Usability Design—Extending Rational Unified Process with a New Discipline

Bengt Göransson [1,2], Magnus Lif [1], Jan Gulliksen [2]

[1] Enea Redina AB, Smedsgränd 9, SE-753 20 Uppsala, Sweden
Bengt.Goransson@enea.se, Magnus.Lif@enea.se

[2] Dept. for IT/HCI, Uppsala University,
PO Box 337, SE-751 05 Uppsala, Sweden
Jan.Gulliksen@hci.uu.se

Abstract. A new discipline, Usability Design, is proposed as an extension to Rational Unified Process (RUP). The aim is to make RUP more user-centred. The discipline springs out of best practices for user-centred design and contains activities that have been carefully tested. Five new roles are introduced to take responsibility for the different activities. An example is given to show the content of the workflow, the workflow details and the activities. The Usability Design discipline help projects to focus on usability and the users throughout the system development lifecycle.

1 Introduction – Usability and User-Centred Design

There are both financial and ethical reasons why usability is worth investing in. An application with high usability can make the user more effective and in less need of support and education. The risk of users making mistakes will be reduced and a system with high usability will provide a less stressful environment for the users.

In the International Standard ISO 9241-11 [7] 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." The effectiveness of a system relates to the work objectives (goals). The efficiency relates to effectiveness in relation to the resources needed to perform the tasks. Satisfaction, according to ISO 9241, concerns acceptability and comfort.

Gould, Boies & Ukelson [2] claim that to build a usable system it is crucial to:

- have early – and continual – focus on users
- do empirical measurement
- do iterative design – redesign
- do integrated design – wherein all aspects of usability evolve together

For the successful integration of usability it must be incorporated in the software development process [9]. When we speak about user-centred design we refer to a process focusing on usability throughout the entire development process and further throughout the system life cycle [3]. Although user-centred design (UCD) is no

guarantee for usable systems, we argue that without it, high usability is usually nothing more than a coincidence.

2 The Rational Unified Process

Rational Unified Process is a software engineering process [10] that significantly has contributed to the software development practice. Today, it is widely used[1] and has contributed to creating a more unified view on software development. However, from a usability point of view RUP has several drawbacks. Problems with the use of the process have been observed in several projects [4]. It does not provide the support needed to produce a usable system.

2.1 How User-Centred is RUP?

In a study by Gulliksen, Göransson, Boivie, Blomkvist, Persson & Cajander [3] the outcome of a project was compared with a list of principles for user-centred design. It shows how a project using RUP with clear intentions to apply a UCD approach ran into several problems that made it difficult to pursue with that approach. Some of the problems are briefly described below:

- No lifecycle perspective on UCD. The developers focused on short-term goals, such as, producing the models and specifications prescribed by RUP. The long-term goals and needs of the users regarding their future work situation were ignored or forgotten.
- Usability designers were ignored. Despite the work produced by the usability designers being of high quality, their results and opinions were ignored in the late stages of the project.
- "Use case mania". When the project started the organisation 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.
- Poor understanding of the design documentation. The user interface design was specified in the Unified Modelling Language (UML) and the users were invited to review it. The users had severe difficulties predicting their future use situation based on the UML notation.
- Problems establishing a user-centred attitude. Single individuals in a project can make a crucial difference when it comes to UCD. In this project people were more concerned about their personal goals than about the needs of the users.

Our view is that RUP is not a user-centred method in itself. On the contrary, there are fundamentals in RUP that actually prohibits a true UCD process:

[1] RUP has become the de-facto standard for software development in Sweden, and currently almost all of our studied development organisations use RUP.

- RUP is an “architecture-centric” and a “use case driven” process rather then user-centred. ”Use cases drive the Rational Unified process end-to-end over the whole lifecycle, but the design activities are centred around the notion of architecture - system architecture, or for software-intensive systems, software architecture.” [8]
- The definition of iteration within RUP differs significantly from how it is defined in for instance UCD. In RUP, an iteration is “A distinct sequence of activities with a base-lined plan and valuation criteria resulting in a release (internal or external)” [17]. The activities inside the iteration are laid out as a waterfall. This prohibits iterations to formally occur within workflows and activities. From a UCD point of view an iteration is a refinement of a certain part of the system (an increment), going through the stages analysis, design and evaluation until goals are reached.
- Usability related activities do only occur within the Requirements discipline (workflow) and primarily in the Elaboration phase. To consider usability only when gathering requirements is fundamentally wrong. From a UCD perspective usability related activities should be included from the start of the project to the very end.
- RUP focuses too much on artefacts. There is a risk that the “big picture” gets lost and that each role puts too much effort in writing documents (fill out forms). Also, the project members are encouraged to get “his/her work done” not to collaborate. In UCD a lot of the work is done in multi-disciplinary teams. UCD is mainly user- and process-driven, with a focus on the activities. The methods vary and so does the outcome (the artefacts).
- No support for user interface design. The focus in RUP is on the use cases and the interaction between the actors and the use cases but it does not give enough support when it comes to designing the user interface supporting that interaction [12]. UML does not really distinguish between human and non-human actors. Designing an interface to another system is completely different from designing an interface for real user.

2.2 Use Cases—Promising But Not Enough

For software developers use cases is the longed-for answer to the quest for a formal notation to express issues relating to the use of the systems and it has now more or less become a de facto standard. It must be acknowledged that use cases constitute a tool for software developers, that is a vast improvement compared to its predecessors. However, based on our experience in practice we have encountered several problems with use cases in relation to a user-centred development. Some of these are discussed below.

The notation of use cases is difficult for users to understand. The reason for this is that users do not interact with the system. They perform work tasks supported by the computer. As a consequence of this it is difficult for users to relate to use cases and understand practically how these will influence the design of the user interface and the interaction.

Use cases are sometimes a good way of communicating internally between usability designers and software developers. Unfortunately, we have observed that

software developers and usability designers have **different requirements on the size of the use cases**. Many software developers prefer small use cases when specifying the functionality. For usability designers use cases instead have to be large to correspond to users' work tasks. We have observed that using small use cases can cause problems since users usually have to communicate with several use cases to perform one work task. **The way use cases are described can have implications on the design space.** We have seen several examples where the use case description defines exactly what interaction elements to present on the screen. Constantine & Lockwood [1] claim that the concept of use case is not clearly enough defined. As a result there are huge variations in style for writing narratives that describe use cases. They also emphasise that use cases can cause problems for the design of the user interface if they are describing the interaction between the user and a particular interface. Instead Essential use cases are introduced where the focus is on the intentions of the users rather then the interaction, making them device independent. However, using use cases as the only means of communication can be insufficient Using prototypes as a complement to illustrate ideas and functionality early on in the process is necessary. A prototype is usually a lot easier for a user to understand and a lot of poor design solutions can be avoided at an early stage.

Muller, Haslwanter and Dayton [14] claim that there are other problems with the use case driven approach. One being that **use cases are modelled with the software system as the focus of attention**. They mean that the use cases give too little priority to the end users and that each use case is a definition of user actions by software engineers. To overcome these problems it is necessary to model the use cases in participation with the end-users. Otherwise there is a great risk that the application will not support the users efficiently in their work.

2.3 RUP—If You Can't Beat Them, Join Them!

The conclusion from this is that RUP and use cases as such are not ideal for user-centred design. But, to manage having an impact in practice we have decided to relate our development process to a RUP based approach, since this is one of the most widely used processes in the large, in-house development organisations we typically work with. What's available in RUP today, with regards to usability and user-centred design?

Plug-ins to RUP. The User Experience plug-in to RUP is developed and maintained by Rational. Primarily, it is focused on website projects, and does not cover the whole user-centred design process. There is a strong focus on creative design and business development, not so much on usability. Therefore, we see a need for a more general plug-in with a strong usability focus.

Usability Engineering Roadmap and User-Centred Design Concept Paper. There is a so-called roadmap for Usability Engineering available in RUP (roadmaps provide a way of describing how to use the general-purpose process described in RUP to solve specific types of problems). A shortcoming of the roadmap is that it does not add any new user-centred activities to the process: "This roadmap explains how the Rational Unified Process (RUP) currently addresses usability engineering techniques". [17] We believe, that for RUP to become more user-centred, there are

certain activities, roles and artefacts that must be added. However, the roadmap, together with the concept paper User-Centred Design, is a good starting point when aiming for a more user-centred approach within the current boundaries of RUP.

User Interface Design in RUP. Recently, RUP has been extended to provide better support for UI-design [11]. Use case storyboards are introduced to extend use cases with information related to the user interface. This is a formal procedure that captures some of the relevant information. However, if this step-by-step procedure is followed there is still a risk that the user will have to work with a fragmented interface where each use case is realised by one interface. Also, we believe that UI-design is a creative process that cannot be described as a step-by-step procedure.

Other Approaches. In our work with organisations we have seen other approaches for how to make RUP more user-centred, one being to make minor changes in the activities performed by the existing role; User Interface Designer. This is a good attempt, but not enough to make the process user-centred. Another approach would be to change all other disciplines to make the process more user-centred. This would probably be good, but the result would be a completely new method, i.e. it would no longer be RUP.

3 The Usability Design Discipline

In this paper we propose a new discipline, Usability Design (for short: *UD discipline*), extending RUP (see Fig. 1). Our aim with the UD discipline is to complement RUP to make it more suitable for systems development where usability is acknowledged to be important. The way to do this is to apply a more user-centred approach to the process.

The proposed discipline is drawn from our previous research and published literature. We have previously observed that UML and use cases do not give sufficient support for user interface design and proposed a method, User Interface Modelling, for gathering requirements relevant when designing for usability [12]. Methods and tools with a strong user focus for designing usable user interfaces have been presented, such as the workspace metaphor [13], corporate style guides [16], and analysis of information utilisation [5]. Also, problems with RUP and user-centred design have been identified and the usability designer has been introduced as an important role in software development projects [4]. Apart from this we have proposed 12 principles to use as a guide in UCD projects [3]. This together with our extensive practical experience as usability consultants in large software development projects forms the base on which the UD discipline for RUP is developed.

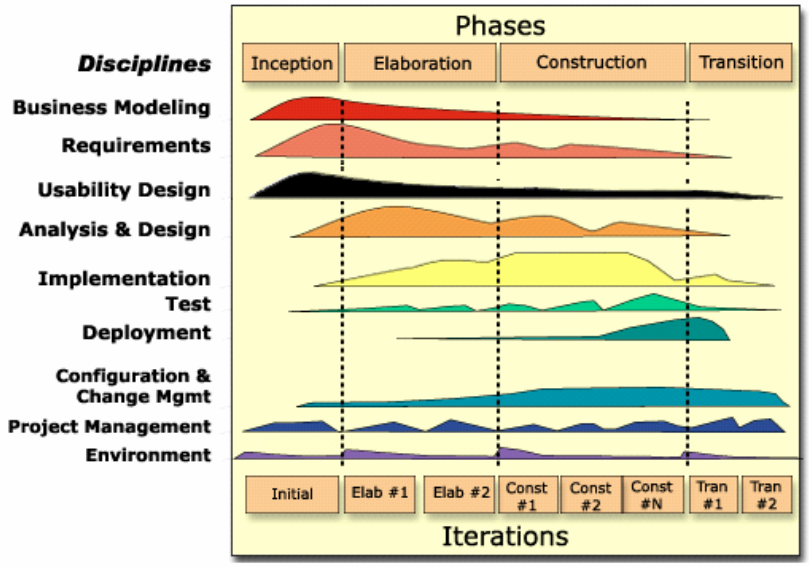

Fig. 1. The Usability Design discipline merges into the overall architecture of the RUP.

3.1 Best Practices

Included in the UD discipline is a best practice document with 12 principles to apply in UCD projects [3]. It covers aspects of the process that cannot be illustrated with the workflow, activities and roles.

The best practices imply that: *User-centred design is a process focusing on usability throughout the entire development process and further throughout the system life cycle*. The following best practices communicate the core of this process:

- **User focus**—the goals of the activity, the work domain or context of use, the users' goals, tasks and needs should control 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**—base lined 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 conducted by effective multidisciplinary teams.
- **Usability champion**—usability experts should be involved from the start of project to the very end.
- **Holistic design**—all aspects that influence the future use situation should be developed in parallel.
- **Process customisation**—the UCD process must be specified, adapted and implemented locally in each organization. Usability cannot be achieved without a user-centred process. There is, however, no one-size-fits-all process.
- **A user-centred attitude must be established**—UCD requires a user-centred attitude throughout the project team, the development organisation and the client organisation.

3.2 Description of the Usability Design Discipline

The UD discipline is concerned with developing systems with a focus on usability.

Purpose
The UD discipline is an extension to RUP. Its main purpose is to put focus on usability through user-centred design. It encapsulates the best practices for UCD and makes it possible to practice UCD within the RUP framework.

Roles
The new roles suggested in the UD discipline are:

- **Usability Designer.** This role is a usability champion and lead. The individual acting in this role must have extensive experience. She should be an HCI expert with a background in computer science and/or behaviour sciences and with hands on experience from the entire user-centred design process. The Usability Designer is responsible for planning and managing the activities in the UD discipline, as well as carrying out some of them.
- **Field Study Specialist.** This role should be experienced in using different usability methods and techniques for categorising user groups and understanding user needs. Typically this person should be able to plan, perform, and analyse user studies. She should also be able to come up with design solutions to meet the identified needs. A general background in HCI is preferable.
- **Interaction Designer.** The Interaction Designer is responsible for the conceptual, interaction and the detailed design. This includes: creating the overall interaction scheme and layout; defining the dynamics of the user interaction and the navigation paths and specifying the information structure. A combination of HCI knowledge and design experience is needed.

- **Graphic Designer.** This role is responsible for the visual shaping of the user interface and works closely together with the Interaction Designer. A background in creative design and experience from interactive media is needed.
- **Usability Evaluation Specialist.** This role is responsible for planning, preparing, executing and reporting usability evaluations. She should be an expert in performing evaluations in usability laboratories as well as in the field. The background for this role is HCI with a focus on behaviour science.

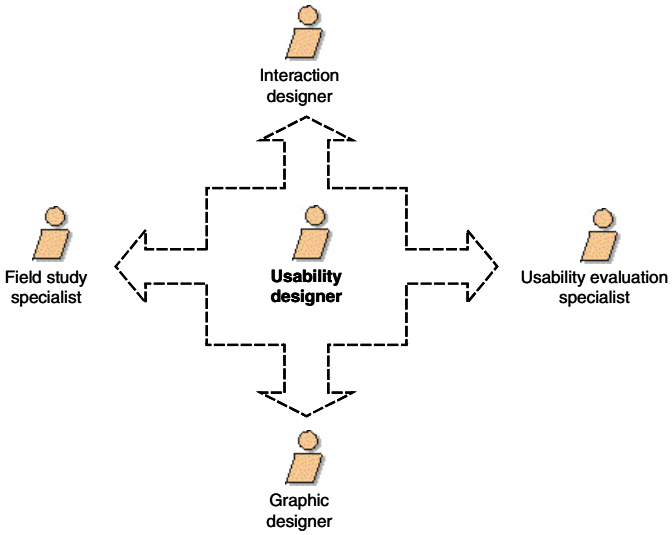

Fig. 2. Illustrating the roles in the Usability Design discipline.

We can clearly see the benefits of having multiple specialised roles in projects. However, if each role is represented by one person there is a risk that important information will get lost when, for instance, the Field Study Specialist hand over the result to the Interaction Designer. Therefore, in many projects, the person acting as the Usability Designer also covers the other roles (see Fig. 2).

All roles must be able to work effectively with, and communicate well with users. Active user participation is crucial for the UD discipline.

The Workflow

The workflow shows, on a high level, the user-centred work performed in each iteration of the UD discipline (see Fig. 3**)**.

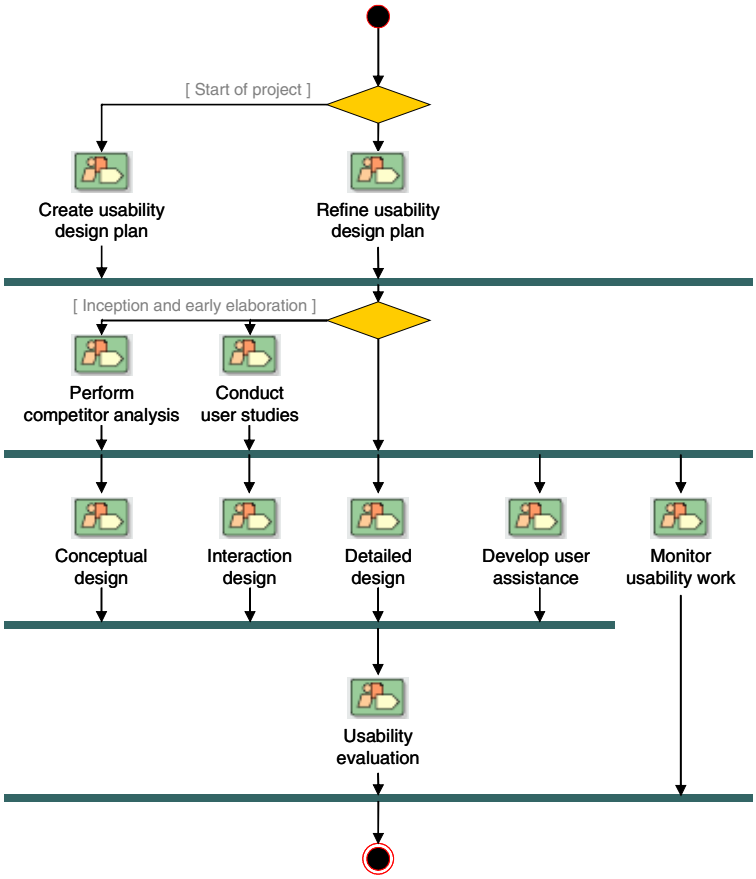

Fig. 3. The core workflow of the Usability Design discipline.

Following is a short description of the main activities:

- **Usability design plan.** A number of methods can be used to perform the different activities in the workflow. In this workflow detail the actual selection of methods is done. The usability design plan is created at the start of the project and is then refined during each iteration. Activities are:
  - Detailed planning of user-centred activities
  - Plan user involvement
  - Conduct project usability workshop
- **Conduct user studies.** The aim of this workflow detail is to understand the potential users of the system, their needs and the context of use. The goals for the

system, design criteria and usability goals are specified. Information is gathered through interviews, observations, workshops, etc. Activities are:

- User profiling
- Use and task analysis
- Analyse context of use and work environment
- Prototyping (optional)

- **Perform competitor analysis.** If a website or a product is developed, similar websites or products may be analysed. For an in-house support system, studies of similar businesses and systems may be appropriate. The aim is to get an insight into state-of-the-art solutions. Activities are:
  - Select relevant systems or businesses to analyse
  - Inspect systems or businesses
- **Conceptual design.** The conceptual design describes the overall structure of the user interface. Usage scenarios are developed together with users. The main components of the user interface are identified. Usually paper prototypes are produced to illustrate potential designs solutions on a high level. Activities are:
  - Brainstorm concepts
  - Develop usage scenarios
  - Develop mock-ups
- **Interaction design.** During this workflow detail the conceptual design is developed into an interactive design illustrating how the user can interact with the system. It shows the navigation, the information and the functionality, simulating the real system. Activities are:
  - Develop information architecture
  - Develop navigation structure
  - Content design and layout
  - Prototypes
- **Detailed design.** The detailed design includes individual parts of the screen, such as data fields, input fields, menus, and buttons, i.e. the thousands of details that can be found in a graphical user interface. Activities are:
  - Detail content design
  - Detail graphic design
  - Refine prototypes
- **Develop user assistance.** This is a parallel design activity focusing on integrating user assistance into the final system. On-line help systems, manuals and user training material are developed. If new ways of carrying out tasks or ways of making the work more efficient are found during the user studies, new work procedures, etc. may be defined within the scope of this workflow detail. Activities are:
  - Develop user assistance material
  - Integrate assistance in business
  - Integrate assistance in system
  - Integrate assistance in work environment
- **Monitor usability work.** At this stage the user interface should be designed and the usability evaluated. However, there will always be smaller design decisions to make “on the fly”. Also, change requests may have impact on the design. This

workflow detail is especially important during the Construction phase. Activities are:

- Give usability support
- Make ad-hoc design decisions
- Refine designs

- **Usability evaluation.** In this workflow detail the usability of the design solutions are evaluated against the usability goals. Evaluations can be performed on preliminary sketches as well as on fully interactive prototypes and the final system. Activities are:
  - Plan evaluation
  - Determine usability metrics
  - Select method
  - Recruit representative users
  - Perform evaluation
  - Analyze results

It is fundamental to see the workflow as an iterative process, where the design solutions are evaluated, which in turn leads to design improvements. The different design workflow details will be intermingled as the project develops. It is important to see that these activities are carried out in parallel. The monitoring of usability work is foremost in focus during the construction phase, when most of the design is in place.

3.3 An Example: Conduct User Studies

An example is given to illustrate what the workflow details and activities contain. Here we will take a closer look at the workflow detail *Conduct user studies* (see Fig. 4).

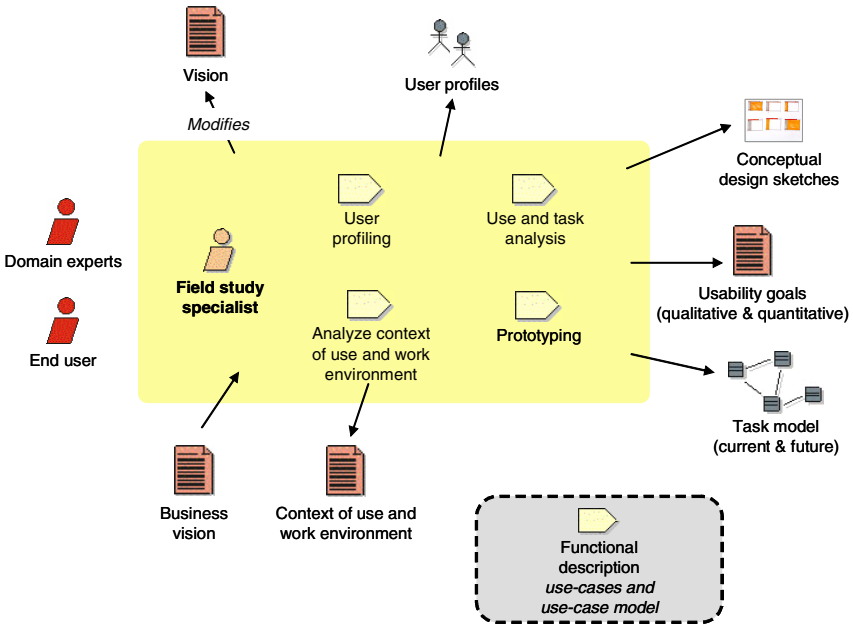

Fig. 4. Workflow detail Conduct User Studies.

The purpose of this workflow detail is to get a thorough understanding of the users and their needs. This is crucial input to the design process. This information is primarily collected and elicited through communication with representative users of the system.

Roles

The work role responsible for this workflow detail is the Field Study Specialist. Secondary roles are the Domain expert and the End user. The Field Study Specialist should be experienced in using different usability methods and techniques for categorising user groups and understanding user needs. Typically this role should be able to plan, perform, and analyse user studies. She should also be able to come up with design solutions to meet the identified needs. The information may be collected from workshops, one-to-one interviews, observations, questionnaires, participatory design activities, etc. Domain Experts are users or former users that are experts in certain areas related to the new system. Typically these users are very experienced and have a vision about how to improve the business and work procedures. Domain Experts should be part of the actual project team. End users on the other hand are the people in the organisation that will use the system in the future. End users are typically involved during field studies and usability evaluations.

Activities

Four activities are performed: *User profiling* where users are categorised into representative groups. This is a more in-depth description of the users than the description of the actors. In *Analyse context of use and work environment* the users

entire work situation and context is analysed and described. *Use and Task* analysis is the core of this workflow detail. We need to understand what people do to be able to develop useful applications. Finally *Prototyping* is included to visualise conceptual design solutions, primarily to elicit requirements and user needs. These concepts should be on a high level, i.e. not too detailed.

Artefacts
The input to this workflow detail is the business vision. The outcome is a number of different artefacts. The user profiles describing the user groups, a list of qualitative and quantitative usability goals, a task model describing both the users' current tasks and the way tasks could be performed in the future, conceptual design sketches and a description of the context of use and work environment. This workflow detail will also provide input to other RUP artefacts such as the vision, the use case model and to the use cases.

4 Discussion

To establish usability activities in software development in practice it must be heavily integrated into the software development process. We have seen a number of RUP projects in which usability professionals have tried to introduce usability activities, but not fully succeeded. There seem to be a tendency to disregard or ignore activities that are not integrated into the process and as such the usability tend to get left out [18]. The UD discipline for RUP has been introduced with the aim to facilitate the production of more usable systems when using RUP. Both RUP and the UD discipline are general to their nature and can be used in all kinds of systems development projects. However, as with RUP itself it is important to tailor the UD discipline and its use together with RUP to the development organisation's needs and to the particular project. This means considering what parts of RUP and the new discipline to use, responsibilities for the roles and how to document things. Each development organisation and project is unique.

In the discipline five roles are introduced, the field study expert, the interaction designer, the usability evaluator, the graphic designer and the usability designer. In many projects the same person will shoulder more than one of these roles. Therefore, we have introduced the Usability Designer as a role in the development organisation [6]. This role has been adopted by a number of organisations and the response has mostly been positive. It is important to note that to be able to fulfil the needs of the project this person has to be experienced. Nielsen [15] claims that a usability expert should have a least 10 years experience of working with usability in projects.

The main focus so far has been to develop and optimise the UD discipline itself. Our intention is now to focus more on how it will work together with the other disciplines. We will do this by continuing to introduce the UD discipline in large development organisations and feed back the results to the HCI and software engineering community. The discipline, as presented here, has only partly been used in different development projects in practice. Having a project performing their

development entirely according to the discipline, as specified here, remains to be done.

References

1. Constantine, L. & Lockwood, L.: Structure and Style in Use Cases for User Interface Design. In Mark van Harmelen (Ed.) Designing Interactive Systems: Object modelling and user interface design (2001)
2. Gould J.D., Boies S. J. & Ukelson J.: How to Design Usable Systems, in Helander, Landauer & Prabhu (eds.), Handbook of Human-Computer Interaction, Elsevier Science B.V. (1997)
3. Gulliksen J., Göransson B., Boivie I., Blomkvist S., Persson J. & Cajander Å.: Key Principles for User-Centred Systems Design. In preparation, http://acsd.hci.uu.se/ (2003)
4. Gulliksen, J., Göransson, B. & Lif, M.: A User-Centered Approach to Object-Oriented User Interface Design, i Mark van Harmelen (Red.), Designing Interactive Systems: Object Modeling and User Interface Design, Addison-Wesley, ISBN 0-201-65789-9 (2001)
5. Gulliksen, J., Lif, M., Lind, M., Nygren, E., and Sandblad, B.: Analysis of Information Utilisation. International Journal of Human-Computer Interaction, 9 (3), (1997)
6. Göransson, B. & Sandbäck, T.: Usability Designers Improve the User-Centred Design Process, i Proceedings för INTERACT'99, Edinburgh, UK (1999)
7. ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). Part 11: Guidance on Usability. International Organization for Standardization, Geneve (1998)
8. Jacobson, I., Booch, G. & Rumbaugh, J.: The Unified Software Development Process, Addison Wesley Longman Inc., Reading, Mass., USA (1999)
9. Kapor M.: A Software Design Manifesto. In: Winograd T. (ed.). Bringing Design to Software, ACM Press, New York, New York (1996)
10. Kruchten, P.: The Rational Unified Process—An Introduction, Addison Wesley Longman Inc., Reading, Mass., USA (1998)
11. Kruchten, P., Ahlqvist, S., & Bylund, S.: User-Interface Design in the Rational Unified Process. In Mark van Harmelen (Ed.) Designing Interactive Systems: Object modelling and user interface design (2001)
12. Lif, M.: User-Interface Modelling—Adding Usability to Use Cases. International Journal of Human-Computer Studies, 3, 243–262 (1999)
13. Lif, M., Olsson, E., Gulliksen, J., and Sandblad, B.: Workspaces Enhance Efficiency—Theories, Concepts and a Case Study. Information Technology and People, 30(4) (2000)
14. Muller, M.J., Haslwanter, J.H, & Dayton, T.: Participatory Practices in the Software Lifecycle In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), Handbook of Human-Computer Interaction (pp. 255-297). Amsterdam: Elsevier Science B.V. (1997)
15. Nielsen, J.: Boxes and arrows: Got usability? Talking with Jacob Nielsen by Chad Thornton. http://www.boxesandarrows.com/archives/print/002321.php (2002)
16. Olsson, E. and Gulliksen, J.: A Corporate Style Guide That Includes Domain Knowledge. International Journal of Human-Computer Interaction, 11(4), (1999) pp. 317–338
17. Rational Software Corporation: Rational Unified Process, version 2002.05.00 (2002)

18. Boivie, I, Åborg, C, Persson, J. & Löfberg, M.: Why Usability Gets Lost or Usability in In-house Software Development, Interacting with Computers, in press (2003)