

tance of using representations that are easy to understand for all the stakeholders, in particular as regards the future work/use situation. UML is clearly not suitable in that respect;

- *Major changes in the project*. Halfway through the project a strategic decision was made within the organization, against our advice, to change the technical platform and continue the development in a web-based environment. The decision was crucial in that it made it very difficult to meet the usability requirements. Insufficient experience with and expertise in the new technology as well as the page metaphor in html created problems. The decision was made with little or no attention to usability matters. This indicates that there was a problem with the attitudes to UCSD and usability within the organization and a problem with user focus;
- *Problems establishing a user centred attitude*. Single individuals in a project can make a crucial difference when it comes to UCSD. We noticed, for instance, problems with resolving conflicts between personal goals and business goals within the project, on an individual level. Again, this indicates that there was a problem with attitudes and user focus in the organization. It also indicates problems with the professional attitude described in the principle on multidisciplinary design.

This case describes how a project with explicit intentions to apply UCSD, nevertheless ran into several problems and obstacles that made it very difficult to pursue the UCSD approach. Our conclusion is that one needs to be very specific about what it takes from the process to comply with UCSD to prevent problems such as the ones described in the pilot study.

Based on the results of the project, we concluded that the principles listed in Gould *et al*. (1997) and ISO 13407 (1999) are not sufficient to maintain a UCSD approach in a project or in an organization. We therefore modified our initial set of principles to clearly indicate that it takes much more to work in a user-centred fashion. We have also run a number of workshops with researchers and practitioners to discuss and confirm the principles. The resulting set is listed below together with a definition of UCSD.

## 4. Definition and key principles

User-centred system design (UCSD) is a process focusing on usability[3] throughout the entire development process and further throughout the system life cycle (figure 4). 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* (Gould *et al*. 1997, ISO 13407 1999). All members of a project must understand the goals of the activity, the context of use, who the users are, their situation, goals and tasks, why and how they perform their tasks, how they communicate, cooperate and interact, etc. This helps in creating and maintaining a focus on the users' needs instead of a technical focus. Activities, such as identifying user profiles, contextual inquiries and task analysis, must be a natural part of the development process. Make sure that all project members have met real or potential users, for instance, by visiting the workplace. Descriptions of typical users, tasks and scenarios could be put up on the walls of the project room/area to maintain a user focus;
- *Active user involvement – representative users should actively participate, early and continuously throughout the entire development process and throughout the system lifecycle* (Nielsen 1993, Gould *et al*. 1997, ISO 13407 1999). The users should be directly involved, both in the development project and in related activities, such as, organizational development and designing new work practices (Greenbaum and Kyng 1991). The users must be representative of the intended user groups. Plans for involving users should be specified from the very start of the project.

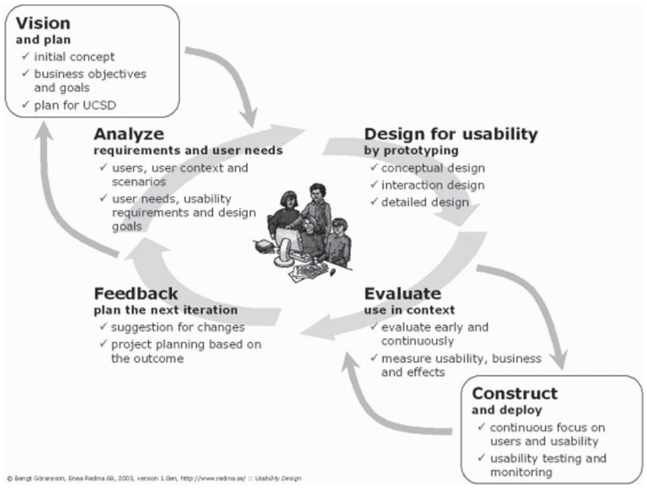

Vision
and plan
✓ initial concept
✓ business objectives and goals
✓ plan for UCSD
Analyze
requirements and user needs
✓ users, user context and scenarios
✓ user needs, usability requirements and design goals
Design for usability
by prototyping
✓ conceptual design
✓ interaction design
✓ detailed design
Feedback
plan the next iteration
✓ suggestion for changes
✓ project planning based on the outcome
Evaluate
use in context
✓ evaluate early and continuously
✓ measure usability, business and effects
Construct
and deploy
✓ continuous focus on users and usability
✓ usability testing and monitoring
© Bengt Göransson, Enea Redina AB, 2003, version 1.0en, http://www.redina.se/ :: Usability Design

Figure 4. User-centred system design (UCSD) is a process focusing on usability throughout the entire development process and further throughout the system life cycle.

Identify appropriate phases for user participation and specify where, when and how users should participate.[4] Emphasize the importance of meeting the users in context, for instance, at their workplace;

- *Evolutionary systems development – the systems development should be both iterative and incremental* (Boehm 1988, Gould *et al.* 1997). It is impossible to know exactly what to build from the outset. Hence, UCSD requires an approach which allows continuous iterations with users and incremental deliveries. This, so that design solutions can be evaluated by the users before they are made permanent. An iteration should contain a proper analysis of the users' needs and the context of use, a design phase, a documented evaluation with concrete suggestions for modifications and a redesign in accordance with the results of the evaluation. These activities do not have to be formal. An iteration could be as short as half an hour, as long as it contains all three steps. Incremental development means that, based on an overall picture of the system under development (SUD), priorities are set and the system is divided into parts that can be delivered for real use. Each increment is iterated as described above. Evaluations of the increments in real use should influence the design of the subsequent increments. Let the software grow into the final product;
- *Simple design representations – the design must be represented in such ways that it can be easily understood by users and all other stakeholders* (Kyng 1995). Use design representations and terminology that are easily understood by all users and stakeholders so that they can fully appreciate the consequences of the design on their future use situation. Use, for instance, prototypes (sketches and mock-ups) and simulations. Abstract notations, such as use cases, UML diagrams or requirements specifications are not sufficient to give the users and stakeholders a concrete understanding of the future use situation (Mathiassen and Munk-Madsen 1986, Bødker 1998). The representations must also be usable and effective. The goal is that all parties involved share an understanding of what is being built;
- *Prototyping – early and continuously, prototypes should be used to visualize and evaluate ideas and design solutions in cooperation with the end users* (Nielsen 1993, Gould *et al.* 1997). Use multiple paper sketches, mock-ups and prototypes to support the creative process, elicit requirements and visualize ideas and solutions. The prototypes should be designed and evaluated with real users in context (contextual prototyping). It is essential to start with low-fidelity materials, for instance, quick sketches, before implementing anything in code. Start with the conceptual design on a high level and do not move on to detail too quickly. If possible produce several prototypes in parallel, since this helps the designers in maintaining an openness and creative attitude to what is being built. Far too often the design space is unnecessarily limited by only sticking with the first set of designs produced;
- *Evaluate use in context – baselined usability goals and design criteria should control the development* (Nielsen 1993, Gould *et al.* 1997). Critical usability goals should be specified and the design should be based on specific design criteria. Evaluate the design against the goals and criteria in cooperation with the users, in context. Early in the development project, one should observe and analyse the users' reactions to paper sketches and mock-ups. Later in the project, users should perform real tasks with simulations or prototypes. Their behaviour, reactions, opinions and ideas should be observed, recorded and analysed. Specify goals for aspects that are crucial for the usability and that cover critical activities as well as the overall use situation;
- *Explicit and conscious design activities – the development process should contain dedicated design activities* (Cooper 1999). The user interface design and the interaction design are of undisputed importance for the success of the system. Remember that to the users the user interface is the system. The design of the SUD as regards the user interaction and usability should be the result of dedicated and conscious design activities. The construction of the SUD should adhere to that design. Far too often, the UI and interaction design 'happens' as a result of somebody doing a bit of coding or modelling rather than being the result of professional interaction design as a structured and prioritized activity;
- *A professional attitude – the development process should be performed by effective multidisciplinary teams* (ISO 13407 1999). Different aspects and parts of the system design and development process require different sets of skills and expertise. The analysis, design and development work should be performed by empowered multidisciplinary teams of, for instance, system architects, programmers, usability designers, interaction designers and users. A professional attitude is required and so are tools that facilitate the cooperation and efficiency of the team;

- *Usability champion – usability experts should be involved early and continuously throughout the development lifecycle* (Kapor 1990). There should be an experienced usability expert (usability designer) or possibly a usability group on the development team. The usability designer should be devoted to the project as an 'engine' for the UCSD process from the beginning of the project and throughout the development process and system lifecycle (Buur and Bødker 2000). The usability designer must be given the authority to decide on matters affecting the usability of the system and the future use situation;
- *Holistic design – all aspects that influence the future use situation should be developed in parallel* (Gould *et al*. 1997). Software does not exist in isolation from other parts of, for instance, a work situation. When developing software for the support of work activities, the work organization, work practices, roles, etc, must be modified. All aspects should be developed in parallel. This includes work/task practices and work/task organization, user interface and interaction; on-line help; manuals; user training, work environment, health and safety aspects, etc. Other parts of the context of use such as: hardware, and social and physical environments, must also be considered in the integrated design process. One person or team should have the overall responsibility for the integration of all aspects;
- *Processes customization – the UCSD process must be specified, adapted and/or implemented locally in each organization*. Usability cannot be achieved without a user-centred process. There is, however, no one-size-fits-all process. Thus the actual contents of the UCSD process, the methods used, the order of activities, etc, must be customized and adapted to the particular organization and project based on their particular needs. A UCSD process can be based on a commercial or in-house software development process, where activities are added, removed or modified. Existing methods and techniques may well be re-used, if they comply with the key principles;
- *A user-centred attitude should always be established*. UCSD requires a user-centred attitude throughout the project team, the development organization and the client organization. All people involved in the project must be aware of and committed to the importance of usability and user involvement, but the degree of knowledge may differ depending on role and project phase (Boivie *et al*. 2003). The key principles defined in this paper can serve as a common ground.

The above 12 principles facilitate the development, communication and assessment of user-centred design processes for creating usable interactive systems, covering analysis, design, evaluation, construction and implementation. Several benefits come with applying the principles, such as their help in maintaining the focus on the users and the usability throughout the entire development process. The UCSD poster is reprinted in Appendix 1.

We fully appreciate that it will be more or less impossible to implement all the principles in one strategic shift. Adopting them gradually is probably more feasible and practicable. It is, however, important to comply with the principles to as high a degree as possible at any point in time.

## 5. Tools for applying UCSD

The principles are, necessarily, general and rather abstract in nature, and cannot be applied as is in practice. We are therefore currently working on activity lists, with potential tools and techniques, for each principle. These lists will provide support for applying the principles and help in understanding and assessing them.

### 5.1. *Activity list*

The purpose of the activity list that accompanies each principle is to elaborate on what it takes to apply a principle. The activity list suggests activities of a general nature alongside appropriate methods, tools and techniques. The principles are general but the activity lists should be developed specifically to fit each organization.

### 5.2. *Complying with the activity list*

The lists suggest activities and it is important to evaluate the applicability of each activity within the current project. If one chooses not to perform a particular activity, it is important to make clear why, and that all parties involved agree with the decision. The activity list serves as both a To-do list and a checklist, where each item can be 'ticked off'. There are three options for each activity:

- No = we decided to not perform this activity. We gave rationales for this decision and had a general agreement on the motives;

- Yes = we performed this activity, in full or to the extent that the project team and management, found appropriate;
- N/A = we found that this activity was not applicable. The rationales for this were clearly stated and agreed on. We have conducted other activities to compensate for this.

Below is a draft activity list for the principle User focus:

### 5.3. *Activity list, tools and methods for the principle; User focus*

- Vision, purpose goal and constraints of the target activity analysed and understood by all project members;
  - *Tools and methods:* Goals analysis, Focus groups;
- Identification, description and prioritization of all user groups;
  - *Tools and methods:* User analysis, personas;
- Visualization and characteristics of target user groups made available to everyone in the project;
  - *Tools and methods:* Decorate a project room with artefacts, etc. that illustrate the users' work situation, environment and characteristics;
- Potential limitations and restrictions in the users' capabilities (for instance vision impairments or language problems) are clear to everyone in the project;
- The development team has focused on the needs of target user groups;
- The users have expressed their impressions of current system and expectations on future system;
  - *Tools and methods:* Users asked about good things and bad things in their current work situation, Think-out loud;
- Users observed as they were performing their tasks in context;
  - *Tools and methods:* Analysis of information utilization, Context-of-use analysis, Field studies, Contextual inquiry;
- Use situation documented;
  - *Tools and methods:* Video and still camera, scenarios, personas;
- Tasks analysed;
  - *Tools and methods:* Task analysis;
- Copies of artefacts (forms, documents archives, notebooks, etc.) used by the users collected.

## 6. Application

In the pilot project described above, an initial set of principles was used to define a UCSD process. The consolidated list of principles was subsequently used to identify mismatches between the development process and a UCSD approach. The definition and principles for UCSD can, however, be used for a number of purposes as listed below:

(a) *Explanation model* – to analyse and communicate why organizations, projects or processes did not meet their goals as regards usability;
(b) *Process development* – for defining a UCSD process;
(c) *Process/Organization customization* – to customize or adapt an organization, project or development process to UCSD, for instance, a commercial development process, such as Rational Unified Process – RUP (Kruchten 1998). Even though RUP prevents rather than promotes UCSD, it may be modified to integrate some of its features (Gulliksen and Göransson 2001);
(d) *Process/Organization assessment* – to assess the user-centeredness of an organization, project or process. Using the principles to identify mismatches, problems may be identified in time to do something about them, which increases the chances of producing a usable piece of software;
(e) *Knowledge transfer* – to teach and transfer knowledge about UCSD and to communicate the basic philosophy of UCSD;
(f) *Procurement support* – support for procurers as a basis for specifying requirements on the design process as such;
(g) In *client-contractor relations* – the client can demand that the contractor work in accordance with the definition and key principles for UCSD. At present, usability is often taken for granted. Clients do not understand that it takes systematic work according to a UCSD philosophy to achieve usability.

Our definition and key principles originate from our experiences and research in contract and in-house development of bespoke software for work situations. We nevertheless see a potential for applying them in other types of development projects. Regardless of the project and the organization, the principles must always be adapted to the context.

## 7. Agile approaches and UCSD

Recently, agile approaches to software development have gained a lot of attention. The rationale behind the agile perspective is to shift the overall focus of software development to a more agile or 'lightweight' perspective. This shift can be seen as a contrast to more formal commercial processes. Agile is not a single, well defined process, instead, it is a generic name for several different processes or methods, sharing a set of core ideas, values and principles of software development. The principles are defined in the Agile Manifesto (Agile Alliance 2001). The most well known of the agile processes is probably eXtreme Programming, XP (Beck 2000).

What is interesting about agile methods is that they are addressing some of the problems of the development process that we found in our research project. For instance, the project focused on short-term goals such as producing models and other artefacts while loosing the overall goal of delivering a usable system. Other problems include use-case mania and poor understanding of the design documentation. Agile processes emphasize the pragmatic use of light, but sufficient rules of project behaviour and the use of human and communication oriented principles (Cockburn 2002). Hence, people are more important than processes and tools. Working software is more important than comprehensive documents and model building, Models and artefacts are only means of communication; consequently prototyping and simple design representations are preferred. Agile developers argue that projects should be communication centric, which implies that effective human communication with project members and users are important, e.g. face-to-face is the ideal way of communicating within a project and with users. Usually, there is a direct collaboration with users and customers – preferably, users and developers should sit in the same room during development.

The problems with the agile approach, is that the different processes have not paid much attention to usability and UCSD. The main focus of agile methods is on delivering working software. This is of course excellent, as usable software also must be delivered and be working. But to get there, the development is focused on making coding effective and there is a risk that usability issues get lost as there is no explicit user-centred focus. Agile projects include some roles that are supposed to work with user interface design and user requirements, but this is in most cases not enough. The whole project must be committed to the importance of usability. Another problem is that the users involved in the development are not always end users. Sometimes they are customers or domain experts. The agile methods seldom make a difference.

Agile processes do not themselves apply to all the key principles of UCSD, but so far we have not seen any reason why agile processes could not be customized or adapted to UCSD.

## 8. Discussion/conclusions

The reader may ask why we have defined yet another set of principles for user-centred systems design, since those existing are not used or do not work the way they were intended. Below, we discuss some of the main reasons why we believe a more precise definition of UCSD is required.

Our pilot study shows that even with an explicit commitment to UCSD and a usability focus, usability may get lost in the software development process. Since few projects have the explicit goal to produce systems with poor usability, we believe that there are obstacles to usability and user involvement in the actual development process. Such obstacles have been described in numerous studies, for instance, Poltrock and Grudin (1994) and Wilson *et al.* (1996, 1997). Our main concern has therefore been to address shortcomings and obstacles in the development process that derail the focus on usability and users' needs.

User-centred design (UCD) methods have gained a great deal of attention recently. According to a recent study (Vredenburg *et al.* 2002) the opinion is that user-centred methods generally increase the utility and usability of computer systems. However, the degree to which organizations adopt UCD methods varies significantly. There is, according to the study, no information on whether or not it is possible to save time and resources by adopting UCD methods. Cost-benefit tradeoffs are, nevertheless, a key consideration when adopting UCD methods (see for example Donahue 2001) This calls for close integration of UCD methods into the development process. Unfortunately, the most common approach is to perform single usability activities using informal UCD methods (Hudson 2001). Such an add-on approach to usability increases the risk of its being cut out when deadlines get tight. We believe that usability faces the risk of becoming a sidecar problem – if somebody in the project is pointed out as having the responsibility for usability all others involved resign from their part of the responsibility. Thus, cost-benefit analysis may in certain situations be used as an argument against usability activities rather than for if they are not tightly integrated.

In a survey examining the attitude about strategic usability (Rosenbaum *et al.* 2000) the authors identified the following obstacles to UCD:

- Resource constraints (28.6%);
- Resistance to UCD/usability (26.0%);
- Lack of understanding/knowledge about what usability is (17.3%);
- Better ways to communicate impact of work and results (13.3%);
- Lack of trained usability/HCI engineers (6.1%);
- Lack of early involvement (5.1%);
- No economic need – customers not asking for usability (3.6%).

We believe that all of these factors are related to a lack of knowledge on how to apply UCD methods and their potential benefits which provides another reason for defining and describing UCSD in more specific terms.

Many organizations pay lip service to usability and UCSD but seem at a loss as to how to achieve it. We have studied organizations that claim that they are committed to usability and UCSD but who are not willing to change their practices in developing software. The same problem applies on the individual level. There is a growing concern among software developers about the usability of the products or software they release on the users. But they often do not know what to do about it.

Yet another reason for a more precise definition of UCSD is that many organizations still do not recognize the benefits of involving users in the development process, despite the fact that active user involvement was judged to be the number one criterion on how to be successful in IT-development projects in the CHAOS report (Standish Group 1995). Clegg *et al.* (1997), for instance, report that most projects in their study had failed to involve users in a satisfactory manner. Nor did they adopt an integrated approach. The impact of new technology on work organization and job design was considered '. . .hugely important but largely ignored in practice' and if addressed, it was usually late in the process and because it was discovered that the new piece of technology was going to change job designs.

UCD has also been criticized on the grounds of its being ambiguous and vague. Constantine and Lockwood (2002), for example, claim that UCD is a '. . .loose collection of human-factors techniques united under a philosophy of understanding users and involving them in design. . .Although helpful, none of these techniques can replace good design. User studies can easily confuse what users want with what they truly need. Rapid iterative prototyping can often be a sloppy substitute for thoughtful and systematic design. Most importantly, usability testing is a relatively inefficient way to find problems you can avoid through proper design' (Constantine and Lockwood 2002: 43). Their remedy is 'usage-centred engineering' where the emphasis is on the usage, not the users, and on model-driven development. We readily agree with the critique against UCD, but not with the remedy. Model-driven approaches rely on skilful designers/developers using abstract models of the domain to base their design on. Model-driven approaches represent a move away from user-centred design, reducing user involvement to that of the users being informants rather than co-designers. We believe, and argue in this paper, that user participation is a key success factor for designing for usability (see also Standish Group 1995, Gould *et al.*, 1997, and ISO 13407 1999) and that software development needs to move towards a user-centred approach rather than away from it. Computer systems (in particular in a work context) must support not only the official rules and version of the work practices but also the particularities in each situation (Sachs 1995, Beyer and Holtzblatt 1998, Harris and Henderson 1999), which required a deep understanding of the context of use. Few development teams have that understanding, and we believe that writing requirements documents or creating abstract models is simply not enough to create that kind of understanding. Only the users themselves can provide that.

To summarize the above discussion, we believe that user-centred systems design must be defined in terms of a process where usability work and user involvement are tightly integrated with the development process. Adding the key principles, furthermore, helps in communicating the essence of UCSD where user involvement is an essential part. By providing a more precise definition of UCSD, we can also avoid problems with ambiguity and vagueness and argue against the use of approaches that are not user-centred.

Hence, the main aim of our definition and key principles is to support the *development process*. This can be achieved by incorporating roles, activities and artefacts for maintaining a focus on usability and users' needs throughout the entire system lifecycle. The definition and key principles may also be used when specifying a UCSD process or when customizing a commercial development process, such as Rational Unified Process – RUP (Kruchten 1998). The key principles originate from our experiences and research in contract and in-house development of bespoke software for work situations. We nevertheless see a potential for applying them in other types of development projects.

Our research, as well as our experiences, show that by applying the definition of and key principles for UCSD, the chances increase of identifying problems in time to do something about them. Consequently, the chances of producing usable software increase.

Finally, we would like to emphasize that what we want to achieve is not simply yet another usability method. We see UCSD as, a new paradigm requiring a profound shift of attitudes towards systems development and user involvement. The attitudes that are required for a truly user-centred approach are embodied in the key principles.

## Notes

1. A *principle* is a commonly accepted fundamental rule or law that can be used to define other principles.
2. To us the usability designer is a role that has a clear position in the development project (see for instance Göransson and Sandbäck 1999). Usability champion has more of a mentor status and is not a role that somebody can shoulder. To be able to act as a usability champion you must have extensive knowledge and experience of the work in practice and also an ability to act as a mentor.
3. Usability is defined as 'the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction, in a specified context of use' (ISO 9241-11 1998), Please note that this definition *includes* the concept of utility or usefulness, often seen as separate from usability.
4. Please note that involving users on a full-time basis in a project quickly turns them into domain experts rather than representative users. It is therefore important to involve user representatives on a temporary basis as well.

## Acknowledgements

This project was performed with financial support from the National Agency for Innovation Research, the Council for Work Life and Social Science Research, and the Swedish National Tax Board. The input received from the number of seminars leading up to the principles, such as IFIP TC 13, and our industrial partners Enea Redina AB, TietoEnator AB and Antrop AB was greatly appreciated.

## References

Agile Alliance. 2001, Manifesto for Agile Software Development. Available: http://www.agilealliance.org.

Beck, K. 2000, *Extreme Programming Explained: Embrace Change* (Boston, MA: Addison-Wesley).

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

Bödker, S. 1998, Understanding representation in design. *Human-Computer Interaction*, **13**(2), 107–125.

Boehm, B. 1988, The spiral model of software development and enhancement. *IEEE Computer*, **21**(5), 61–72.

Boivie, I., Åborg, C., Persson, J. and Löfberg, M. 2003, Why usability gets lost, or usability in in-house software development. *Interacting with Computers*, **15**(4), 623–639.

Buur, J. and Bödker, S. 2000, From usability lab to 'design collaboratorium': reframing usability practice. *Proceedings of DIS 2000* (New York).

Calde, S., Goodwin, K. and Reimann, R. 2002, SHS Orcas: The first integrated information system for long-term healthcare facility management. Conference on Human Factors and Computing Systems, Case studies of the CHI2002/AIGA Experience Design Forum. (New York, NY: ACM Press).

Clegg, C., Axtell, C., Damodaran, L., Farbey, B., Hull, R., Lloyd-Jones, R., Nicholls, J., Sell, R. and Tomlinson, C. 1997, Information technology: a study of performance and the role of human and organizational factors. *Ergonomics*, **40**(9), 851–871.

Cockburn, A. 2002, *Agile Software Development* (Boston, MA: Addison-Wesley).

Constantine, L. L. and Lockwood, L. A. D. 2002, User-Centered Engineering for Web Applications. *IEEE Software*, **19**(2), 42–50.

Cooper, A. 1999, *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity* (Indianapolis, Indiana: SAMS).

Donahue, G. M. 2001, Usability and the bottom line. *IEEE Software*, **18**(1), 31–37.

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

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

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

Gulliksen, J. and Göransson, B. 2001, Reengineering the systems development process for user centered design. In Michitaka Hirose (ed.) *Proceedings of INTERACT 2001* (Amsterdam: IOS Press).

Göransson, B. and Sandbäck, T. 1999, Usability designers improve the user-centred design process. In *Proceedings for INTERACT'99*, (Edinburgh, UK).

Harris, J. and Henderson, A. 1999, A better mythodology for system design. In M.G. Williams, M. W. Altom, K. Ehrlich and W. Newman (eds) *CHI 1999 Conference on Human Factors in Computing Systems Proceedings* (ACM Press) 16–21 May, 1999.

Hudson, W. 2001, Toward unified models in user-centered and object-oriented design. In M. Van Harmelen (ed.) *Object Modeling and User Interface Design: Designing Interactive Systems* (Boston: Addison-Wesley).

ISO 9241-11. 1998, *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). Part 11: Guidance on Usability* (Geneve: International Organization for Standardization).

ISO 13407. 1999, *Human-centered design processes for interactive system* (Geneve: International Organization for Standardization).

Jacobson, I., Booch, G. and Rumbaugh, J. 1999, *The Unified Software Development Process* (Reading, MA: Addison Wesley Longman Inc.).

Kapor, M. 1990, *Software Design Manifesto* (reprinted in Terry Winograd's *Bringing Design To Software,* Addison-Wesley, 1996).

Karat, J. 1996, User Centered Design: Quality or Quackery?, in the ACM/SIGCHI magazine, *Interactions July + August 1996*.

Karat, J. 1997, Evolving the scope of user-centered design. *Communications of the ACM*, **40**(7), 33 – 38.

Kruchten, P. 1998, *The Rational Unified Process—An Introduction* (Reading, MA: Addison Wesley Longman Inc.).

Kyng, M. 1995, Making Representations Work. *Communication of the ACM*, **38**(9), 46 – 55.

Mathiassen, L. and Munk-Madsen, A. 1986, Formalizations in Systems Development. *Behaviour and Information Technology*, **5**(2), 145 – 155.

Mayhew, D. J. 1999, *The Usability Engineering Lifecycle, A Practitioner's Handbook for User Interface Design* (San Francisco, CA: Morgan Kaufmann Publishers Inc.).

Nielsen, J. 1993, *Usability Engineering* (Cambridge, MA: AP Professional).

Norman, D. A. 1986, Cognitive engineering. In D. A. Norman and S. W. Draper (eds) *User Centered Systems Design* (Hillsdale, NJ: Lawrence Erlbaum Associates Inc.).

Norman, D. A. and Draper, S. W. (eds). *User Centred Systems Design* (Hillsdale, NJ: Lawrence Erlbaum Associates Inc.).

Poltrock, S. E. and Grudin, J. 1994, Organizational obstacles to interface design and development: Two participant observer studies. *ACM Transactions on Computer-Human Interaction*, **1**(1), 52 – 80.

Rosenbaum, S., Rohn, J. A. and Humburg, J. 2000, A toolkit for strategic usability: results from Workshops, Panels and Surveys. In T. Turner, G. Szwillius, M. Czerwinski and F. Paterno (eds) *CHI 2000 Conference on Human Factors in Computing Systems Proceedings*. 1 – 6 April, 2000, ACM Press.

Sachs, P. 1995, Transforming work: collaboration, learning, and design. *Communications of the ACM*, **38**(9), 36 – 44.

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

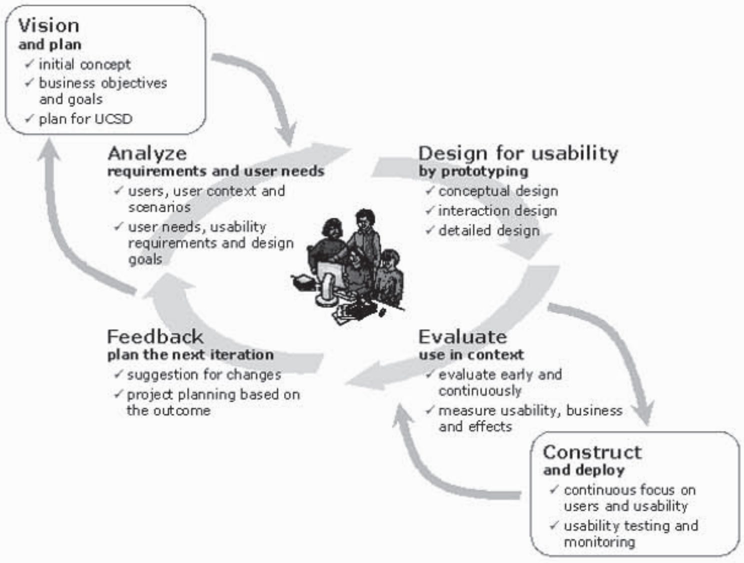
Vredenburg, K., Mao, J.-Y., Smith, P. W. and Carey, T. 2002, A survey of user-centered design in practice. In *Proceedings of CHI'2002 Conference on Human Factors in Computing Systems Proceedings* (Amsterdam), pp. 471 – 478.

Wilson, S., Bekker, M., Johnson, H. and Johnson, P. 1996, Cost and benefits of user involvement in design: practitioners' views. In A. Sasse, J. Cunningham and R. Winder (eds) *People and Computers* (London: Springer Verlag), pp. 221 – 240.

Wilson, S., Bekker, M., Johnson, H. and Johnson, P. 1997, Helping and hindering user involvement – A tale of everyday design. In C. Ware and D. Dixon (eds) *Proceedings of CHI'97*. ACM.

**Appendix 1: The UCSD poster**

## Key Principles for User-Centered Systems Design


Vision
and plan
✓ initial concept
✓ business objectives and goals
✓ plan for UCSD
Analyze
requirements and user needs
✓ users, user context and scenarios
✓ user needs, usability requirements and design goals
Design for usability
by prototyping
✓ conceptual design
✓ interaction design
✓ detailed design
Feedback
plan the next iteration
✓ suggestion for changes
✓ project planning based on the outcome
Evaluate
use in context
✓ evaluate early and continuously
✓ measure usability, business and effects
Construct
and deploy
✓ continuous focus on users and usability
✓ usability testing and monitoring

**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, 2003