APPLYING SYSTEM DEVELOPMENT METHODS IN PRACTICE - The RUP example

Sabine Madsen and Karlheinz Kautz

1. INTRODUCTION

System development methods have already long been controversially discussed, but there is still a lack of knowledge and understanding based on empirical studies about how systems development is actually conducted in practice, how system development methodologies and methods are used and to what degree they are used as proposed in the literature (Floyd, 1986; Nandhakumar & Avison, 1999). The purpose of this paper is to contribute to this understanding. It reports how and to what degree Rational's Unified Process (RUP) was used in two commercial development projects.

RUP is considered a state-of-the-art, object-oriented methodology with a focus on iterative and incremental development features and has been promoted as a solution to problematic issues in systems development such as unfinished projects, budget and time overruns, erroneous systems and systems with lacking functionality (Boehm, 1988; Jacobsen et al., 1999).

This paper presents an empirical case study in a consultancy firm. The case study is based on interviews with experienced project managers and systems developers, who participated in the two projects. The paper is structured as follows: Section 2 introduces the background and related work of the study. Section 3 introduces the conceptual framework, which is used to analyze the empirical findings from the case study. In section 4 RUP is explained and section 5 describes the research approach, which has been used for data collection.

1 Sabine Madsen and Karlheinz Kautz, Copenhagen Business School, Department of Informatics, Høvitzvej 60, DK-2000 Frederiksberg, Denmark.
collection and analysis. Section 6 presents the case study and section 7 contains a discussion of the empirical findings. The last section summarizes the main conclusions from our investigation.

2. BACKGROUND

The major part of systems development literature concerns methodological development. Numerous books provide a vast number of different methods specifying prescriptive guidelines on how to develop systems. These methods are by and large based on the assumption that system development is a rational, goal-driven and managed process and it is taken for granted that there is a need for a method to facilitate this process (Truex et al., 2000). The theoretical proposition is that the use of methods reduces production time and complexity and improves the development process as well as the quality of the final system.

Another stream of literature concerns amethodical systems development. The term amethodical as coined by Truex et al. (2000) does not entail an anti-methodological or methodless approach to systems development. It rather presents a critique of the methodological literature. As the authors put it, it implies management and orchestration of systems development without strictly predefined structure, sequence control, rationality or claims for universality, but it does not mean chaos or anarchy. The main argument is that the assumptions behind the prescriptive methods do not resemble how systems development is conducted in practice. According to the amethodical view systems development is a unique, negotiated and opportunistic process driven by accident (Truex et al., 2000). Such a position has earlier been formulated by Floyd et al. (1989) and Kautz (1993) using the concept of evolutionary systems development and have, in addition, recently been reframed as adaptive (Highsmith III, 2000) or agile development (Cockburn, 2002). This point of view is also supported by empirical studies, which suggest that systems development can be characterized as a somewhat amethodical activity, where formalized methods are not used at all or where only certain tools or techniques from a particular method are used (Bansler & Bødker, 1993; Fitzgerald, 1997, 1998).

Based on their study Bansler & Bødker (1993) conclude that there is a wide gap between how the Structured Analysis Method is proposed in the literature and how it is used in practice. Stolterman (1994) and Fitzgerald (1997) further report that experienced developers adapt and apply methods in a pragmatic way. Systems developers tailor the method to the project at hand by omitting method aspects, which are too time-consuming, cumbersome or irrelevant for the particular situation. Also indicating a problematic relationship between prescriptive methods and practice, Wastell (1996) reports from a case study in which the Structured Systems Analysis and Design Method (SSADM) was used rigorously. However, instead of improving the development process, the method inhibited creative thinking and caused the developers to focus on details instead of on the overall aim of the project.

Recently the discussion regarding system development methods has gained renewed interest in the context of web development. One stream of literature argues that traditional development methods are applicable for web development (Chen et al., 1999; Murugesan & Deshpande, 2001), while another stream argues that development of web-based systems is
fundamentally different and therefore entirely new methods and approaches are required (Braa et al., 2000; Greenbaum & Stuedahl, 2000; Baskerville & Pries-Heje, 2001; Carstensen & Vogelsang, 2001). Between these extremes it has been suggested that front-end oriented web development (of the user interface) requires new methods and approaches, but back-end oriented and technically complex web development (of the functionality) requires traditional methods (Pressman, 1998). This is supported by Eriksen (2000). Based on an empirical study he concludes that traditional development methods are useful for development of back-end functionality, but provide little guidance with regard to the web-based front-end.

The two projects in this case study were large-scale back-end oriented web projects. However, due to their scale and complexity we do not perceive them to be any different from traditional systems development projects and below we will refer to them as such.

3. **RESEARCH FRAMEWORK**

Numerous definitions of the concepts methodology and method exist (Avison & Fitzgerald, 1995). To analyze the findings from our case study we draw upon Mathiassen et al.’s (1990) definition of a method. They define a method as a disciplined, structured approach to solve a problem, which is characterized by (1) its area of application, (2) the underlying perspective and (3) guidelines for performing the process with the help of a) techniques, b) tools and c) principles of organization.

A method has an area of application for which it is suitable depending on type of information system, on project size, on team size etc. Furthermore, a method is based on an underlying perspective, i.e. on a set of assumptions. These assumptions determine the type of questions, which are asked to analyze the problem at hand, the type of solutions, which are proposed etc. At the more practical level a method consists of a number of guidelines regarding techniques, tools and principles of organization. A technique indicates how an activity should be undertaken; a tool is linked to a technique and is used to ensure that an activity is undertaken in the most effective way. Principles of organization indicate how people and groups should work together and how limited resources should be allocated. Examples of principles of organization are: division of the project into phases and guidelines about user involvement.

4. **RATIONAL’S UNIFIED PROCESS**

---

2 The terms methodology and methods are much debated in the literature, but a discussion of their distinguishing characteristics is not part of this paper. For the purpose of this paper we will use the terms interchangeably.
In the early 1990’s a large number of different object-oriented methods had emerged. Jacobsen, Booch and Rumbaugh had each authored their own method, but in the mid 1990’s they joined forces to unify their different notations into one consistent modeling language, now known as the Unified Modeling Language (UML). They continued collaborating and further unified the prevailing ideas about systems development into Rational's Unified Process, a full-fledged process model that claims to support the entire systems development life cycle (Jacobsen et al., 1999).

RUP was originally developed for traditional and large-scale systems development, but it is not only a single process. RUP provides a generic process framework, which can be customized to fit many different projects, different types of organizations, different levels of competence and different project sizes (Jacobsen et al., 1999). Thus, RUP's application area is claimed to be very broad.

RUP is characterized as a use case driven, architecture centered, iterative and incremental process model. Use cases define the functionality of the system and each use case describes the step by step actions, which the system performs to provide the user with a result. Use cases are used for requirement specification and for splitting a project into suitable and manageable increments. For each increment one of the most important activities is to find, test and evaluate the architecture, i.e. the technical systems design consisting of the infrastructure, components and interfaces, which make up the system.

The RUP terminology contains 4 phases, called the inception, elaboration, construction, and transition phase. The inception and elaboration phases are also labeled the engineering stage and during these phases the analysis, design and planning activities are undertaken. During the construction and transition phases, also called the production stage, the coding, testing and deployment activities are performed. A project is divided into a number of iterations and for each iteration the project goes through all four phases. Furthermore, RUP is based on incremental coding, which means that for each iteration an increment, i.e. a part, of the overall system has to go through all four phases. This allows the project team to incorporate the lessons learnt, when the next iteration is initiated and the idea is to help the project team discover major obstacles in time. Thus, RUP provides a framework for tailoring an iterative and incremental process and for selecting tools and techniques to fit a given project.

RUP has a strong focus on documents and the activities in the inception and elaboration phases mainly concern the creation of diagrams and writing of textual descriptions. The UML notation and the software program, Rational Rose, support this work.

5. RESEARCH METHOD

The research presented in this paper is based on empirical data from a case study in a consultancy firm in Norway, which used RUP in two projects. The case study consisted of seven interviews. Each interview lasted 45-90 minutes and the participants were the Director of Process and Technology (responsible for system development methods at the company), a Method Consultant, two Project Managers, one Chief Programmer and two Systems
Developers. The participants covered a wide range of roles and activities in systems development and had between 4-10 years of experience with systems development projects. Below we refer to the participants with the more general term of consultants.

Data collection was carried out using semi-structured interviews and each interview was taped. The interviews were structured around an interview guide, which focused on their experiences with RUP as a development perspective as well as its techniques, tools and principles of organization.

After the interviews had been conducted, the main topics were identified and a detailed, descriptive account of each interview was written up. Each participant received a copy of this account for correction and approval. Furthermore, a management summary outlining the main topics and conclusions from the case study was written. This, too, was sent to the participants for correction and approval.

6. CASE STUDY

The case study was performed in a large consultancy firm with more than 500 employees and considerable experience with systems development. RUP was ‘officially’ introduced in the company in January 2001. Top management had decided that the initiation of RUP should take place via a number of pilot projects and at the company level RUP had to be adapted to the company's approach to system development. Thus, a number of guidelines on when and how to use RUP had to be developed. However, when this study was conducted in October 2001 the management decisions regarding the introduction of RUP had not been clearly communicated to the organization and the development of guidelines and a formal adaptation of RUP had not yet taken place.

The case study concerns the experiences, which the interviewed consultants had gained while using RUP on two large-scale projects, project A and B. Project A lasted 12 months and involved 18 people. At the time of the interviews the project was ready for the final delivery to the customer. Nine of the team members were from the consultancy firm, including the project manager. The other nine team members were from a supplier of a large ERP system, which constituted the basis software for the project. The consultancy firm had the overall responsibility for the project. Furthermore, they had the responsibility for the presentation layer, i.e. the design and coding of the front-end, and the integration of the presentation layer and the data from the back-end. The supplier had the responsibility for the back-end and for supplying data to the presentation layer.

Project B was initiated in May 2001 and the first phase of the project had just finished, when the interviews for this case study were conducted in October.12 people had been involved in the project, 6 of these working full time. The second phase of project B is expected to last another 15 months and will involve around 10 people full time.

With regard to project A RUP was chosen as the development methodology due to an internal wish to try this method, where as for project B RUP was chosen due to a request from the customer. Below the experiences, which the two project teams have had with RUP, will be
presented. The description will be structured according to the following RUP features: the
development case document, iterations, use cases, architecture, documents and as a
connecting link RUP’s relation to formal development contracts.

6.1. Development Case

The purpose of the development case document is to help the project team tailor the
process to fit the project. The development case document describes the kind and number of
tools, techniques and documents to be used in a particular project. One consultant stated that
RUP recommends that the team dedicates time - RUP proposes 2 weeks - in the beginning of
the project to perform this activity and that the development case document is updated
throughout the entire project. Thus, the methodology itself indicates that it takes a lot of time
to plan for using RUP. The team has to plan how many iterations they need, which documents
they need and later in the process they have to plan for incremental coding.

In both projects a development case document was developed and at the beginning of
each phase an attempt was made to use it, but in both projects it resulted in too many
documents. In project A some of this documentation was actually not used during coding and
implementation and one consultant expressed that it takes a lot of skill and experience with
RUP to select the right documents and to find the right level of detail. The B team did not find
the development case template particularly helpful for the purpose of tailoring an iterative
process to fit the project. They felt that the focus was far too much on documents and less on
iterations.

6.2. Iterations

The purpose of iterative development is to ensure a learning process, where experiences
from one iteration can be incorporated in the next. The aim is to learn about project risks,
changing and emerging requirements as well as technical obstacles as early in the process as
possible.

In both projects the development process was divided into two main phases. Using the
company's own rhetoric these two phases were a specification phase covering the inception
and elaboration phase according to RUP terminology, i.e. the engineering stage and an
implementation phase covering the construction and transition phase, i.e. the production
stage. In both projects the contract was re-negotiated between the two phases.

In project A RUP was used on request of the project team, but when the contract was re-
negotiated after the specification phase, the customer was not interested in iterative and
incremental development. Instead the customer wanted a traditional development process with
a design phase, an implementation phase, test etc. and this development process had to be a
part of the contract. Even though the contract was based on a traditional waterfall model
during the second phase of the project, the templates from RUP were still used for design.
However, the team found it difficult to plan for and use iterative and incremental coding. They
were running on a very tight schedule and did not have any experience with RUP from
previous projects. They therefore ended up following the process, which they were familiar
with. So due to the customer and the contractual circumstances as well as lack of experience with RUP, iterations were not used during the last part of project A.

In project B there were three iterations during the specification phase, where the customer received use cases and other documents for approval after each iteration, but after the specification phase the project team experienced difficulties in getting the customer to accept the final set of deliverables. The project team realized that they had not paid sufficiently attention to getting the customers' full accept after each iteration. Furthermore, they became aware that the customer was expecting the deliverables to be very detailed, while the B project team had tried to work at a broader level as recommended by RUP. The customer expected the outcome of the specification phase to be a detailed documentation of what was to be the final system, as if the system was developed according to the traditional waterfall model. In contrast, the B project team expected the outcome of the specification phase to be use cases and architecture documents, which had to be further refined during the following phases. In other words, there was a mismatch between expectations due to different development perspectives and even though the customer had requested that the system should be developed according to RUP, the interviewed B team members felt that the customer had not really understood the RUP process. The customer lacked an understanding of the process as a learning process, where decisions are very abstract and broad in the beginning, but get continuously more and more detailed as the project team and the customer get a better understanding of the system.

6.3. The Contract

The purpose of the contract is to formally establish the economic and legal context in which development of a given project can take place. However, both teams experienced a mismatch between the assumptions, which are underpinning the legal contracts and traditional ways of doing business and the assumptions, which RUP is based on. The consultancy firm has normally followed the traditional business rules for commercial systems development, where the contract and the payment is tied together with and based on a requirements specification. According to these business rules it is assumed that the outcome of the first phase is a complete and final description of the system, which will be the basis for the rest of the project and a fixed price contract. RUP, however, assumes that the project team does not really know what they are going to develop before much later in the process. Therefore, they have to start at an abstract, general level and work their way to a more and more detailed understanding of the system via iterations, a focus on architecture and early coding on core parts of the system. Thus, when developing according to RUP, the project team learns about the project throughout the process and therefore it is not possible to have a complete and final requirement specification early on in the project.

The interviewed consultants experienced this as a huge challenge with regard to the customer. The customers in project A and B wanted a fixed price contract. They wanted to be sure that they would get, what they were paying for. The consultants explained that the problem with a fixed price contract is that the price is estimated based on a number of
assumptions about project scope, scale and functionality, but with RUP’s iterative development perspective it is explicitly recognized that these assumptions will change during the process. However, when the project is based on a fixed price contract, the project team has to be critical towards changing requirements, which will increase the project costs, and consultants from both teams stated that a fixed price contract inhibits a true learning process.

RUP’s focus on systems development as a learning process was not only a challenge with regard to the customer. It was also a challenge for the systems developers, because they too felt uneasy and out of control when working at the more abstract levels. They were used to working according to the traditional waterfall model, so they also assumed that they should have a complete description and understanding of the final outcome early on in the project.

6.4. Use Cases

The purpose of use cases is to help the project team with requirement specification and the division of the project into suitable increments. Starting by identifying the different types of users in order to develop an apprehension of the system’s purpose, use cases are refined to describe the system’s functionality at a more detailed level, understandable and appropriate for both users and developers.

However, with regard to requirement specification it was a challenge for both teams to identify and describe the use cases. One consultant stated that the examples in the books from Rational are very easy and obvious, but in practice most requirements do not come as neat and simple use cases. Instead the requirements had to be split into several use cases and for both teams it was a time-consuming and challenging task to identify the relevant ones. Furthermore, one of the interviewed B team members stressed that it is important to determine the purpose of the use cases. The experience from project B – although intended differently by the methodology - was that the use cases were too focused on functionality and less on who the user is and what the user’s goal is. The consultants themselves thought that the use cases in the beginning should have focused on the user in order to help the developers understand who and what the system was projected for. This would have been a better starting point for a subsequent refinement with a focus on functionality.

In both projects use cases were used for requirements specification, but in project A they were not used as a tool for planning an incremental development process. With regard to project B it is the intention to plan for and use iterative and incremental development based on use cases, when the project enters the implementation phase.

6.5. Architecture

The purpose of RUP’s strong emphasis on architecture is to guide the project team in establishing a stable architecture, which the project can be based upon and to ensure that the team discovers technical difficulties as early as possible. Therefore, RUP prescribes that the architecture is tested and evaluated continuously throughout the development process and both teams dedicated time to perform this activity.
However, in project A the customer abandoned the architecture, which was recommended by the project team. One of the main components in the recommended architecture was an ERP system from a particular company, but the customer had already established a business relationship and a preference for another supplier. The chosen architecture was not tested and evaluated anew, and the project team did, therefore, not discover the major technical obstacles before late in the implementation phase. Furthermore, the recommended architecture was based on an object-oriented technology, but the chosen ERP system was not object-based. This meant that the design documents, which had been created in the specification phase, were useless for actual coding and implementation.

In project B the architecture was also tested and evaluated in the specification phase, but the project had not entered the implementation phase, when this study was conducted, and therefore we are unable to report further from their experiences.

6.6. Documents

The purpose of RUP’s analysis and design documents is to create a documentation set, which is useful and valuable throughout the entire development process.

Both teams used Rational Rose for drawing use case diagrams and in both projects it was not difficult for the customer to understand the use cases. But the interviewed consultants experienced that due to the amount of pages with use cases it was difficult for the customer to evaluate them all and to give useful feedback to the project teams.

Furthermore, both teams used architecture documents, including class diagrams and domain models, as RUP suggests, and they did not experience difficulties in the outset when using the templates and drawing diagrams in Rational Rose. But the architecture documents did pose a challenge for both teams later. The A team experienced difficulties with the usage and maintenance of the class diagrams, because the chosen architecture was not object-based and in the end the class diagrams were discarded altogether. For the B team the architecture documents turned out to be a challenge for the customer. The team had made a number of architecture documents from different perspectives, as recommended by RUP. However, it was difficult for the customer to understand these different perspectives, and especially to see them as part of one coherent description, and therefore they were reluctant to accept them.

7. DISCUSSION

Even though the interviewed consultants experienced difficulties with the usage of RUP, they all stated that they had a positive impression of the method and would like to use it again. However, when comparing their experiences with Mathiassen et al.’s (1990) definition of what characterizes a method, the project teams only used 2 out of 5 characteristics. In both project A and B the techniques and templates from RUP had been used, but the project teams – although they had planned and attempted to - did not succeed in using RUP for structuring the development process and they did not adopt the underlying development perspective.
RUP was not used for tailoring and managing an iterative and incremental process. Instead the two projects under investigation ended up following a traditional development process, i.e. a waterfall model, supplemented with tools and techniques from RUP. RUP was primarily used as toolbox, from which tools and techniques were selected and applied in a pragmatic way. This is in keeping with Fitzgerald (1998), who concludes that in practice the most evident contribution of a system development method is as a toolbox, and not as a process framework. It also supports Truex et al. (2000), who argue that systems development is an opportunistic, negotiated and compromised activity, which does not follow the rationalistic ideal of a predefined process. In our study opportunism, compromise and negotiations became obvious and visible through the developers’ and customers’ lack of experience with the methodology, the contractual circumstances and the general behavior of the customers.

Whether a project team with more experience with RUP could have avoided the described situation, has however to be questioned. The project members were after all very experienced IT professionals. Thus, the role of the development contracts and that of the customers has to be revisited. Our case study indicates that iterative development and the explicit focus on systems development as a learning process cause difficulties, when systems development is performed according to a fixed price contract. This type of contract necessitates strict cost control, and thereby inhibits a true learning and development process based on intermediate, not fully documented specifications. Mathiassen & Bjerkes (2000) present a similar argument and discuss the balance between trust and control with regard to contracts and client-contractor relationships. While trust promotes creativity and mutual learning, a contract promotes decisions and monitoring of progress according to the agreement. Mathiassen & Bjerkes (2000) therefore suggest that there is a need for a well-adjusted relation between trust and control in order to create an environment for learning. They conclude that it is impossible to improve systems development practices without changing the current form of contracts. This case study supports that argument.

Mathiassen & Bjerkes (2000) reason more about the customer-supplier liaison. In our case the customers had a tremendous influence on the course of the projects and the utilization of the methodology. They had a different understanding of the methodology; they were only partly interested in incremental development and more in favor of detailed written specifications as results of distinct phases. In one project they even made a technical, architectural decision, which was in conflict with the methodology’s development approach. In such an environment the development organization and the developers could not apply the methodology as described in the method guidelines and as intended by themselves. They had to make concessions, go through negotiations and find a pragmatic and practical way to deliver the demanded product. As our case demonstrates, systems development has to reconsider the customer-supplier relationship to enhance practice.

As an object-oriented and iterative method RUP has been marketed as a solution of the problems in systems development. But it appears that even though RUP is claimed to be a modern method with a seemingly broad area of application, it pays little attention to the context in which commercial systems development takes place. However, whether it is feasible
to incorporate all activities performed in systems development in one methodology has to be doubted. In line with Cockburn (2002) our study shows that systems development, as actually performed, is such a complex process that it cannot be accurately described. And if it could, no one would be able to read such complicated description and learn from it, how to perform systems development. The challenge is to find an equilibrium between the methodical and the amethodical elements of systems development.

8. CONCLUSION

This case study has described how and to what degree RUP was used in two large-scale development projects in a consultancy firm. In summary we put forward three main conclusions.

In both projects tools and techniques from RUP were used, but the project teams did not succeed in using RUP as a framework for structuring and managing the process and they did not adopt the underlying development perspective. Thus, RUP was used as a toolbox, not as a process framework. These findings are in accordance with other empirical case studies and lend support to the proposition that in practice system development is a somewhat amethodical activity, where methods are used for selecting and applying tools and techniques in a pragmatic way. In the literature method is primarily related to the concept of process (Truex et al., 2000), but in these two projects the prescriptive process seemed to play a secondary, if not insignificant, role.

Furthermore, we have reported that the project teams experienced difficulties with RUP's iterative development features when developing according to a fixed price contract, because a fixed price contract requires strict cost control, thereby inhibiting the learning process, which is the aim of iterative development. RUP is claimed to have a broad area of application, but this case study indicates that it pays little attention to the contractual and economic issues of system development.

Finally, we discussed the customer-supplier relationship in systems development. RUP promotes the active involvement of clients and future users. This had severe consequences for the cooperation of the different stakeholder groups and the application of a methodology like RUP. To enhance systems development there is clearly a need to rethink the customer-supplier relationship, both in the context of methodologies and beyond, and future research is needed to further explore the advantages and disadvantages of iterative development in a commercial setting.
LITERATURE


Chen L., Sherrell L.B. & Hsu C., 1999, "A Development Methodology for Corporate Web Sites", First ICSE Workshop on Web Engineering (WebE-99), Los Angeles, USA.


