

***Rational Unified Process (RUP) and its relationship with  
the end-user in the software system development process***

Master Thesis, 20 credits, INF801, in informatics.  
Presented January, 2008

Author: Lorena Azócar Mérida.

Supervisor: Odd T Sten

Examiners: Hans Lundin, Erik Wallin

**LUND UNIVERSITY**  
Department of Informatics

# **Rational Unified Process (RUP) and its relationship with the end-user in the software system development process**

**Lorena Azócar Merida**

Master thesis, January 2008

## **Abstract**

The user-centered design or usability design or human computer interaction there are very close or similar concept, and a very wide subject of study. The Rational Unified Process is very complex by itself. There are many criticisms again The Rational Unified process because it is alleged lack of tools and helps in the user-centered development. Relating these areas by finding out the relationships between the end-user (central point in the user-centered), and the Rational Unified Process in the software development process is the purpose of this study. The study process was divided in three parts: the review of the existing literature about the two main areas, in order to generate relationships between them. Then using the relationships as a foundation for the research, and verifying the relationships.

Keywords: user-centered design, end-user, The Rational Unified Process (RUP).

# Contents

<b>1. INTRODUCTION .....</b>	<b>5</b>
1.1 Background.....	5
1.2. Problem space.....	5
1.3 Research question .....	7
1.4 Purpose .....	8
1.5 Scope.....	8
<b>2. HUMAN-COMPUTER INTERACTIONS .....</b>	<b>9</b>
2.1 Human-computer interactions .....	9
2.2 Participatory design.....	9
2.3 Usability and Human-computer Interaction .....	10
2.4 User and individual user differences .....	10
2.5 User-centered design .....	11
2.6 Key principles for user-centered system design.....	12
<b>3. THE RATIONAL UNIFIED PROCESS .....</b>	<b>14</b>
3.1 History Review .....	14
3.2 Fundamentals of the Rational Unified Process.....	14
3.2.1 The Rational Unified Process is User-case Driven .....	14
3.2.2 The Rational Unified Process is Iterative and Incremental.....	15
3.3 Software Best Practices .....	16
3.3.1 to develop software iteratively .....	16
3.3.2 To manage Requirements.....	17
3.3.3 TO use component-based architectures .....	17
3.3.4 To visually model software .....	17
3.3.5 To continuously verify software quality .....	17
3.3.6 To control changes to software.....	18
3.4 Usability Design as a Discipline in Rational unified process.....	18
<b>4. END-USER AND RATIONAL UNIFIED PROCESS INTERACTIONS.....</b>	<b>19</b>
<b>5. RESEARCH METHODOLOGY FRAMEWORK .....</b>	<b>22</b>
5.1 Qualitative approach .....	22
5.2 Research strategy.....	22
5.3 Generating Relationships (Hypothesis) .....	22
5.4 Data collection.....	23
5.4.1 Literature review.....	23
5.4.2 Purposeful sampling .....	23
5.4.3 Participating Companies .....	24
5.4.4 Interviews.....	24
5.4.4.1 Face-to-face interviews.....	24
5.4.4.2 Telephone interviews.....	24
5.4.4.3 Semi-structured interview .....	25
5.5 Questionnaire.....	25
5.6 Data analysis .....	27
5.6.1 Coding.....	27
5.7 Validity.....	29
5.8 Ethics .....	29
<b>6. DESCRIPTION OF THE COMPANIES .....</b>	<b>31</b>
6.1 Volvo Information Technology AB.....	31
6.2 Guide Redina AB.....	31

6.3 <i>Tempogon AB</i> .....	31
<b>7. PRESENTATION OF THE RESULTS OF THE EMPIRICAL DATA</b> .....	<b>32</b>
<b>8. VERIFYING OF THE RELATIONSHIPS</b> .....	<b>35</b>
<b>9. CONCLUSION</b> .....	<b>38</b>
9.1 <i>Relationships</i> .....	38
9.2 <i>Purpose and research question</i> .....	39
9.3 <i>Problems connected to the research</i> .....	39
9.4 <i>Further Research</i> .....	40

# 1. Introduction

*This chapter will provide a brief description of the research area, the background, problem space and introduces the research question as well as the purpose and scope of this study.*

## 1.1 Background

Since organizations struggle to keep up in an increasingly competitive enterprise world, competitive pressures and rapid technological changes are forcing computer-based system designers toward shorter product development life cycles. The importance of creating a product with a higher standard of quality, reliability, and performance in a relative short time is a desired objective. To achieve this goal is impossible without the implementation of methods that allow for the control and integration of the many facets of software development. (Jacobsson et al 1998).

At the same time, there has been a growing awareness that system success depends on product quality improvement, ease-of-use, and user acceptance. The product should look attractive, and be easy to use, and learn to use by the end-user. This growth has been catalyzed in part by the community of researchers and professionals, leading to the development of human-computer interactions (HCI) and product usability through user-centered designs (UCD) system approaches (Gulliksen & Göransson, 2002).

## 1.2. Problem space

Usability is a central concept in the field of Human-Computer interaction (HCI). One of the main goals of the field is to contribute to the development of more easily utilized products. Usability focuses on early specification and measurable requirements, which goes well with the general ideas of the development process. The problem with measurement is its orientation to aspects of the development process; it ascertains superficial questions concerning the user interface or time efficiency for test tasks in lab environments. These measurements do not have much relevance for the end-users (Löwgren & Stolterman, 2005).

To provide developers with the ability to determine the usability consequences of their design decisions is one of the biggest problems in user-centered design. There are some design guidelines imposed by authorities (ISO) that developers must follow in order to increase usability of the software product, however these standards are merely suggestions and generalities. Such design standards are based upon psychological, cognitive, ergonomic, sociological, economic or computational theories. Developers may not always have the relevant background to understand the consequence of those theories in the design they are creating (Dix et al 1997).

The traditional software lifecycle puts an enormous emphasis on design. Design is seated at the very beginning of the cycle. Development is based on this initial design and developers make an effort to achieve the design as a goal. In practice, designers do not realize all of the requirements for a system before they begin. This is a critical point; not all

the requirements for an interactive system can be determined from the start (Dix et al 1997).

Any approach to interactive system design relies on a clear understanding, early on the design, of the tasks that the user will need to execute. The problem with this is that often only the end-users themselves have the knowledge of what tasks they will perform. Thus, the deficiency use of notations and techniques that support the user's perspective of the interactive systems is at the root of this issue. It is very difficult for an expert on human cognition to predict the cognitive demands that an abstract design may require of a future user if notation does not indicate the kind of information the user will use in order to interact with the system (Dix et al 1997).

Information technology has transformed working life. The proportion of computer users among office workers in Sweden increased from 65% to 90% from 1989 to 1997. (Marklund, 2000). Today, most workplaces use computer systems and, for the most part, they have contributed to improve the quality of the work environment. Computer systems easily and effectively perform many workplace tasks. However, there are many examples of serious problems in relation with the development and implementation of IT support in work activity. (Gulliksen & Göransson, 2002).

Computer systems are often poorly designed and user-unfriendly. This leads to inefficient use and to a variety of cognitive problems (e.g. confusion, lack of overview and memory overload (Nielsen, 1993). Complicated systems are very difficult to understand and operate for the end-user and may negatively influence the work of the end user. This may lead to stress and irritability, which often generates long-term health issues (Gulliksen & Göransson, 2002).

The economic area is also affected. Many unsuccessful projects are stopped, and others are implemented with mediocre results and great expense. IT-developments in particular are known for being frequently late, costly and stopped without achievement of the intended result. (Gulliksen & Göransson, 2002).

Göransson (2004a, p.7-13) states *'that developing computer systems is a risky business. He supported his state with the result of an extensive survey amongst 1.200 Swedish companies was conducted in 2002 by the five major unions within the Swedish Trade Union Confederation (LO) and UsersAward . It reveals that:*

- *Less than 50% of users report that 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 users report that IT-systems provide a proper overview of their work tasks.*
- *Only four out of ten report that internal and external communication has improved.'*

This issue is matter of study outside of Sweden too. The Standish Group has, in its CHAOS-rapport (Standish Group, 1995), the result of a survey of American IT-development projects. The United States spend 250 billion dollars every year on approximately 175.000 different IT-projects.

In this study, 365 IT-companies with 8380 different IT-projects were analyzed with the following result:

- 31.1 % of the companies' projects were cancelled.
- 52.7 % were performed with changed plans
- 16.2 % were performed according to plan.

On average, changing plans increased cost 189 %. Furthermore, 81 billion dollars were spent every year on projects that never lead to any result.

The American study concluded that the common factors among successful projects (16.2 %) were high participation of end-users in the entire development process, and clear specification requirements. Gulliksen & Göransson claim this survey shows the obvious relationship between the user-centered practices and successful projects (Gulliksen & Göransson, 2002 p.22).

### **1.3 Research question**

During the past two decades, research into methods for system development, human-computer interaction, user interactions, etc. has been quite intense. Even after all this research, reality demonstrates that the above-mentioned problems remain unsolved. Even worse, these problems seem to be increasing in number and seriousness. Why is this happening? One credible explanation is that there exists a lack of competence and insufficient knowledge on usability in practical system development, combined with poor understanding of the effects of deficient IT-systems (Gulliksen & Göransson, 2002).

Design is difficult in the development of user-centered systems. There are many different disciplines and realms of knowledge, which require specific care of in practical development work. Usability frequently disappears in the development work because it is difficult to systematically focus on usability during the whole development process (Ottersten & Berndtsson, 2002).

The complexity of new IT-systems demands a strong development methodology. Rational Unified Process (RUP) seems to fulfill these requirements. RUP is an iterative development process, where the development lifecycle is parted in activities called disciplines. These disciplines (requirements, design, development and test) are performed under the four phases: Inception, Elaboration, Construction and Transition. The advantage of using this iterative approach is the earlier detection and corrections of error and misunderstanding (Strand, L, 2001).

With an object-oriented view and strong focus on system architecture, RUP gathers "the best practices", or the product of many years experience (Jacobsson et al 1998). However, the Rational Unified Process is criticized due the lack of user participation in the whole system development. Therefore, RUP is deficient in usability and user-centered design (Gulliksen & Göransson, 2002).

Peter Boersma (2003) writes several reflections about RUP in his article *Introducing User-centered Design to an E-Government Software Development*. He points out that this method “integrates usability concerns into the RUP development approach” but that it is still only “used to describe the technology aspects of the user’s interaction with a Web-based system,” so it does not model the user-centered design aspects.

Uppsala University & Eneas Rendica AB have already presented an approach with the aspiration of creating a new discipline in RUP with the title of Usability Design Expert. The addition of the Usability Design discipline is proposed to treat RUP’s weaknesses and make it possible to apply a user-centered system in the RUP framework (Gulliksen & Göransson, 2002). Nevertheless, this discipline does not have a spread guideline in RUP to date. The question remains:

*What is the relationship between the Rational Unified Process, and the end-user in the software system development process?*

#### **1.4 Purpose**

The purpose of this study is to characterize end-user interaction with the Rational Unified Process in the system development process. The end-user interaction considered as an important factor in the creation of the user-centered design in a Rational Unified Process framework.

#### **1.5 Scope**

The successful IT-system is a combination of different factors, among them the high participation of the end-users in the whole development process, clear specification requirements, good planning and support from the stakeholder. (Gulliksen & Göransson, 2002, p.23). Nevertheless, the focus of this study is the end-user’s interaction with the Rational Unified Process. User-centered design covers a wide area in information technology; however, in this study it is limited to the IT-system as the end user’s support in his/her workplace context.



## 2. Human-computer interactions

*This chapter describes one of the major the subject in this study. The human-computer interaction and the concepts I will be using in the study. The chapter contains a brief history of the HCI, participatory design, user and the individual user differences, user-centered design and ends with the key principles for user-centered design.*

### 2.1 Human-computer interactions

According Löwgren & Stolterman (2005) the field of Human-Computer Interaction (HCI), has its roots in what was, in the late 1970's, called "software psychology methods" and "the scientific tradition". The phenomenon of study was a human interacting with a computer, and the intention was to accumulate empirical knowledge through controlled experiments. This study founded the bases for more general theories concerning human thought and action in front of a computer. A scientific psychology should ideally help to arrange interfaces to make them easier to use, efficient, error free, and enjoyable.

There is no general unified theory of HCI, however the general understanding of HCI groups the following the three major issues: people, the computers and the task that are performing. The system must support the user's task, which gives the fourth focus: usability (Dix et al 1997).

Even before the HCI community, starting around 1990, a philosophy had developed earlier with the Scandinavian school of system development under the name of *Participatory Design*. Participatory Design philosophy is the base of much of the work developed in HCI (Löwgren & Stolterman, 2005).

### 2.2 Participatory design

Participatory design is a process of mutual learning, where developers and users learn from and about each other. Participatory design entails that: not only do users participate in design, but also designers participate in use. (Löwgren & Stolterman, 2005). Participatory design can also be described as a developed philosophy, which involves the end-user in the whole design cycle. The end-user becomes incorporated as a member of the design team in the role of "expert in work". Participatory design has three specific characteristics. First: improve the work environment and task by the introduction of the design. This makes work oriented rather than system oriented. The second one is collaboration: The user takes part in every stage of the project. Finally the iterative approach: the design is evaluated at each stage (Dix et al 1997).

#### 2.2.1 The participative design as law

Participative design has been promoted in law and accepted work practices in Sweden. The Swedish Work Environment Law stipulate that; "*The worker should be given the possibility to participate in the design of his/her own work situation and in changes and development that concerns the work*" (The Swedish Work Environment Act, 2005). However, Participative design has not been widely practiced, perhaps due to the time and cost that this type of project involves (Dix et al 1997).

## 2.3 Usability and Human-computer Interaction

The primary objective of a work system is to allow the user to achieve particular goals in some application domain. In other words, the interactive system must be usable. The designer of an interactive system or work system faces two questions: How can an interactive system be developed in order to ensure its usability? Moreover, how can the usability of an interactive system be demonstrated, or measured? (Dix et al 1997).

There are many usability definitions to be found in literature, but they all are more or less similar in their content. The following definitions are specifically relevant in this study context, since they describe what usability is and how it can be measured.

### 2.3.1 Nielsen's definitions:

According to Nielsen (1993, p.29), usability is a set of components and is associated with five attributes:

- *Learnability*: The system should be easy to learn so that the user can rapidly begin to get work done with the system.
- *Efficiency*: The system should be efficient to the user so that once the user has learned the system, a high level of productivity is possible.
- *Memorability*: The system should be easy to remember so the casual user is able to return to the system after a period of time without the user having to re-learn everything.
- *Error*: The system should have a low error rate, so that users make few errors during the use of the systems. If they do make errors, they should easily recover from them. Furthermore, catastrophic errors must not occur.
- *Satisfaction*: The system should be pleasant to use so that users are subjectively satisfied when using it. They should like it.

### 2.3.2 International Standard ISO 9241 definitions:

The International Standard ISO defines the standard differently. According to ISO standard 9241-11:

- *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.
- *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*: The extent to which the user finds the use of the product acceptable, in specified context of use.

## 2.4 User and individual user differences

Nielsen (1993, p.42), argues the concept of “user”, should be defined to include everybody whose work is affected by the product in some way, including the users of the system's product or output even if they never see a single screen. However in this study context, the ISO 9241-11 definition of a user as a “person who interacts with the product”

was utilized. This was chosen because the end users' active role as producer in user-centered design, and as consumer of the system is relevant to this study.

According to Nielsen (1993, p.14), the two most important issues for usability are the users' task and their individual characteristics. It is therefore an important aspect of usability to know the user. Nielsen classified the three main dimensions about which users' experience differs: Experience with the systems, with computer in general, and with the task domain. Therefore, it is important to know the user. By knowing the users' work experience, educational level, age, or previous computer experience, it is possible to anticipate their learning difficulties with the systems, and to better set to appropriate limits for the complexity of the user interface. It is also important to know the amount of time the user will have available for learning, and whether they will have the opportunity to attending training courses; the interface must be much simpler if users are expected to use it with minimum training. The users' work environment and social context also need to be taken into consideration. In total, a great deal of information is required to characterize individual users. This information may be collected by questionnaires or interviews. Nielsen points out the convenience of observing and talking to users in their own working environment.

## **2.5 User-centered design**

There is no general definition of user-centered design. HCI "experts" interpret the concept in different ways. Again, although several definitions occur in literature review; the following two examples are presented; the first one is very clear and concise and the second one very passionate.

Preece (1994, p.722) defines user-centered design as *"an approach which views knowledge about users and their involvement in the design process as a central concern"*

Donald Norman (1986, p.61) writes, *"User-centered design emphasizes that the purpose of the system is to serve the user, not to user 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 rest of the systems."*

User participation in the design process avoids mismatches between the users' task and the developers' model of task. Users are no designer, so it is not reasonable to expect them to come up with design ideas from scratch. However, they are very good at reacting to concrete designs they do not like, or that will not work in practice. To benefit from user involvement, it is necessary to present these suggested system designs in form of prototypes (Nielsen, 1993).

Nielsen (1993, p.54), also states that user participation in the design process should not just consist of asking users what they want, since users often do not know what they want or need, or even what the possibilities are.

## 2.6 Key principles for user-centered system design

As early, as 1980s Gould and Lewis (1985 cited in Preece 1994) presented principles of user-centered design as follows:

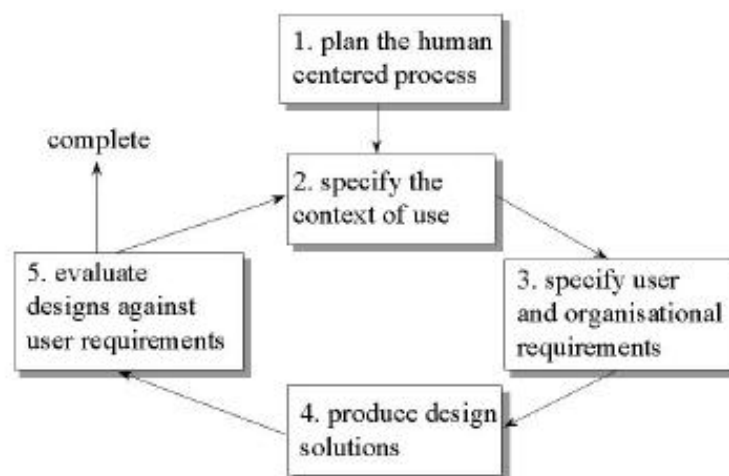
- To focus on users and their task early in the design process, including user guides, helping and ensuring that users' cognitive, social and attitudinal characteristics are understood and accommodated.
- To measure reactions by using prototype manuals, interfaces and other simulations of the system.
- To design iteratively because designer, no matter how good they are, cannot get everything right the first few times.

According to Preece et al (1994, p.504), user-centered system design has evolved from the original definition provided by Gould and Lewis to include almost any approach that emphasizes methods, techniques and representation for software systems, and which place the user at the core of the development process. These principles are still valid; the ISO principles are quite similar.

Below is the model of the ISO 13407:1999, which list the principles for human-centered design. It stipulates:

- The active involvement of users and a clear understanding of user and task requirements
- An appropriate allocation of function between users and technology
- The iteration of design solutions
- Multi-disciplinary design

ISO 13 407 Model Overview



More recently, Gulliksen et al (2003, p.10) presented a much more extensive list of principles, which includes the attitudes of the people involved in the system development process as the most essential piece in the complex puzzle of systems development. Here is the complete principle list provided by Gulliksen:

- *User focus* – the goals of the activity, the work domain or context of use, the user’s goals, tasks and needs should guide the development early on.
- *Active user involvement* – representative users should actively participate, early and continuously throughout the entire development process and system lifecycle.
- *Evolutionary systems development* – the systems development should be both iterative and incremental.
- *Simple design representations* – the design must be represented in such a way that users and all other stakeholders can easily understand it.
- *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 performed by effective multidisciplinary teams.
- *Usability champions* – 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 system design process must be specified, adapted and/or implemented locally in each organizations.
- A user-centered attitude should always be established.

User-centered design is a wide and interesting area, and the review presented here is by no means complete. Nevertheless, it should contain information that allows for a good understanding of HCI, usability, the user, and user-centered design. All these are relevant concepts in this study. Since the study is looking for the relationship between the RUP and the end-user in the development process, where the suitability of RUP being used as method for user-centered system is criticized. In the next chapter, a short historical review of the Rational Unified Process will be provided.

### 3. The Rational Unified Process

*This chapter will provide a general description of The Rational Unified Process (RUP), a brief history review, of The Rational Unified Process fundamentals, and the Software best practices. The Rational Unified Process has a very complex vocabulary and get a whole picture would make this study too long, and the reader can lose the picture about this paper. The chapter presents a very slight approach about what is RUP and how it works.*

#### 3.1 History Review

After leaving Ericsson in 1987, Ivar Jacobson founded Objectory AB. There he developed Objectory (“Objectory” is a abbreviation of “object factory”), which was a system that provided a better way to develop the lifecycle activities: requirements, analysis, design, implementation, and testing with the help of use cases. Rational Software Corporation acquired Objectory AB in 1995. Working in Rational Software Corporation Ivar Jacobson, Grady Booch and James Rumbaugh created the Unified Modeling language. Booch and Rumbaugh previously had released the Unified Method in October 1995. By 1998 the Rational Objectory Process was a process capable of supporting the entire software development lifecycle. In June of the same year Rational released the new version of the product with the name of Rational Unified Process, the new name was product of the unification of development approaches, using the Unified Modeling Languages, and unification of the work of many other methodologies. (Jacobsson et al 1998).

#### 3.2 Fundamentals of the Rational Unified Process

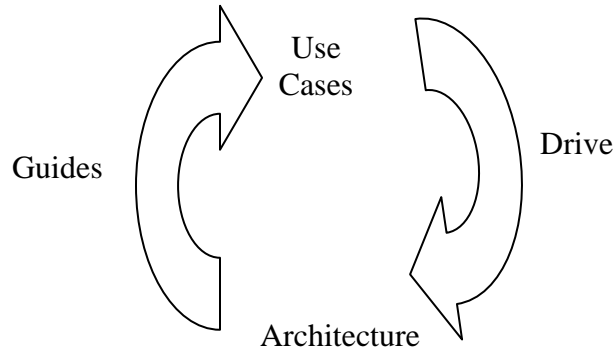
The Rational Unified Process (RUP) is different from other software development methods as the waterfall, in the way it performs the whole development chain. As it has been mentioned before, within RUP the common lifecycle (requirements, analysis, design, implementation, and test) is parted in activities called disciplines. These disciplines are performed in four phases: interception, elaboration, construction and transition (Strand, 2001). The Rational Unified Process has three fundamental characteristics: RUP is user-case driven, architecture centric, and is iterative and incremental.

##### 3.2.1 The Rational Unified Process is User-case Driven

In the unified process, the term *user* does not just denote the human being that uses the software. In the RUP *user* represents someone or something. The term used in rational to describe a human-machine interaction is *use-case*. Use-case captures functional requirements. All the use-cases together make up the use-case model, which describes the complete functionality of a system. Use cases specify the requirements of the system and drive the development process (Jacobsson et al 1998).

The Rational Unified Process is Architecture-Centric  
Architecture gives a picture of a whole system, which is necessary to control its development. Architecture involves usability, functionality, performance, resilience,

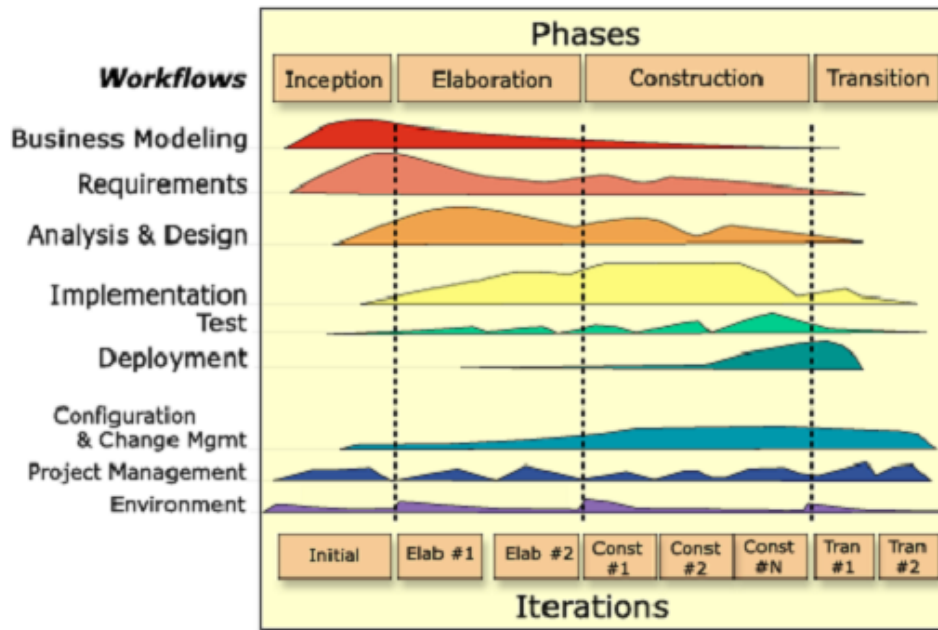
reuse and comprehensibility, all within economic and technological constraints. Architecture is a part of design; it is about making decisions regarding how the system will be built. There are interactions in RUP between the use-cases and the architecture, the use-cases drive the architecture to make the system provide functionality and a reasonable performance goal. The most significant use-cases and other inputs are used to implement the architectural baseline (Jacobsson et al 1998).



The use cases drive the development of the architecture, and the architecture guides which use cases can be realized (Jacobsson et al 1998).

### 3.2.2 The Rational Unified Process is Iterative and Incremental

Rational Unified Process has the strategy to split a project into several parts. The iterative and incremental process allows the balance among use-cases and the architecture of this balance is achieved gradually with the iterations. Iteration refers to steps in the workflow, and increments, as well as to growth in the product. RUP divides the process into four phases: Inception, Elaboration, Construction, and Transition. Within every phase, a number of iterations are done. The following figure shows the effort variation associated with the disciplines depending on which phase the project is executed (Jacobsson et al 1998).



The five workflows requirements, analysis, design, implementation, and test take place over the four phases, inception, elaboration, construction and transition (Jacobsson et al 1998 p.78).

### 3.3 Software Best Practices

According Kruchten (2000 p.24-25) RUP contains the called “Best practices”. These practices acquired this title because they are commonly used in successfully organizations. The best practices are to:

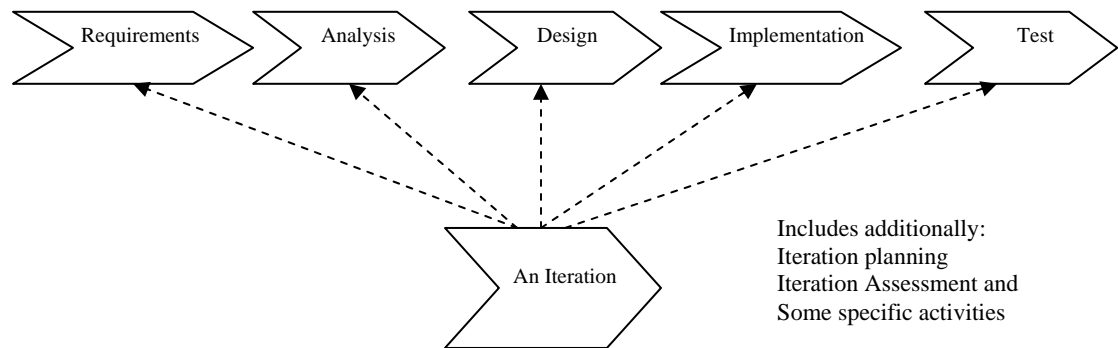
1. develop Software iteratively
2. manage Requirements
3. use component-based architectures
4. visually model software
5. continuously verify software quality
6. control changes to software

#### 3.3.1 to develop software iteratively

The development of software is split into several manageable steps. Thus only a little project must be designed, implemented, integrated, and tested. Once a result is deemed satisfactory, the next step is taken. This process is made iteratively until all the steps are taken. When all the steps are completed, the product can be delivered to the end-user. Iterations help in attaining better control over system development. Another important benefit of the iteration is that it contributes to stronger architecture. Because it is tested from the beginning, changes are relatively less expensive (Jacobsson et al 1998).

Even if a system does not have full functionality, the end-user and stakeholders can still test the preliminary release. They may figure out how the final product might look and suggest changes or point out if the need for additional or change requirements on the system (Jacobsson et al 1998).





Every iteration passes through the five core workflows. It is initiated with a planning activity and finished with assessment. (Jacobsson et al 1998 p.32).

### 3.3.2 To manage Requirements

Requirement management is a methodical way to maintain a system, and it includes documentation, communicating and managing the changing requirements of the systems. Requirement management contributes better control of complex project and increases the understanding of the system. Requirement Management helps to involve the end-user early in the project and contributes to the common understanding of all the stakeholders. (Kruchten, 2000).

### 3.3.3 TO use component-based architectures

The focus under the earliest iterations is to design a robust architecture that supports the actual and future requirements of a system. RUP provides a systematic way to design, develop, and validate the architecture. RUP has component-based architecture, with modularity that offers several advantages; allowing for identification, isolation, design, development, and testing of well-formed components (Kruchten, 2000).

### 3.3.4 To visually model software

Rational Unified Process is complemented by the Unified Modeling Language (UML) that is a graphical language for visualizing, specifying, constructing and documenting software artifacts (Kruchten, 2000).

### 3.3.5 To continuously verify software quality

The Rational Unified Process has focus on product and process quality. The first, product quality, pertains to the quality in the system and all the elements involved in like components, subsystems, architectures, and so on. The second, process quality, regards aspirations to an acceptable measure of quality criteria that is done under the whole development process. RUP plays a role, but all members in the development team are involved in the quality assurance (Kruchten, 2000).

### 3.3.6 To control changes to software

Changes are frequent in system development. RUP uses the control change to keep control of changes and ensure that everything and everyone is coordinated. It is used as a knowledge repository, to keep track of defects, misunderstandings, and project commitments and activities associated with specific artifacts and releases (Kruchten, 2000).

### 3.4 Usability Design as a Discipline in Rational unified process

The usability design discipline (UDd) is an example of how usability design can be integrated in a commercial systems development process, the Rational Unified Process. The UDd was developed as a plug-in to the RUP and does not include any advice on how to modify the other disciplines to better include and make use of the artefacts in UDd. The UDd complements the RUP. It promotes and facilitates usability work within the software development process by means of a user-centered approach. (Göransson & Gulliksen, 2003c).

Gulliksen et al (2003, p.22) state 'the main activities in the UDd are:

- Usability design plan involves detailed planning of the user-centered activities and user involvement. The plan is refined in each iteration.
- Conduct user studies – i.e. interviews, observations, workshops, etc- to understand the potential users of the system, their needs and the context of use. Specify the goals for the system, the design criteria and usability goals.
- Perform competitor analysis to get inspiration from similar state-of-the-art systems of products.
- Conceptual design describes the overall structure of the user interfaces. Use brainstorming, usage scenarios, sketches and mock-ups to illustrate potential high-level design solution.
- Iteration design outlines details in the conceptual design illustrating potential user interactions (navigation, information and functionality), simulating the real system.
- Detailed design includes fine-tuning all the design details in the GUI.
- Develop user assistance, a parallel design activity focusing on on-line help systems, manuals and user training material. Also covers new work procedures.
- Monitor usability work- i.e. handling late change requests and ad hoc decisions in the construction phase.
- Usability evaluation of design solutions against the usability goals. Evaluation should be performed on all design suggestions and solutions from preliminary sketches to fully interactive prototypes and the final systems.'

## 4. End-user and Rational Unified Process Interactions

*The previous chapters have presented the major sources of this study: User-centered design as a part of Human computer Interaction (HCI) and the Rational Unified Process (RUP). On base of the review literature, there exist several relationships between the end-user and RUP, which follow. It is not the intent to claim that these are the only possible relationships. These relationships are distinguished with bold letters*

**R1: Interactive systems cannot be completely specified from the beginning of the software lifecycle. Use-cases may provide a concise method for modeling the functional requirements, but they are not necessarily good enough to catch the non-functional requirement. As RUP is use-case based, lacks ability to handle the non-functional requirement.** This deficiency of RUP for capturing non-functional requirements has been pointed repeatedly by usability experts: The RUP is inherently engineering-oriented. RUP development processes prescribe that all documentation should be modeled on formal notations, which means that the activities and rich descriptions of users are substituted by actors, logical relations and flows of events. Descriptions of the context of use or knowledge about the design space cannot be accommodated in this method of capturing information (Göransson, 2004a). Rational Unified Process is basically oriented towards software development, and is not designed to cover the user needs (Gulliksen, & Göransson, 2002). There are usability integration features within the RUP development approach, but it is only used to describe the technology aspects of the user's interaction with the system, so it does not model user-centered design. (Boersma, 2003).

**R2: Prototypes provide realism and allow the designer to evaluate the system's impact with the end-user. Within RUP, method prototypes are made by the developer based on the use-cases, and they provide a vision of the developers that is not always the vision and need of end-users.** Prototypes must integrate both functional and non-functional requirements. In order to reach this goal they have to be built with end-user collaboration. Prototypes must be constructed according user-centered principles: Prototypes should be used to visualize and evaluate ideas and design solutions in cooperation with the end users (Göransson, 2004). Prototypes should provide realism and allow the designer to evaluate the system's impact with the end-user (Dix et al 1997). According to Donald (1986, p.61) from the point of view of the user, the interface is the system. Let the requirements for the interaction drive the technology. Doing user-centered designs involve to start with the needs of the end-user, and these are the ones should to be presented in the prototypes.

**R3: End-user interactions involve the end-user observing and evaluating throughout the whole system development process. End-user interaction can be limited because of time and cost, especially in RUP, which employs a primarily iterative process.** The adoption of usability design which is oriented towards user-centered systems, has a lot to do with budget and the politics of return-

on-investment (ROI) for a company. It involves activities that usually are not part of a systems development process, e.g. studying people work, prototyping with users, trying out design, evaluating with users and iterating solutions (Göransson, 2004). Centered design is expensive a project take a longer time and the number of people involve is larger. These reasons limit the popularity of this kind of development. Participative design has not been widely practiced, perhaps due to the time and cost that this type of project involves (Dix et al 1997). However, usability experts argue that fixing errors is quite more expensive. User-centered design reduces cost of training and documentation and improves the employee's productivity because usability reduces the cost for absence due to illness (Göransson, 2004).

**R4: “The end-user has difficulties to explain their needs.” This is frequent opinion of the expert designer and developer. While this may be true, a closer observation from the designer of end-user tasks can help lead to a better understanding of user-needs. RUP framework does not provide the tools for documenting a closer observation of the end-user and in this way, to identify user needs.** The computer can change the nature of human work from complex to routine and vice versa. When designing a computer system, it is important to consider how the work is currently carried out and how the resulting system will change it (Preece, 1994). System design should be grounded in the work of future users. Interviews combined with observations should be performed, where the goal is to construct a rich picture of the actual work situation: roles, responsibilities, problems with the work and existing tools. This gives a better idea of the *real* work situation, not just what people say they do (Löwgren & Stolterman, 2005). Users often have problems asking for solutions that really meet their needs. User-centeredness is often seen as an option that must be specifically required by the end-user, but the end-user does not always have the knowledge needed to ask for this kind system. A closer observation from the developer to the end-user tasks could help to a better understanding of user's needs (Jacobsson et al 1998). There are other attitude problems. Sometimes the end-user is perceived as naive and narrow-minded. A lack of a common vocabulary may lead to misunderstanding between the end-user and the developer. Persons with different professional backgrounds frequently have problems to understand each other within a discussion related to professional activities. For example, the use of terms and concepts to define and describe daily work may confuse the end-user (Boivie, 2005). As was pointed before, RUP relies on use-cases and UML as support language. Constantine (2001) argues the inconvenience of RUP in finding the needs of the end-users. He says that UML used in the RUP framework provide neither diagram type nor notation for representing user interface in either abstract or realistic form. As it has been mentioned before, a very important approach to interactive system design relies on a clear understanding, early on in the design, of the tasks that the user will need to execute. In summary, RUP does not realize the problem of tasks the end-user will perform are being often only known by the end-user. In addition, RUP has deficient use of notations and techniques which support the user's perspective of the interactive systems (Dix et al 1997).

**R5: Developers are often so busy learning and developing skills with the emerging technologies, and techniques that are showing up everyday, that they don't have much left to take care of end-user needs; Maintaining the documentation in RUP demands a lot of extra work and , which could be better spent in the user-centered design.** RUP puts a strong emphasis on documentation (Kruchten, 2000). RUP demands extra production and maintenance of documentation; this task takes a lot of time and effort that could be used in the development process (Brown, 2003). People who work in the development area usually have a technical background, and it is comprehensible they may be more interested in technology and program codes. The race to learn and use new techniques is very difficult today. Competitive pressures and rapid technological changes are forcing computer-based system designers toward shorter product development life cycles. (Jacobsson et al 1998). But consider the quite interesting but a bit extreme, Ethic and Moral approach made by Göransson (2004). He argues that, developers have power over the user and they must use this power in an ethical fashion. Ethics are also an issue when a system designer knows that they can do better, but neglect to do so because it takes extra effort.

**R6: Rational Unified Process is oriented to software development and not designed to cover the user needs.** RUP is essentially a system development method designed to steer projects, produce a rich documentation and to deliver results on time. It does this with an object-oriented view and strong focus in the system architecture (Kruchten, 2000). This strong orientation towards system architecture and the impossibility of gathering rich information about the regular user's task in the use-cases, are the arguments used by RUP critics against RUP's being used as user-centered design (Gulliksen, & Göransson, 2002). As has been mentioned in earlier chapters the RUP performs throughout the whole development chain; requirements, analysis, design, implementation, and test which are called disciplines. These disciplines are performed in four phases: inception, elaboration, construction and transition. End-user participation is mainly at the inception and the transition phases (finding out requirements and system delivering). In the inception phase, use-cases are elaborated on the basis of system functional requirements. The development is based in the initial design and the developers make the effort to achieve the goal. In practice, designers do not find out all of the requirements for a system before they begin. The point is that if not all the requirements for an interactive system can be determined from the start; then the uses-cases that support the whole development process are based on functional requirements. The end-user interactions with the systems are limited to the views-design, due principally to the deficiency in use of notations and techniques supporting the user's perspective. It is very difficult to predict the cognitive demands that an abstract design would require of the future user if the notation does not reflect the kind of information the user would use in order to interact with the system (Dix, et al 1997).

## 5. Research Methodology Framework

*In this chapter, the research methodology framework will be presented, the qualitative approach, research strategy, the generating relationship as hypotheses, the data collections, the questionnaire for the interviews, as well as the data analysis, validity and that it ends with ethics.*

### 5.1 Qualitative approach

Görasson (2004, p.48) states that ‘*Rational Unified Process RUP is the predominant development process in Sweden. It seems that those who do not use the RUP use some in-house development process or no explicit process at all.*’ The Rational Unified Process RUP is a well-known method for system development and there is a considerable amount of literature about it (Kruchten, 2000). On the other hand, there is also diverse literature that deals with the human computer interaction, and uses the term “end-user”, frequently. However, the connection between RUP and the end-user is not documented specifically. The relationship between RUP and the end-user may be considered as partially unexplored, and suited to study by qualitative approach.

By choosing a qualitative approach, the chance increases to receive a broader, more holistic result. Furthermore, an advantage with qualitative research is that the focus does not need to be completely clear when beginning the investigation. When the research subject is very broad at the beginning of research, and the research question not very clear, a qualitative focus allows the study to proceed without a determinate problem question (Bryman, 2002). By not having a hypothesis to verify or falsify from the beginning, the possibilities to find a greater number of relationships between RUP and the end-user increase.

### 5.2 Research strategy

An explorative study was chosen on the basis of lack of information related to the end-user and its interaction with RUP methodology in the development process. An exploratory study has the feature of answering the “what questions,” which matches with the purpose of this study. Developing hypotheses and propositions for further inquiry is another of the characteristics of an explorative study (Yin, 2003). These reasons make the explorative study suitable to achieve the goal of this study.

### 5.3 Generating Relationships (Hypothesis)

Relationships between RUP and the end-user were generated at the beginning based in the existent literature about the main areas: RUP and user-centered design. These relationships were verified with empirical data as outcomes of the real life experience obtained in a series of interviews. The verification process will be presented in detail later in this chapter.

Relationships are similar to hypotheses called propositions in qualitative research. (Bryman, 2002). From the literature, a common concern was taken about RUP being used as method in user-centered design and put together. On other hand tried to figure it out

what you do need to carry out a user-centered within a RUP development framework. On basis of this collected information, several relationships were tailored. These relationships in turn form the base of the interview questionnaire.

## **5.4 Data collection**

Data collection can be described as interrelated activities aimed at gathering good information to answer to research question. This includes finding people or places to study, gaining access, designing the strategy for the purposeful sampling of individual or places, and data collecting (Creswell, 1998). For this study, the activities that are aimed towards collecting data are the literature review and purposeful sampling as criteria for selecting an interview subject.

### **5.4.1 Literature review**

Review of literature allows for collection of information about its structure, process and relationships, thus increasing the familiarity of the researcher with the research object (Sarantakos, 1998). The literatures used in this study are mainly found at the libraries from Lund and Malmö Universities. The articles used are found on the internet mainly from Uppsala University and companies involved in the main sources areas: Rational Unified Process and usability design (user-centered design). The literature review forms a very important part of this study. It helps to create a base for the generation of the relationships (hypotheses) which, are they key of this study.

### **5.4.2 Purposeful sampling**

The purposeful selection of participants represents a key decision in a qualitative study; a researcher chooses participants based on their ability to contribute to an evolving theory (Creswell, 1998). Decision about where to conduct research and who to include is calling “sampling,” and it is an important part in the research method. Purposeful sampling or criterion-based selection is important because allows achieve representative individuals suitable for the study (Maxwell, 2005). The appropriate participants are organizations involved in the software development using RUP as a development method and companies engaged in the user-centered issue as well. Finding the suitable participants who are willing to be involved in the study has been the most difficult part of this study. Qualitative inquiry typically focuses in depth on relatively small samples, or even a single case. The logic and power of purposeful sampling lie, in selecting information-rich cases for study in-depth (Patton, 1990). It was difficult to find a developer with a good understanding of RUP and even harder to find user-centered designers with experience in systems developed with the RUP method. Nevertheless, the respondents who took part in the interviews ended up giving very meaningful information from a strong relation with both RUP as the user-centered development.

### 5.4.3 Participating Companies

All the companies that participating in this study use the Rational Unified Process as development method and are interested in usability design. Finding companies that matched these criteria was not an easy thing since RUP, despite its popularity, is used mainly in very large companies. Most of these companies run their own private student-programs, and are not available to help students from outside. The companies which finally participated in the study are nonetheless very representative because they are deeply involved with RUP as a usability issue.

The companies, which participated in this research, belong to different areas and they are describing in more detail later chapter. A brief list follows here:

- **Volvo Information Technology AB.** According to information presented on its website, Volvo provides solutions for all areas of the industrial process, leading users of IT in the automotive industry. Volvo Information Technology has been awarded “World-beating cost efficiency and quality” in 2001 by the Compass Consulting which is an international consult company.
- **Guide Redina AB,** works with user-centered developments. Guide Redina, claims to have built its own discipline into the RUP.
- **Tempogon AB,** is a consultancy company. According the information provided on its website, Tempogon created a model that takes care of technique, people and process in a harmonic way.

### 5.4.4 Interviews

This study aims to find the interaction of the end-user with the Rational Unified Process in the system development. In order to achieve this goal, interview seemed the most appropriate method. Interviews are often an effective and valid way of understanding someone’s perspective and getting worthwhile data (Maxwell, 2005). The study combines one face-to-face and two telephone interviews. All interviews were taped. After being transcribed and concluded, they were sent to the interviewees so they could confirm their answers.

#### 5.4.4.1 Face-to-face interviews

Personal interviews are very useful in order to avoid misunderstanding on the side of the researcher. This kind of interview gives the interviewee the possibility to clarify the questions, as well as their answers (Creswell, 1998). This kind of interview can be more enjoyable because the interviewee and interviewer feel more relaxed and less concerned about the duration of the interview.

#### 5.4.4.2 Telephone interviews

A telephone interview provides the best source of information when the researcher does not have direct access to individuals (Creswell, 1998). According to Bryman (2002, p.111-112) the telephone interview can be seen as a considerably cheaper and less time



consuming alternative since the researcher does not need to spend time and money on traveling to the respondents. As was explained before, finding people who will participate in the study is not an easy task. The “utopia” of having an interviewee who is interested in the issue is the ideal situation, but is not always reachable. Since the companies that accepted to take part in this study were located in different regions of Sweden, telephone interviews were a good alternative for collecting data. The combination of IP-telephony (Skype), PrettyMay, a software which allows to record the conversations from Skype and Express Scribe a software were implemented to help to carry out the transcription process more easily.

#### 5.4.4.3 Semi-structured interview

Semi-structured interviews to collect data provide detailed qualitative data and allow for elaboration and follow-up questions. A questionnaire was prepared upon the framework proportioned by the existing literature (Creswell, 1998). A qualitative approach increases the probabilities for receiving a broader result. This type of interview gives the researcher the opportunity to ask further questions based on the answers that seem to be essential, or that can contribute to a better understanding of the issue (Bryman, 2002). The respondents usually have different backgrounds and experiences, and so the depth of the interview may vary from a respondent to another. A semi-structured interview offers the opportunity to collect more data because of the possibility to ask questions which were not contemplated at the beginning of the interview. In this study specifically, there are two types of interview subjects: Developers and usability designers, hence a semi-structured interview allows fitting of the questionnaire according to the type of interviewee.

### 5.5 Questionnaire

*As was explained before, the relationships are the base of the interview questionnaire. Some questions correspond to several relationships. The questionnaire follows:*

Relationships	Questions
<p>These questions have as purpose of assessing the interviewee competence in the area of the study.</p>	<ol style="list-style-type: none"> <li>1. Would you tell me little about your work?</li> <li>2. How long have you work with The Rational Unified Process?</li> <li>3. Which roles do you play in the system development with RUP?</li> </ol>
<p>R1: Interactive systems cannot be completely specified from the beginning of the lifecycle Use-cases provide a concise method for modeling the functional requirements but not necessarily, they are good enough to catch the non-functional requirement. RUP is use-case based since it has a lack to handle the non-functional requirements.</p>	<ol style="list-style-type: none"> <li>4. What is the major contribution of RUP as development method?</li> <li>5. RUP is a used-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?</li> <li>6. Who is the responsible for finding out the requirements? There is a specific person for this role?</li> <li>7. Does the end-user participate in the building of uses-</li> </ol>

	cases?
R2: The prototypes provide realism and allow the designer to evaluate the system's impact with the end-user. Within RUP, method prototypes are made by the developer based in the use-cases, since they provide the vision of the developers that is no always the vision and needs of end-users.	<p>8. Is there, in the RUP, the possibility of end-users observation the in their own work environments?</p> <p>9. Do you build prototypes for the end-user consideration before the start with the elaboration phase?</p>
R3: The end-user interaction involve end-user observing and evaluating under the whole system development. The end-user interaction can be limited because of time and cost especially in RUP that works with iterative process.	<p>10. Do you think the end-user is rewarded by RUP being used as development method?</p> <p>11. The participation of the end-users in the system development is mostly in the interception (requirement specification) and the transition (delivering and training). Do you think that RUP needs to be modified in that way it manages the requirements?</p> <p>12. What do you think about creating a new discipline, which takes care of the user-centered issue?</p>
R4: "End-user has difficulties to explain their needs" this is frequent approach of expert design and developer. A closer observation from the designer to the end-user tasks could help to a better understanding of user-needs. RUP framework does no provide the tools for documenting a closer observation to the end-user and in this way caching the needs of the users.	<p>8. Is there, in the RUP, the possibility of end-users observation the in their own work environments?</p> <p>9. Do you build prototypes for the end-user consideration before the start with the elaboration phase? Do you have time left to take care of the end-user needs?</p> <p>15. Do users ask for user-centred design?</p> <p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are too busy learning new techniques and developing skills, with the new technologies that show up everyday that they don't have.....</p>
R5: Developers are to busy learning and developing skills with emerging technologies, and techniques that are showing up everyday, that they don't have much more left to take care of the end-user needs; Keeping the documentation in RUP demands a lot of extra work and time that could be used in the user-centered design.	<p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have much more time left to take care of the end-user needs?</p>
R6: Rational unified process is oriented to software development it is not designed to cover the user needs.	<p>14. How often do you need to adapt to an existent architecture?</p> <p>15. Do users ask for user-centred design?</p> <p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have</p>

## 5.6 Data analysis

A quick analysis directly after the interviews is useful so as not to forget what the respondent said during the interview. An additional advantage is that the researcher may form new ideas and possible new questions for future interviews (Kvale, 1996). Miles and Huberman (1994) also recommended early analysis in order to get the most out of interviews.

By listening the whole interview before transcription, it gives the opportunity to analyze, take notes and memos, develop coding categories, and contextualize relationships on what is seen or heard in the interview (Maxwell, 2005). The analysis this study was divided in three phases: data reduction, data display, and conclusion-drawing and verification, as Miles and Huberman (1994) have recommended. Data reduction refers to the process of selecting, focusing, simplifying, abstracting, and transforming the data that appear in the transcriptions. Data display is described as an organized, compressed assembly of information that permits conclusion drawing and action. Interviewers write into the suitable tables or matrix for get a better understanding of information and help to establish relations easily (Miles & Huberman, 1994). In this study, the transcription was not verbatim but the essence of the taped interviews. The transcribed material was sent to the respondents so they could verify their answers. The answers were then coded by grouping them into the different relationship in order to make it possible to analyze and verify the relationships and draw conclusions.

### 5.6.1 Coding

Coding is a process by which a researcher forms categories of information about the phenomenon being studied by segmenting information. These initial coding became after in the relationships (hypotheses) (Creswell, 1998). According to literature, the basis for coding was generated by classifying and grouping the different answers into the different relationships.

From relationships between the RUP and the end-user, generated by the literature review the foundation for coding was created. The foundation was used to classify, and group the different answers into the relationships.

Foundation for Coding	Answers for Questions
<p>These questions have as purpose of assessing the interviewee competence in the area of the study. In order to find a purposeful sampling the interviewee has to be a developer in knowledge in RUP or a usability designer.</p>	<ol style="list-style-type: none"> <li>1. Would you tell me little about your work?</li> <li>2. How long have you work with The Rational Unified Process?</li> <li>3. Which roles do you play in the system development with RUP?</li> </ol>
<p>R1: Interactive systems cannot be completely specified from the beginning of the lifecycle</p> <p>Use-cases provide a concise method for modeling the functional requirements but not necessarily, they are</p>	<ol style="list-style-type: none"> <li>4. What is the major contribution of RUP as development method?</li> <li>5. RUP is a used-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?</li> </ol>

<p>good enough to catch the non-functional requirement.</p> <p>RUP is use-case based since it has a lack to handle the non-functional requirements.</p> <p>Within RUP a user is not necessarily a person, and then the end-user has a poor participation in the developer that uses RUP as development method.</p>	<p>6. Who is the responsible for finding out the requirements? There is a specific person for this role?</p> <p>7. Does the end-user participate in the building of use-cases?</p>
<p>R2: The prototypes provide realism and allow the designer to evaluate the system's impact with the end-user.</p> <p>End-user provides valuable information upon the prototypes observations.</p> <p>Within RUP, method prototypes are made by the developer based in the use-cases, since they provide the vision of the developers that is not always the vision and needs of end-users.</p>	<p>8. Is there, in the RUP, the possibility of end-users observation the in their own work environments?</p> <p>9. Do you build prototypes for the end-user consideration before the start with the elaboration phase?</p>
<p>R3: The end-user interaction involve end-user observing and evaluating under the whole system development.</p> <p>The end-user interaction can be limited because of time and cost especially in RUP that works with iterative process.</p>	<p>10. Do you think the end-user is rewarded by RUP being used as development method?</p> <p>11. The participation of the end-users in the system development is mostly in the interception (requirement specification) and the transition (delivering and training). Do you think that RUP needs to be modified in that way it manages the requirements?</p> <p>12. What do you think about creating a new discipline, which takes care of the user-centered issue?</p>
<p>R4: "End-user has difficulties to explain their needs" this is frequent approach of expert design and developer.</p> <p>A closer observation from the designer to the end-user tasks could help to a better understanding of user-needs.</p> <p>User centered systems are oriented to fulfill end-user needs in their diary work.</p> <p>RUP framework does no provide the tools for documenting a closer observation to the end-user and in this way caching the needs of the users.</p>	<p>8. Is there, in the RUP, the possibility of end-users observation the in their own work environments?</p> <p>9. Do you build prototypes for the end-user consideration before the start with the elaboration phase? Do you have time left to take care of the end-user needs?</p> <p>15. Do users ask for user-centred design?</p> <p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are too busy learning new techniques and developing skills, with the new technologies that show up everyday that they don't have.....</p>
<p>R5: Developers are to busy learning and developing skills with emerging technologies, and techniques that are showing up everyday, that they don't have much more left to take care of the end-user needs.</p> <p>Keeping the documentation in RUP demands a lot of extra work and time that could be used in the user-centered design.</p>	<p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have much more time left to take care of the end-user needs?</p>

<p>RUP need a role oriented to keep the user-centered point of view under the whole process in order to created user-centered systems.</p>	
<p>R6: Rational unified process is oriented to software development it is not designed to cover the user needs.</p>	<p>14. How often do you need to adapt to an existent architecture?</p> <p>15. Do users ask for user-centred design?</p> <p>16. Is the user-centred design part of the standard requirement, even if the users do not ask for it?</p> <p>17. Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have</p>

With the data fragmentation, the comparison between interviews could be done easily. This strategy helps to establish patterns and relationships; however, this process is accused of fragmenting and data manipulation. (Bryman, 2002). In the open coding phase, the researcher examines the data, looking forward for salient categories of information. Using the constant comparative approach, the researcher tries to find instances that represent the category this process, the researcher keeps doing this process until the new information obtained does not provide valuable information the category (Creswell, 1998). In this study, the questions were sorted according the relationships they were testing. The relationships then work as a base for the questions that have been used during the interviews as well as base for evaluating and presenting the result of the study.

## 5.7 Validity

The gap between the original research strategy and the research practice can arise because of the impossibility or extreme difficulty of obtaining an available sampling (Bryman, 2002). A serious limitation in this study could be the absence of the end-user point of view since the study is about the RUP and its relationship with the end-user. Nevertheless, a usability expert in this study represents the end-user. Other limitations could be the lack participation of companies. A larger number of companies could have contributed to a more reliable result. Another possible limitation with this study is the abundance literature where RUP is criticized for it's orientation to the development process instead of to the user, while failing to relate the two. This last limitation, however, can be seen as a positive one in some respects, as it creates the space needed to carry on this study.

## 5.8 Ethics

Informed consent involving the voluntary participation of the individual was obtained by explaining the research purpose as all possible outcomes that could be associated with the participation of the study. The intent was to eliminate suspiciousness and uncertainty that can exist against the research (kvale, 1997). A clear explication about the research topic and the purpose of the study was always the first step in the relationship with the subject interviews. Upon asking for an interview in a formal letter, confidentiality was guaranteed in order to gain the willing reply of the respondents. Moreover, before

starting the interview the respondents were informed that the interview would be recorded, transcribed and sent them. Thus, they could confirm and correct their answers in order to make sure the participation cannot harm the respondent in any way.

## 6. Description of the Companies

*The presentation and description of the companies follows here, together with a short description of how they are related to the RUP and usability design.*

### 6.1 Volvo Information Technology AB

Volvo Information Technology AB, introduces itself as a solution provider for all areas of the industrial process. According information presented in its website Volvo is leading user of IT in the automotive industry. Among its clients, include the Volvo Group, Ford-owned Volvo Car Corporation, and other industrial companies. Volvo Information Technology has been awarded with the “World-beating cost efficiency and quality” in 2001. (<http://www.volvo.com/volvoit/global/en-gb/> 2007).

Rational unified Process, RUP is the standard system development process used at Volvo IT. The support for design and usability in standard RUP is weak and difficult to locate. In order to correct, this weakness Volvo IT has implemented the Design and Usability discipline developed by Guide Redina and Uppsala University.

### 6.2 Guide Redina AB,

Guide Redina AB, introduces itself as a consultancy practicing User-centered System Design. Enea Redina claims to provide full service in the system development process – from pre-study, analysis and design to implementation, test, and deployment. Its clients are small to medium enterprises especially within health care, telecommunication and the public sector. (<http://www.redina.se/> 2007).

Enea Redina, in cooperation with Uppsala University, has proposed a new discipline into the RUP. The aim is to make RUP more user-centered. According to Enea Redina, the new discipline extracts the best practices from user-centered design. The usability design discipline helps projects to focus on usability and the users throughout the system development lifecycle.

### 6.3 Tempogon AB

Tempogon AB, is a consult company operating since 1999. According with the information provided in its website, Tempogon AB, started as franchise organization and still this is the major idea. Around the franchise, it develops solutions for its clients upon existent and well-tested concepts.

Tempogon claims to have a very close cooperation with IBM rationale by using IBM’s tools in the areas of development, advertising, license handle, and maintenance. Tempogon AB, provides IT-development principally with the Rational Unified Process (RUP) and Unified Markup Language (UML). (<http://www.tempogon.se> 2007).

## 7. Presentation of the results of the empirical data

*The results from the study are presented as grouped by the different relationships. The complete answers from each respondent are attached as appendixes. Three interviews were carried out, two of the interviewees are with developers and the other one with a usability designer.*

**R1: interactive systems cannot be completely specified from the beginning of the lifecycle. Use-cases provide a concise method for modeling the functional requirements but not necessarily, they are good enough to catch the non-functional requirement. RUP is use-case based since it has a lack to handle the non-functional requirements.** Two of the respondents agree that finding out the requirements to build the use-cases how take around 20 and 25 percent of the whole project depending on how large is the project. However, one of the developer respondents argued that one does not “*define use-cases at the beginning as in the water-fall method, but use-cases are built under the whole process, of course at the inception phase [one works] more, but it is difficult to say how much time*” (Appendix 3). All the respondents agree that is hard to get all the requirements from the beginning and the use-cases need to be changed frequently. The impact of the change varies depending how detailed are the use-cases. One of the main benefits of RUP is allowing for management of the changes that are always present in projects. The individual who is responsible for building the use-cases is a systems analyst. One of the developer respondents pointed that the disadvantage with the user-cases is they can be very abstract. He said that sometimes use-cases represent the whole process, and the end-user is concerned about supplying information about in his/her every day work and not thinking about the functions that could be improved. Then, the analyst needs to have a strong usability perspective to find out what is the best way to improve the future system.

**R2: The prototypes provide realism and allow the designer to evaluate the system’s impact with the end-user. Within RUP method, prototypes are made by the developer based in the use-cases, since they provide the vision of the developers that is no always the vision and needs of end-users.** All of the respondents use power point as the most common tools to build prototypes. Real prototypes are not always present in the project. Prototypes are built depending on how large the project is. All respondents agree the prototypes help finding out how the user wants to work, to find the best way to surf and get information. One developer respondent points out that “*the use-cases are suitable to illustrate functionality but you need prototypes and complete examples to show how the systems will be, what vision the analyst has...*” (Appendix 3).

**R3: The end-user interaction involve end-user observing and evaluating under the whole system development. The end-user interaction can be limited because of time and cost, especially in RUP that works with iterative process.** All the respondents agree RUP allows a closer participation of the end-user in the system development process. However, each one has a different way to defend the end-user’s participation. A developer respondent alleges that the system analyst identifies the actors and discusses their process with the end-users point of view. He says too, the end-user has participation, directly or indirectly in the approbation of the one phase before to continue with the next phase. The other developer respondent argues, “*We try to involve the end-user in the whole project. Of*



course, they are more involved at the beginning and the end; we try to do tests after all the iterations. They come to the test that we carrying out and participate in the test” The (Appendix 2). The Usability systems designer respondent points out that the end-user participation is also relevant with the prototypes.

**R4: “End-user has difficulties to explain their needs” this is frequent approach of expert design and developer. A closer observation from the designer to the end-user tasks could help to a better understanding of user-needs. RUP framework does no provide the tools for documenting a closer observation to the end-user and in this way caching the needs of the users.**

The usability designer respondent says, *“In almost all projects that I work in, I need to go to the user’s place and see when they work in their ordinary environment. It is part of my work. We need to observe them for understanding their job”* (Appendix 4). The developer respondents focus their attention to the system functionality. They contact the end-users and collect information; they figure out the requirements and later discuss them with the end-user. One of the developer respondents says, *“Most of the users want to continue doing their work in the same way. I mean they get used to work with the old system or how they work now so the analyst has to be good, to show they [sic] the vision of the changes and how these changes could make the process easier”* (Appendix 3). The other developer respondent argues, *“I think we should focus on the functionality because we are system development and the usability requirement can be taken later”* (Appendix 2). *“On other hand sometimes, it can be a disadvantage because RUP could produce too much documentation and we expend a lot of time maintaining the documentation instead of working in the system’s functionality. However, in the end the system you get is a very fine documented system”* (Appendix 4).

**R5: Developers are to busy learning and developing skills with emerging technologies, and techniques that are showing up everyday, that they don’t have much more left to take care of the end-user needs; Keeping the documentation in RUP demands a lot of extra work and time that could be used in the user-centered design.**

There are different points of view among the respondents. The developer respondents agree that developers have the tendency to be more technique oriented, and they say that most the people who work in IT are interested in new technique and programs codes than they are in usability. They agree too about the fact that the usability issues need to be considered in the development process. The user-centre systems designer respondent argues: *“I am not the best one to assess that because I am not a developer, but I think usability must to be lift in the development process because it does not matter how high technology we use in the development, if the person who works every day with the system is not happy with them. People sometimes don’t realize how annoying it would be to work or be obligated to work with a system that really does not help you or even worse. The work is more difficult. I have seen employees that just stop using the system”* (Appendix 4).

**R6: Rational unified process is basically oriented to software development it is not designed to cover the user needs.**

All the respondents agree the Rational Unified Process offers many benefits in the development process. All of them also agree that RUP offers many facilities for managing changes strategically. One of the developer respondents acknowledges that RUP collects experience and puts them into a book. He states that RUP allows you to organize a project in a more efficient way by parting a large project to several smaller and more controllable projects. Another of the developer respondent points out that *“when an organization is so large it has to find some way to steer it in the right direction. With RUP we have the possibility of take a better control of the whole project”* (Appendix 2). All respondents agree too, that use-cases are useful to find out requirements and they are a guideline when you need to make decision. However, they also agree that RUP has a deficiency and a poor support for the specification of non-functional requirements. Volvo IT, which already has implemented a usability discipline in RUP, explains its motive, admitting that *“the support for design and usability in standard RUP is weak, difficult to locate and, to some extent, misleading”* (Appendix 2).

Finally the user-centre systems designer points of that RUP, allows for communication among project members. RUP works very well as a communication language. Documentation is also a good contribution because everyone involved in the project knows what is going on, but sometimes it can be too much because we expend a lot of time maintaining the documentation instead of working in the system’s functionality.

## 8. Verifying of the relationships

*The different relationships have here been analyzed and compared with the research from literature. The purpose is to verify whether proposed relationships have solid foundations or not.*

**R1: interactive systems cannot be completely specified from the beginning of the lifecycle. Use-cases provide a concise method for modeling the functional requirements but not necessarily, they are good enough to catch the non-functional requirement. RUP is use-case based since it has a lack to handle the non-functional requirements.**

This relationship has been partially verified in this study. The information gathered from respondents accords with literature in the RUP seems have a deficiency for capturing non-functional requirements. As one respondent has indicates (Appendix 3). The capture of relevant data and requirements is a process that begins in the first development stage, and continues under the whole development process. This fact is frequently remarked upon in the consulted literature, as well as in chapter 2. After literature review and comparison of the respondents answers use-cases could be considered as an excellent tool to describe function. Nevertheless, at the same time they seem to be much too abstract to catch the rich description of the everyday work of the end-user. Furthermore, the fact that at the beginning, the analyst looks for functional requirements rather than non-functional ones, contributes to the lack of non-functional requirements in the use-cases.

**R2: The prototypes provide realism and allow the designer to evaluate the system's impact with the end-user. Within RUP, method prototypes are made by the developer based in the use-cases, since they provide the vision of the developers that is no always the vision and needs of end-users.**

Developers build prototypes on the base of use-cases, and use-cases are oriented to functional requirements. End-users approve prototypes because the functional requirements are satisfied; but they are not looking for the non-functional requirement at this stage. This relationship is therefore valid. Prototypes that show what a future system could look like seem not be very popular since these kinds of prototypes demand more time and effort. There is not much information at this time about prototypes within RUP literature. However, the respondents use power point as the most common tool to build prototypes. According the literature and the respondents, these prototypes are more oriented to show functionality because the non-functional requirements are only incorporated at the end of the development process. All respondents agree that prototypes help to find out how the user wants a system to work; to find the best way to surf and get information. The prototyping approach to system development helps to deal with the problem of understanding requirements. It is also a good tool to check that users' requirements really are being met by the design at different stages (Preece, 1994).

**R3: The end-user interaction involve end-user observing and evaluating under the whole system development. The end-user interaction can be limited because of time and cost especially in RUP that works with iterative process.**

This relationship was only partially validated because the literature and the respondents have taken a different approach to this question. The literature defends end-user participation under the whole development. *“The end-user is incorporated in the role of expert in work as a member more of the design team. Participatory design has three specific characteristics: Improve the work environment and task by the introduction of the design. This makes work oriented rather than system oriented. The second one is the collaboration: The user takes part in every stage of the project. Finally the iterative approach: the design is evaluated at each stage.”* (Dix et al 1997). Between the respondents there does not exist a consensus. The developer-respondents believe that the end-user already participates in the whole process because of the fact that a system-analyst identifies the actors and discusses their process with the end-user point-of-view. Moreover, the end-user has participation directly or indirectly in the approbation of transitions between phases of development. One developer respondent argues: *“we try to involve the end-user in the whole project. Of course, they are more involved at the beginning and the end. We try to do a test after the all iterations. They come to the test that we carry out and participate in the test”* (Appendix 2). The user-centered systems designer respondent points out that the end-user participation is their work relevant to the prototypes. There are different opinions about what the *“end-user participation in the whole process”* means. However all the respondents agree that RUP allows a closer participation of the end-user in the system development process.

**R4: “End-user has difficulties to explain their needs” this is frequent approach of expert design and developer. A closer observation from the designer to the end-user tasks could help to a better understanding of user-needs. RUP framework does no provide the tools for documenting a closer observation to the end-user and in this way caching the needs of the users.**

This relation is valid because the collection of requirements in RUP is performed by asking questions to the end-user. Answers are put in the use-cases; there is no room here for the observation of the end-user and consequently the user’s needs remain unknown. There is a general agreement among the respondents that, in this relationship, a closer observation of the end-user in their own work environment helps to clarify user duties and needs. This relationship is interesting because even though there is a consensus about the importance of the end-user in their work environment, only the usability-designer practices the observations in all the projects in which they participate. It is important to point out that usability-designers are not always present in a project. According to literature and one of the developer-respondents, usually the usability-designers are only called in towards the end of the development to take care of the view-design.

**R5: Developers are to busy learning and developing skills with emerging technologies, and techniques that are showing up everyday, that they don’t have much more left to take care of the end-user needs; Keeping the documentation in RUP demands a lot of extra work and time that could be used in the user-centered design.**

The careful management of documentation within RUP seems to be one of its additional disadvantages. At length of this study, it became apparent that user-centered systems require extra effort. Since keeping documentation in RUP demands extra effort too, these relationships seem valid. Moreover, the developer-respondents agree that developers have the tendency to be more technique oriented and they indicate that most of the people who work in IT are more interested in new techniques and programs codes than they are in usability. They agree too, in the fact that the usability issue needs to be considered in the development process. They all welcome the implementation of new role-player who takes care of usability aspects in the development process.

**R6: Rational unified process is oriented to software development it is not designed to cover the user needs.**

This final relationship too, is valid. Since all respondents and literature agree in this point; the Rational Unified Process is a software engineering process and it contributes to increase the project development. As in the literature review, the respondents praise the advantages of using RUP as development method. RUP offers many benefits as development method, but at the same time, RUP has a deficiency and offers poor support in the area of specification and non-functional requirements. Volvo, which has already implemented a usability discipline in RUP, explains its motive as being: *“the support for design and usability in standard RUP is weak, difficult to locate and, to some extent, misleading”* (Appendix 2).

## 9. Conclusion

*The handling of the research question and the result of the study is presented as well the problems connected with this study a reflection about future research topics*

### 9.1 Relationships

The purpose of this study was to ascertain end-user interactions with the Rational Unified Process in the system development process. End-user interaction is considered as an important factor in the creation of the user-centered design in a Rational Unified Process framework. Basing on a review of literature, several relationships were described between the end-users and the Rational Unified Process (RUP) in software system development in a real work environment. The relationships were tested by collecting empirical data from the individuals with experience in the research area. Perspectives on the relationship between the end-user and RUP as development method were offered from the developer and the usability designer points of view. The proposed relationships are:

R1: Interactive systems cannot be completely specified from the beginning of the lifecycle. Use-cases provide a concise method for modeling the functional requirements but not necessarily, they are good enough to catch the non-functional requirement. RUP is use-case based since it has a lack to handle the non-functional requirements.

R2: The prototypes provide realism and allow the designer to evaluate the system's impact with the end-user. Within RUP, method prototypes are made by the developer based in the use-cases, since they provide the vision of the developers that is not always the vision and needs of end-users.

R3: The end-user interaction involve end-user observing and evaluating under the whole system development. The end-user interaction can be limited because of time and cost especially in RUP that works with iterative process.

R4: "End-user has difficulties to explain their needs" this is frequent approach of expert design and developer. A closer observation from the designer to the end-user tasks could help to a better understanding of user-needs. RUP framework does not provide the tools for documenting a closer observation to the end-user and in this way caching the needs of the users.

R5: Developers are too busy learning and developing skills with emerging technologies, and techniques that are showing up everyday, that they don't have much more left to take care of the end-user needs; Keeping the documentation in RUP demands a lot of extra work and time that could be used in the user-centered design.

R6: Rational unified process is basically oriented to software development it is not designed to cover the user needs.

## **9.2 Purpose and research question**

What are the relationships between the Rational Unified Process and the end-user in the software system development process? Is the research question presented in this study Based on literature about the Usability design and The Rational Unified Process (RUP), some relationships were generated. Interview respondents later verified them, with varying results. All relationships, more or less, were considered as existing in the development process with the RUP framework. It is not the intention to claim that this study covers all possible relationships between the end-user and RUP. The participation of the end-user in the system development is an important factor in the project success. The diffuse understanding of the user participation showed in the relationships 1 and 3 (about the user-participation in the whole process) tells that there is still a long road to go through.

By analyzing the answers from the interviews, it is also clear that the different relationships have mutual relations, or overlap. For example, the utilization of the use-cases (R1) and the end-user observation in their own work environment (R3) both affect prototype elaboration (R2). Each individual relationship could be studied in a more detailed way. Nevertheless, this study focused to identify the relationships rather than the relationships between themselves.

The importance of end-user participation in the development process, in order to build systems that are more effective, has to be understood by the individual that orders the system and the other stakeholders. The user-centered design has to be a part of the requirements of the system. The literature contains many references about the lack of the interest of the developers in the user-centered issue. In defense of the developers, it should be stated that dealing with new techniques, complex systems and deathliness puts a high amount of pressure on developers; user-centered design demands a lot of extra effort from them. Moreover, they usually have a commitment with the individual that ordered the systems, a customer that they need to please. The end-users usually are different from the customer. Customer understood as stakeholder the one that actually orders the system.

## **9.3 Problems connected to the research**

The purpose of this study was to find out the interaction of the end-user with the Rational Unified Process in the system development process. In order to fulfill the objective for this study, companies were found that use RUP as system development method, and are concerned about the essential role of usability/user-centered design. Very few companies match this criterion since very few companies using RUP have the time or desire to consider the usability issue as a problem within their organizations. RUP seems to be a very popular method in the media, but the reality is hard to find companies that use RUP. The research area is interesting, but plagued with difficulties to collect both literature and empirical data that covers the research area in a credible way. The participation of a larger number of the participants would contribute to the reliability and credibility of this study.

## **9.4 Further Research**

As pointed out before, the importance of end-user participation in the development process, in order to build systems that are more effective, has to be understood by the individual that orders the system and the other stakeholders. The active participation of the end-users in the development process with RUP, and other development processes is a large field to study. The development of the processes that allow measurement of usability in a more accurate way can contribute in the development of user-centered systems. This research has fulfilled some of the relationships it set out to find, but it is certain there are further aspects to explore.



## 10. References:

Boerma, Peter (2005) *Introducing User-centered Design to an E-Government Software Development*. Bulletin  
<http://www.asis.org/Bulletin/Feb-05/boersma.html> (viewed 03/02/2008).

Boivie, Inger, (2005) *A fine balance: Addressing usability and user needs in the development of IT systems for the workplace*  
[http://www.nada.kth.se/~artman/Doktorand\\_texter/IngerB.pdf](http://www.nada.kth.se/~artman/Doktorand_texter/IngerB.pdf) (viewed 11/02/2008).

Brown, Barclay (2003), *Top five RUP implementation process killers*, Rational Software  
[http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/jun03/f\\_topfive\\_bb.pdf](http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/jun03/f_topfive_bb.pdf) (viewed 23/05/2008).

Bryman, A. (2002), *Social Research Methods*. Oxford, University Press.

Constantine, L & Lockwood, L (2001) *Structure and Style in Use Cases for User Interface design*, University of Technology, Sydney.

Creswell, John W. (1997), *Qualitative inquiry and research design*, Sage Publications.

Dix, A., Finlay, J., Abowd, G. & Beale, R, 1997, *Human-Computer interaction*, Pearson Education Limited

Gould, J. & Lewis C. (1985), *Designing for usability: Key principles and what designers think*, Communication of the ACM.

Gulliksen, Jan & Göransson, Bengt, (2002), *Användarcentrerad systemdesign*, Studentlitteratur AB.

Gulliksen, J., Göransson, B., Boivie I., Blomkvist, S., Persson, J. & Cajander Å (2003), *Key Principles for User-Centred System Design*, published in an international journal: Special Section "Designing IT for Healthy Work" in *Behavior & Information Technology*, Nov-Dec, Vol. 22 No. 6, Taylor & Francis.

Göransson, Bengt, (2004a), *User-centred Systems Design*, Dissertations from the Faculty of Science and Technology, Uppsala.

Göransson, Bengt, Lift Magnus, Gulliksen, Jan, (2003b), *Usability Design – Extending the Rational Unified Process with a new discipline*, Uppsala University  
<http://opus.uu.se/publication.xml?id=32982> (viewed 03/03/2007)

Göransson, B., Gulliksen, J., Boivie I., (2003c), *Usability Design – The Usability Design Process-Intergrating User-Centred System Design in the Software Development Process*, Software Process:Improvement and Practice (SPIP), vol.8, Wiley & Sons.

Jacobsson, I., Booch, G. & Rumbaugh, J. (1999), *The unified software development process*.

Kvale, Steinar (1996), *An introduction to qualitative research interviewing*

Löwgren, J., Solterman E. (2005), *Thoughtful interaction design*, The MIT Press, Massachusetts.

Marklund, S. *Arbetsliv och Hälsa*, (2000), Arbetslivsinstitutet, Solna, Sweden.

Miles, M. B, and Huberman, A. M. (1994): *Qualitative Data Analysis*, Sage Publications

Nielsen, Jakob (1993), *Usability Engineering*, AP Professional, Cambridge, MA., USA

Norman, D, & Draper, S (1986), *User Centered System Design: New perspectives of human-computer interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates, New Jersey.

Ottersten, I. & Berndtsson, J., (2002), *Användbarhet i praktiken – Praktiska handgrepp, grundbegrepp och tankemodeller*, Studentlitteratur.

Patton, Michael (1990), *Qualitative Evaluation and Research Methods*, Sage Publications

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

Rudisill, M., Lewis, C., Polson, P., & Mackay, T. (2000), *Human-compute and interface desing*.

Sarantakos, S (1993), *Social Research*, MACKMillan Education Australia Pty Ltd, South Melbourne.

Strand, Lotta, (2001), *UML & RUP – Att lyckas med OO-projek*, Docendo Sverige AB, Stockholm.

Strauss, A. & Corbin J. (1990), *Basics of Qualitative Research, Grounded Theory Procedures and techniques*, Sage Publications.

The Work Environment Act (2005)

[http://www.av.se/dokument/inenglish/books/AML\\_eng.pdf](http://www.av.se/dokument/inenglish/books/AML_eng.pdf) (viewed 16/05/2007)

Yin, Rober (2003), *Case study Research Design and methods (third edition)*, Sage publications,

<http://www.usabilitynet.org/tools/13407stds.htm> (viewed 26/11/2006)

# 11 Appendixes

## 11.1 Appendix 1: Interview guide

### Interview Questionnaire

Interview Questionnaire	
<b>Project:</b> The rational unified process (RUP) and its relationship with the end-user in the software system development.	
<b>Date:</b>	
<b>Company:</b>	
<b>Position of interviewee:</b>	
1.	Would you please tell me little about your work?
2.	How long have you work with The Rational Unified Process?
3.	Which roles do you play in the system development with RUP?
4.	What is the major contribution of RUP as development method?
5.	RUP is a use-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?
6.	Who is the responsible for finding out the requirements? There is a specific person for this role?
7.	Does the end-user participate in the building of use-cases?
8.	There is into the RUP the possibility of end-users observation the in their own work environments?
9.	Do you build prototypes for the end-user consideration before the start with the elaboration phase?
10.	Do you think the end-user is awarded of RUP being used as development method?

11. The participation of the end-users in the system development is mostly in the interception (requirement specification) and the transition (delivering and training) do you think that if RUP needs to be modified in that way it manages the requirements?
12. What do you think about creating a new discipline, which takes care of the user-centered issue?
13. How often do you need change the use-cases once the elaboration phase has begun?
14. How often do you need to adapt to an existent architecture?
15. Do the users ask for user-centred design?
16. Is the user-centred design part of the standard requirement even if the users don't ask for it?
17. Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have much more time left to take care of the end-user needs?

## 11.2 Appendix 2: Interview

### Interview Protocol

<b>Interview Protocol</b>
<p><b>Project:</b> The rational unified process (RUP) and its relationship with the end-user in the software system development.</p>
<p><b>Date:</b> 08/01/2007</p>
<p><b>Company:</b> Volvo information technology AB</p>
<p><b>Position of interviewee:</b> Project Manager</p>
<p><b>Short description of the subject interviewee experience with RUP</b></p>
<p>As Project Manager with over five years working with Rational Unified Process, she has worked in Volvo IT for two years, and before she worked in IKEA IT using RUP as development method. She has a wide experience with Rational Unified Process since she has used different version of RUP known as Rational IBM. She has occupied different roles into RUP in the system development process from system analyst to project manager.</p>
<p><b>What is the major contribution of RUP as development method?</b></p>
<p>I think that it is that you can manage changes in a better way. The major difference of RUP with respect to the older method is that the facilities that RUP offers for managing changes with RUP work strategically.</p> <p>You can take photograph of the user's needs have the possibility of take a test of what is going on in the project in a earlier phase and you can manager de end-user needs in a better way.</p> <p>I think RUP is one of the best available ways to work with managing change now. The way we use RUP, the end-user gets a lot of power over the end-result. I genuinely think that RUP is the best available method for developer and in the end; it is about making the application the user wants. I think we get closer with RUP than we got with other developer methods.</p> <p>The users get what they really require. I think with RUP we get closer to that, the one end-user really wants. The user is constantly giving us feedback.</p> <p>When an organization is so large it has to find some way to steer it in the right direction. A lot a work in the process project was going really bad, but with RUP we have the possibility of take a better control of the whole project and check that it is going in the right way.</p>
<p><b>RUP is a used-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?</b></p>
<p>Normally at least 20 %, I guess ..</p>
<p><b>Who is the responsible for finding out the requirements? There is a specific person</b></p>

<b>for this role?</b>
The system analyst is the person who catches the user's requirements I prefer the same system analyst works in the whole project even in the late phases when you get changes I prefer the same system analyst.
<b>Does the end-user participate in the building of uses-cases?</b>
Yes they do. We try to get as much participation from the customer in this work as possible
<b>There is into the RUP the possibility of end-users observation the in their own work environments?</b>
No, normally the customer does not get into the constructions phase, we build the first use-cases in our own environments
<b>Do you build prototypes for the end-user consideration before the start with the elaboration phase?</b>
Depending on how large the project is; we decide to make prototypes or just a power point presentation.
<b>Do you think the end-user is awarded of RUP being used as development method?</b>
We have actually trained them in development method in the big projects that I am managing. They have to be informed what this method means. I think that it is really important to explain to the end-user which role they play in this project because it is very different from the water-fall process. There, they don't need to know what RUP is in detail, but here I think that it is important to explain them the part they are playing in the project.
<b>The participation of the end-users in the system development is mostly in the interception (requirement specification) and the transition (delivering and training) do you think that if RUP needs to be modified in that way it manages the requirements?</b>
We use RUP a little differently because we try to involve the end-user in the whole project. Of course, as you said, they are more involved at the beginning and the end, but we try to do a test after the all iterations. I think that it is very important to get the user involved. They come to the tests that we carry out after the every iteration and participate in the test. We have accepting tests during the whole project. Every 4 or 5 weeks we do accepting tests where we have end-user really involved in the project
<b>What do you think about creating a new discipline, which takes care of the user-centered issue?</b>
We have that! We actually have had this discipline, starting from two years ago. I attended a conference dealing with this discipline. I think we actually have the possibility to get some support from the usability group coming down and discussing matters.

**The interviewee kindly completed this question with the following information:**

Some information about the Design & Usability RUP Discipline implemented at Volvo IT:

*Rational Unified Process, RUP, is the standard system development process used at Volvo IT. Unfortunately, the support for Design and Usability in standard RUP is weak, difficult to locate and, to some extent, misleading. The solution at Volvo IT is the Design & Usability discipline, based on a discipline developed by Guide Redina and Uppsala University, and customized for Volvo IT.*

**Purpose**

*The purpose of the new discipline is to integrate highest quality proven Design and Usability practices and research findings with standard RUP, in order to enhance the total quality of the systems delivered to our customers. Like all other RUP disciplines, Design & Usability must be adapted to each project where it is applied. A minimum requirement on each adaptation is that it covers user studies, user-interface prototyping and usability evaluation, in some form.*

**How often do you need change the user-cases once the elaboration phase has begun?**

That is a really tricky question because it depends on how detailed you have to be in the use-cases. Most recently we didn't have that level of detail in the use-cases. We handle the small changes like: what the screen should look like. We also handle some activities in the project out-side the use-cases. In other projects we have very detailed use-cases, including the screen-shop. Then you need to change the use-cases most often. Of course, you have to change the use-cases after the elaboration phase has begun. It is really hard get all requirements from the beginning. The big benefit of RUP is allowing managing the changes that are always present in all projects. Some times there are less or more important, but changes are needed in all projects.

**How often do you need to adapt to an existent architecture?**

Very large projects work actually with a quite new architecture and we don't have the limits of the architecture. It is hard to change the architecture. You have to adapt the requirements to the existent architecture and sometimes you have to convince the user that this requirement is not so important because you cannot afford to change the architecture in order to fulfill all the requirements.

**Do the users ask for user-centred design?**

That one is a bit of a tricky question because they want to be involved, but at the same time they don't to have responsibilities. Do you get me? If you ask them they will say "we want to be involved," but afterwards they don't take the full responsibilities that these kinds of projects involve.

**Is the user-centred design part of the standard requirement even if the users don't ask for it?**

In order to answer this question, I have to refer to the last project we did. This was a tricky one because the customer started the project with another supplier and they had concentrated a lot of screen design and only in designs question. I think we should focus on this kind of question later and start with the functionality because we are in system development and the usability requirement can be taken later.



**Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have much more time left to take care of the end-user needs?**

We have the tendency to be more technique-oriented. I mean, most of the people who work in IT are interested in new techniques, but I think we need to focus a little more on the end-user and Volvo IT has taken that into consideration. We already have a usability team that help us in the new projects to cover this gap between the development and usability design.

## 11.3 Appendix 3: Interview

### Interview Protocol

<b>Interview Protocol</b>	
<b>Project:</b> The rational unified process (RUP) and its relationship with the end-user in the software system development.	
<b>Date:</b> 08/01/2007	
<b>Company:</b> Tempogon AB	
<b>Position of interviewee:</b> Project Manager	
<b>Short description of the subject interviewee experience with RUP</b>	
As a Project Manager with over fifteen years of working with Rational Unified Process, he has known about RUP from the beginning and followed it's development at Rational Software and after with Objectory AB. He has worked with this method at Ericsson Telecom and now he works at Tempogon AB. He has occupied different roles into RUP in the system development process from system analyst to project manager.	
<b>What is the major contribution of RUP as development method?</b>	
Condensed experience. Unified process is not an invention for itself, I think the major contributions of RUP are that RUP collects experiences and puts them together in a box. Better said: in a book. RUP gathers and unifies terms and concepts, and lifts discussion to a higher level by using names, terms, concepts. RUP is a specific guideline. I think that at the beginning RUP was associated with failed projects, because RUP, by itself, doesn't guarantee success. You need experienced people to work with- and get good results with RUP. I see RUP as a framework with common model concepts that allow you to organize the project in a more efficient way. RUP offers a structure that divides a large project into several smaller and more controllable projects. The use-cases are really useful to find out requirements and they are a strong guideline when you need to make a decision.	
<b>RUP is a used-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?</b>	
You do not define use-cases at the beginning as in the waterfall method, but use-cases are built under the whole process; of course, at the inception phase, we work more, but it is difficult to say how much time. I care about the estimate because all the projects are different.	
<b>Who is the responsible for finding out the requirements? There is a specific person for this role?</b>	
The system analyst is the person who catches the user's requirements.	
<b>Does the end-user participate in the building of uses-cases?</b>	
Yes they do.	

<b>There is into the RUP the possibility of end-users observation the in their own work environments?</b>
Yes, we identify the actors and discuss their process from the end-user's point of view. A disadvantage with the use-cases is that they can be very abstract, some times, in seeing the whole process. The user focuses in his/her every day work and does not think about the functions that could be improved. Then the analyst needs to have a strong usability perspective to really find out the best way to improve the future system.
<b>Do you build prototypes for the end-user consideration before the start with the elaboration phase?</b>
Prototypes, power points. We need to find out how the user wants to work, to find the best way to surf and get the information that the user needs. We need to complete the use-cases with some user experience.
<b>Do you think the end-user is awarded of RUP being used as development method?</b>
They do not need to know what method is being used. Anyway, using use-cases is a tool to explain functions and roles and their relationships to the end-user. That is easy to understand for end-users.
<b>The participation of the end-users in the system development is mostly in the interception (requirement specification) and the transition (delivering and training) do you think that if RUP needs to be modified in that way it manages the requirements?</b>
In other methods we usually start with the data model and this is very difficult to understand to the end-user because files and data design are out of their way to think about their daily work. Using the use-case allows to the end-user see easily the functionality of the systems. They have participation directly or indirectly in the approbation of the the transition between one phase and the next.
<b>What do you think about creating a new discipline which takes care of the user-centered issue?</b>
Just I said before, I think that RUP has a deficiency and poor support in the specification of non-functional requirements.
<b>How often do you need change the user-cases once the elaboration phase has begun?</b>
Every day. Of course you have to change the use-cases after the elaboration phase has begun. It is really difficult to get all requirements from the beginning. The big benefit of RUP is that it allows managing the changes that are always present in all projects. Sometimes they are more or less important, but changes are needed in all projects.
<b>How often do you need to adapt to an existent architecture?</b>
Very large projects actually work with a quite new architecture and we do not have the limitations of the architecture. It hard to change the architecture. You have to adapt the

requirement to the existent architecture and sometimes you have to convince the user that this requirement is not so important because you cannot afford to change the architecture in order to fulfill all requirements.

I do not think that RUP is architecture centered. Of course, upon the existent architecture changes can be easier or more difficultly done. But this is not a problem in the method itself, but that the architecture is not flexible enough. Besides, architecture is always present in companies

**Do the users ask for user-centred design?**

Yes, absolutely

**Is the user-centred design part of the standard requirement even if the users don't ask for it?**

The use-cases are suitable to illustrate functionality, but you need prototypes and complete examples to show how the system will be and what vision the analyst has. Most of the users want to continue doing their work in the same way. I mean, they get used to work with the old system or how they work at the moment, so the analyst has to be good at showing the vision of the changes and how these changes could make the process easier.

**Do you think that developer are to much busy learning the new techniques and developing skills, with the new technologies that show up everyday that they don't have much more time left to take care of the end-user needs?**

The common developer is not especially interested in usability, but more interested in the C# code the technique details. Usability needs its own discipline into RUP, whether in the requirement, or in the design. Prototypes, I think, are a good way to incorporate usability in the design.

## 11.4 Appendix 4: Interview

### Interview Protocol

<b>Interview Protocol</b>
<b>Project:</b> The rational unified process (RUP) and its relationship with the end-user in the software system development.
<b>Date:</b> 11/01/2007
<b>Company:</b> Guide Redina AB
<b>Position of interviewee:</b> Usability designer
<b>Short description of the subject interviewee experience with RUP</b>
<p>I am a usability designer consultant; we are a system development company and work with different types of applications, web-sites and everything under the system development. We have focused on user-centered design and usability, as a usability designer, that means that I am responsible for usability related activities. I have worked with RUP for a long time; even as usability and user-centered researcher, first with Unified Modeling Language, and after with RUP.</p> <p>Working with the Rational Unified Process, we have developed a new usability design discipline, which aims to complement RUP with a more user-centered focus.</p>
<b>What is the major contribution of RUP as development method?</b>
<p>I think that the main contribution of RUP is that it allows for communication among project members. RUP works very well as a communication language. Documentation is also a good contribution; we have a very good vision of what is going on. On other hand, sometimes it can be a disadvantage because RUP can produce too much documentation and we expend a lot of time maintaining the documentation instead of working in the system's functionality. However, the end system you get is a very well-documented system.</p> <p>RUP is a very popular method and it is important to add to it a user-centered focus.</p>
<b>RUP is a used-case based method, how much time do you take for building the use-cases? Which percent represent that of the whole development?</b>
<p>I am not usually responsible for building the use-cases. I work together with the systems analyst for use-cases. We work parallel with the designer and prototypes. This task takes 20 or 25 percent of the whole project the time; it can vary depending of the project.</p>
<b>Who is the responsible for finding out the requirements? There is a specific person for this role?</b>
<p>In normal cases when we find out the requirements, the requirement analyst and I, as usability designer, work in parallel together. The requirement analyst works with the documentation and I work with the prototypes. In my role as usability designer, I meet with the end-user and find out the end-user's needs. I meet the end-users, make some interviews, and workshops in order to collect the information I need to build the</p>

prototypes.
<b>Does the end-user participate in the building of uses-cases?</b>
Yes, the importance is to catch the requirements but I think that it is relevant that they work with the prototypes. In this way, it is easier to document what they have for requirements
<b>There is into the RUP the possibility of end-users observation the in their own work environments?</b>
In almost all projects that I work in, I need to go to user's workplace and see how they work in their ordinary environments. It is a part of my work. The observation can be done in different ways. The usability designer is next to the end-user and observes the end-user using their system; how they interact with the systems. If you do a good and careful observation, you can answer the question of whether the system has a usability design.
<b>Do you build prototypes for the end-user consideration before the start with the elaboration phase?</b>
This question is difficult to answer. The prototypes vary. They can be power point or rough draft on paper or some other tool. If we talk about prototypes on paper then they could be hundreds for one project. If we talk about developed, running prototypes then there could be two.
<b>Do you think the end-user is awarded of RUP being used as development method?</b>
I would say that in many projects they don't need to know. If they are involved, for example, in use-case formulation then they are awarded, but sometimes when we are working with the first study and we are just observing them to understand their job, it does not matter which methods we use, they don't need to know.
<b>What is the main contribution of the new usability discipline to RUP?</b>
Since the goal of this discipline is to develop user-centered systems with the implementation of the usability role, we expect to get a strong focus on users and usability. That is the advantage of using this discipline. Another one is that you work in a more simple way. I mean, you work with usability easier if you have a well-defined discipline within RUP. Your work with usability within RUP's terms.
<b>How often do you need change the user-cases once the elaboration phase has begun?</b>
Not applicable to this interviewee
<b>How often do you need to adapt to an existent architecture?</b>
It is difficult to say because there are many factors involved and they can vary. They could be technique, time, and whatever limitations. Then the changes are a product of the compromise between the stakeholders.
<b>Do the users ask for user-centred design?</b>
Yes, in certain cases when they are discontent and they really want to get a change.
<b>Is the user-centre design part of the standard requirement even if the users don't ask for it?</b>
The usability degree varies. The usability issue is important to us as usability designers, but it can have different levels.

**Do you think that developers are too busy learning the new techniques and developing skills, with the new technologies that show up everyday So that they don't have much more time left to take care of the end-user needs?**

I am not the best one to assess that because I am not a developer, but I think usability must be in the development process, because it does not matter how high technology we use in the development if the person who works everyday with the system is not happy with it. People sometimes don't realize how annoying it can be to work, or be obligated to work with a system that really does not help you. The work is more difficult. I have seen employees that just stop using the system. So ask them if the technique is important.