Seminar in Software Engineering
Specification Methods

Report on Practical Projects

Institut für Informatik
Universität Zürich

Prof. Dr. Martin Glinz
Nancy Schett

WS 2000/2001
January 23, 2001

Author :

Laurent Bagnoud
Bülachstr. 7g
8057 Zürich

laurent.bagnoud@switzerland.org

Table of Contents

1 Introduction

The purpose of this paper is to expose how the specification of requirements for a project is managed and accomplished using the Rational Unified Process (RUP) of the firm Rational Software Corporation. We will first recall the basic theoretical concepts of RUP. The different activities and artifacts of the requirement core workflow which guide the analyst throughout the specification of the requirements will be then dissected and explained. The practical use of RUP for the specification of requirements for a project will be explored with two real projects : the Java-GUI for Profile Generator AMAT project by Zühlke Engineering in Schlieren and the Marconi project by ISNet in Sierre. For a practical understanding of these projects the author of this paper met the project directors and asked them to describe the objective aimed by the project and its environment. The way they made use of RUP for the management and the specification of the requirements and a brief valuation of RUP will also be given by the project directors of the two projects. In the last section of this paper, the author will give his personal appreciation of RUP in general and a valuation of the way Zühlke and ISNet use it for the specification of requirements.

2 The Rational Unified Process

2.1 Introduction to the Rational Unified Process

It is not the purpose of this paper to fully describe the entire Rational Unified Process (RUP). We will only recall some basic ideas of it and dress a rough portrait of its meaning in the management of a project. Note that an evaluation version of the online documentation can be found on the web site of the Rational Software Corporation (www.rational.com/rup).

RUP is a Software Engineering Process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget.

RUP provides every team member with easy access to a knowledge base with guidelines, templates and tool mentors for all critical development activities. By having all team members accessing the same knowledge base, no matter if they work with requirements, design, test, project management, or configuration management, RUP ensures that all team members share a common language, process and view of how to develop software.

RUP is supported by tools which automate large parts of the process. They are used to create and maintain the various elements of the software engineering process: visual modeling, programming, testing, etc. They are invaluable in supporting all the bookkeeping associated with the change management as well as the configuration management that accompanies each iteration.

RUP is a configurable process. It fits small development teams as well as large development organizations. It is founded on a simple and clear process architecture that provides commonality across a family of processes.
The Rational Unified Process is an online mentor that makes process practical by providing extensive guidelines, templates, and examples for all critical e-development activities.

2.2 Process Overview

RUP is based on six best practices that are observed to be commonly used in industry by successful organizations. These 6 best practices are :

1. Develop software iteratively
2. Manage requirements
3. Use component-based architectures
4. Visually model software
5. Verify software quality
6. Control changes to software

Each team member receives guidelines, templates and tool mentors to take advantage of these 6 best practices. We won't detail these 6 points in this paper. For a in-depth explanation we recommand you read the whitepapers of the Rational Software Corporation [1].

The Rational Unified Process can be described in two dimensions : the horizontal and the vertical axis. The horizontal axis represents time while the vertical axis represents how the process is described in terms of activities, artifacts, workers and workflows. Figure 1 depicts how the process is structured along the two axis.

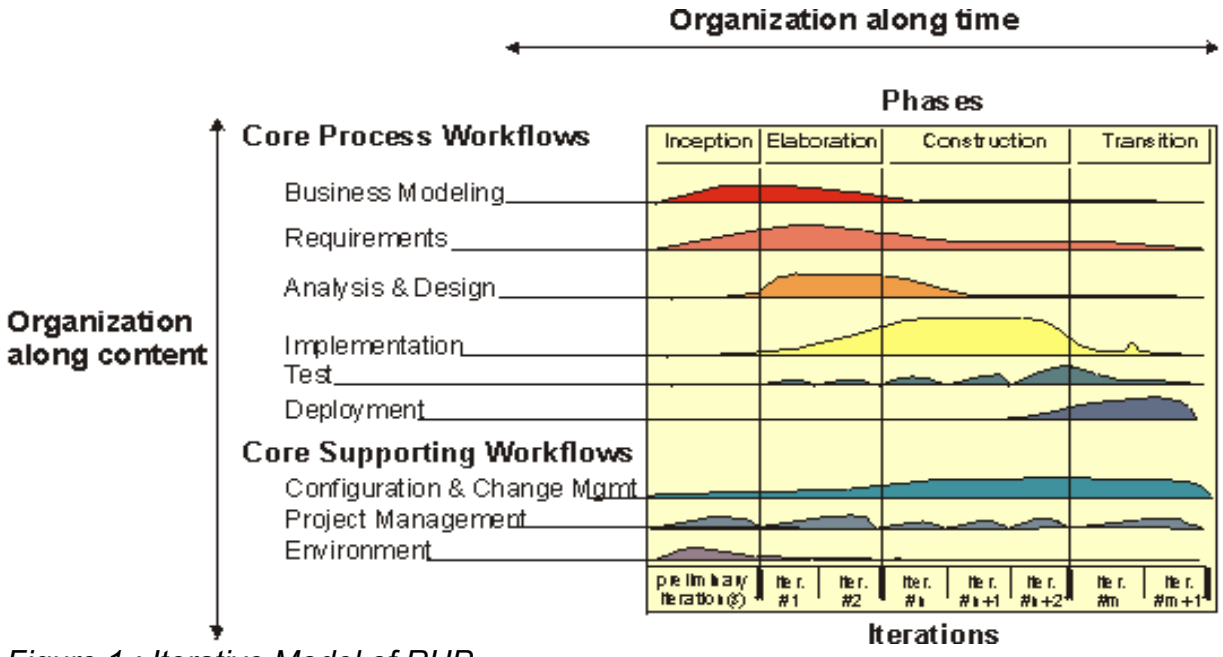

Figure 1 : Iterative Model of RUP

3 Specification of Requirements with RUP

Within the RUP terminology, we define a requirement as a condition or capability to which the system must conform. There are many different kinds of requirements. One way of categorizing them is described as the FURPS+ model, using the acronym FURPS to describe the major categories of requirements with subcategories as shown below.

- Functionality,
- Usability,
- Reliability,
- Performance and
- Supportability.

The "+" in FURPS+ reminds you to include such requirements as design constraints, implementation requirements, interface requirements and physical requirements.

During the Requirements process, we will have to manage the requirements. The RUP definition of requirements management is that it is a systematic approach to

- eliciting, organizing, and documenting the requirements of the system, and
- establishing and maintaining agreement between the customer and the project team on the changing requirements of the system.

Keys to effective requirements management include maintaining a clear statement of the requirements, along with applicable attributes for each requirement type and traceability to other requirements and other project artifacts.

We will now have a closer look at the different steps in the requirements core workflow of the RUP. When we work with the Rational Unified Process, we can adopt the online documentation provided by the Rational Software Corporation. It gives us a step-by-step process to reach the aimed goal of the specification of the requirements. The workflow details are very useful and describe the essential for each team member to work efficiently. Each element of these workflows can be clicked on. A window with the definition of the element, explanations, check-lists, templates or useful links to other topics will then appear. The requirements core workflow can be represented by the following graph.

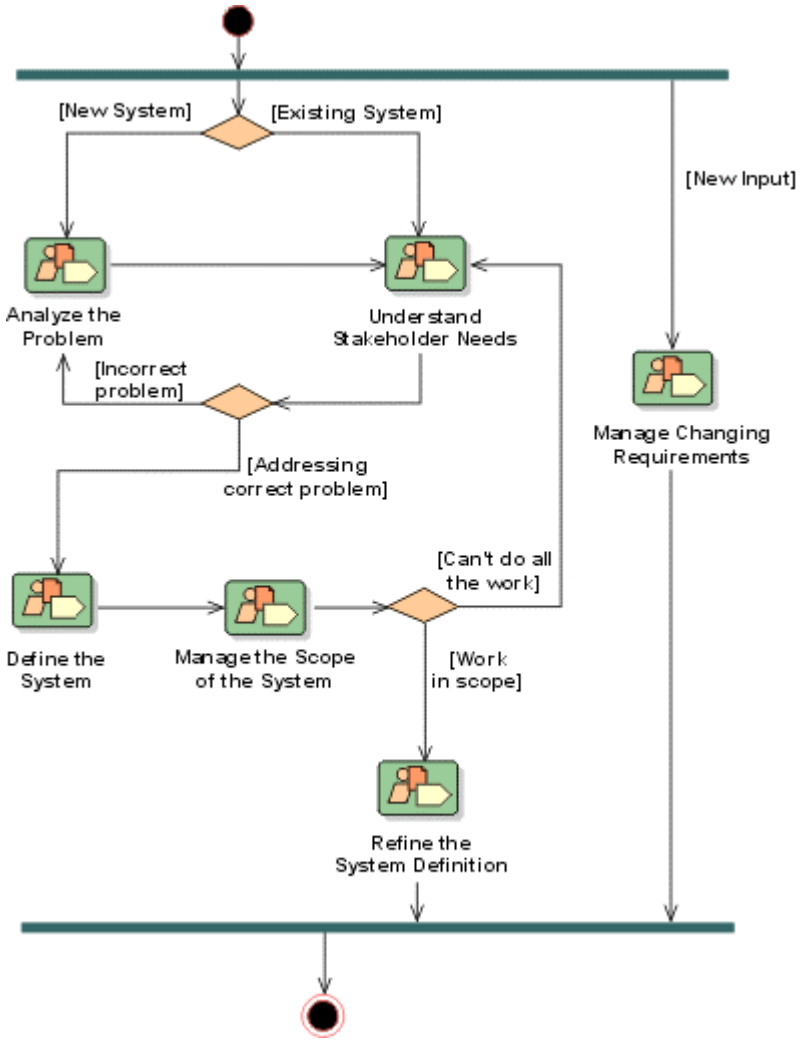

Figure 2 : Requirements : Overview

Each icon in the diagram describes a significant step in the specification of requirements. In the following sections, we will see the purpose and meaning of each icon.

3.1 Analyze the problem

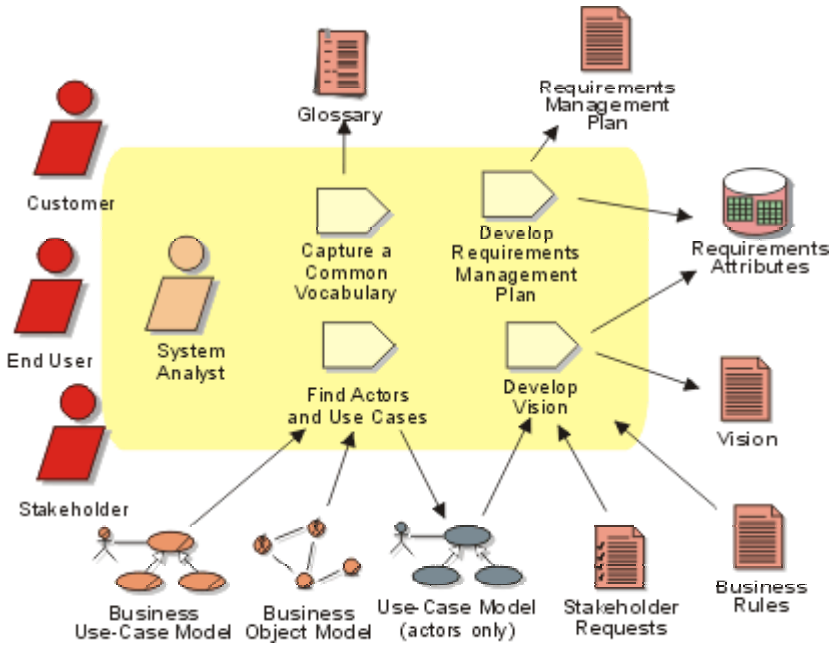

Figure 3 : Workflow Detail : Analyze the Problem

In this first step of the requirements core workflow, we want to gain agreement on the problem being solved, identify the stakeholders, define the system boundaries and identify constraints imposed on the system. *Figure 2* gives us a visual view of the different elements involved.

The first purpose of this workflow is to avoid misunderstandings among the stakeholders of the project. Before we go further, it is of central importance that all parties involed agree on what exactly is the problem we are trying to solve with our system. Therefore it is necessary to agree on common terminology which will be used throughout the project. To reach this goal we will define the terms used for the project in a common *glossary*. It is also very important to define who are the *stakeholders* of the project. Some of them will be represented by actors in the use case model.

The r*equirements management plan* will provide guidance on the requirements artifacts that should be developed, the types of requirements that should be managed for the project, the requirements attributes that should be collected and the requirements traceability that will be used in managing the product. Traceability is the ability to trace a project element to other related project elements, especially those related to requirements.

The first artifact that documents the problem analysis information is the *vision* document. It identifies the high-level user or customer view of the system to built. It describes also the main features the appropriate solution should provide. The vision document provides a complete vision for the software system under development and supports the contract between the client and the development organization.

The techniques that can be applied to define the problem are the brainstorming, fishbone diagrams or pareto diagrams.

3.2 Understand Stakeholder Needs

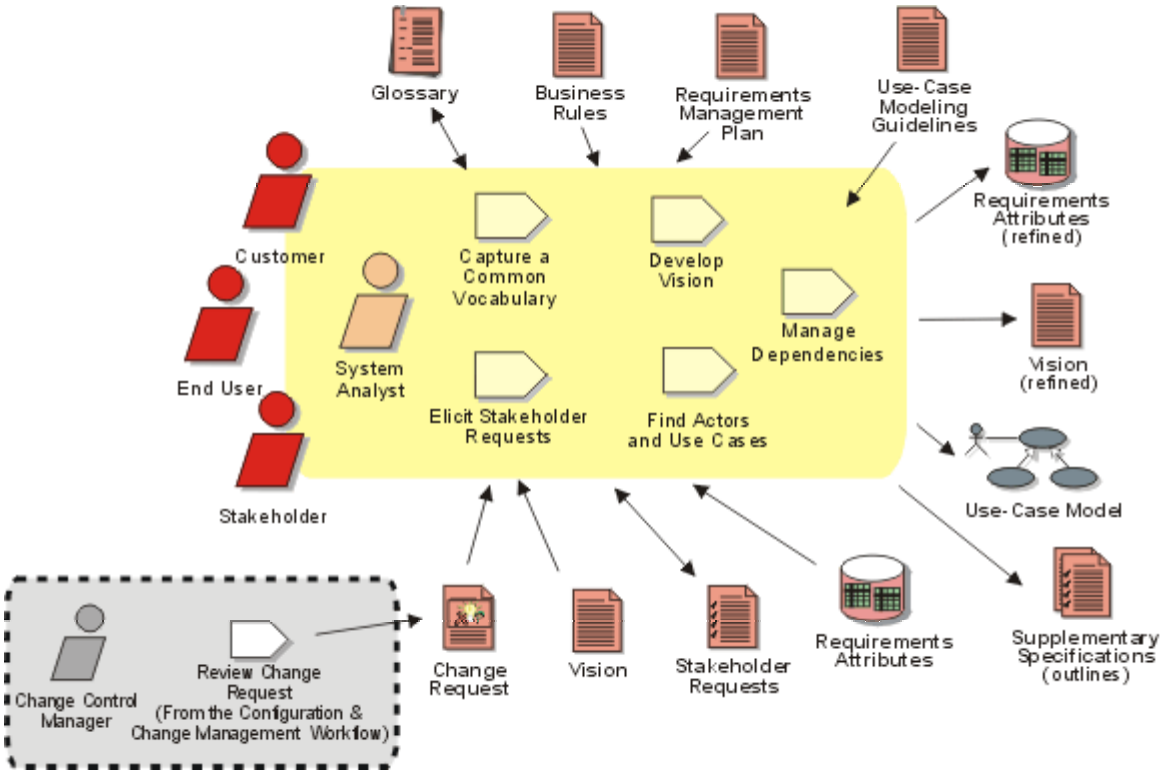

Figure 4 : Workflow Detail : Understand Stakeholder Needs

The purpose of this workflow is to collect the information from the stakholders of the project so as to understand what they really want to develop. The needs of the stakeholders can be seen as a wish list. The central activity here is to obtain the stakeholders requests using inputs like business rules, enhancement requests, interviews and requirements workshops. This will help the system analyst to define the outputs such as collections of prioritized features and attributes which will be used in the next steps of the requirements workflow (define the system and manage the scope of the system). All these outputs will result in a refinement of the vision document, as well as a better understanding of the requirements attributes. In this workflow the actors and use cases and their functional requirements will be discussed. The non-functional requirements that don't fit in the use cases should be documented in the supplementary specifications document. The glossary should also be refined and completed so as to incorporate all the terms used in the project.

The techniques that can be applied to understand stakeholder needs are interviews, requirements workshops, brain-storming, use case workshops, storyboarding, role playing and review of existing requirements.

3.3 Define the System

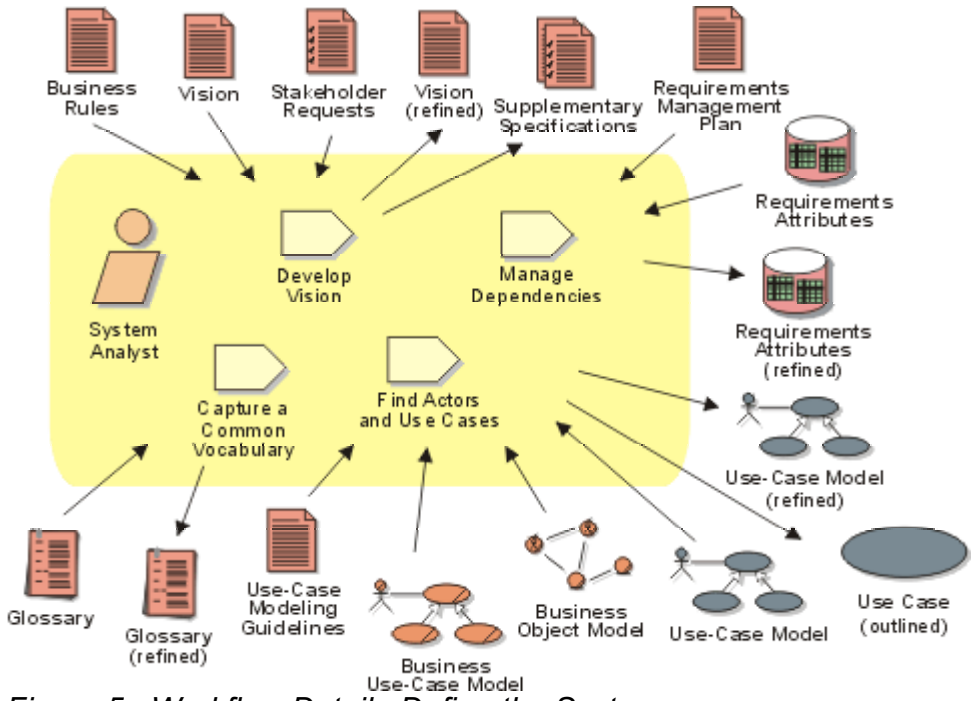

Figure 5 : Workflow Detail : Define the System

The intention of this workflow is very broad. The alignment of the project team in the understanding of the problem, the refinement of the use case model and the vision document, the analysis of the stakeholders requests and the formal documentation of these results are the main objectives of the workflow.

The following techniques can be applied : requirements workshops, use case workshops and storyboarding.

3.4 Manage the Scope of the System

The key of managing successful projects is to manage the project scope to fit the available ressources. These ressources comprise time, people and money.

One purpose of this workflow is to define the set of use cases that represent some significant, central functionality. It should also define which requirements attributes and traceabilities to maintain. The features and requirements have to be refined and prioritized.

The *software architecture document* provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system.

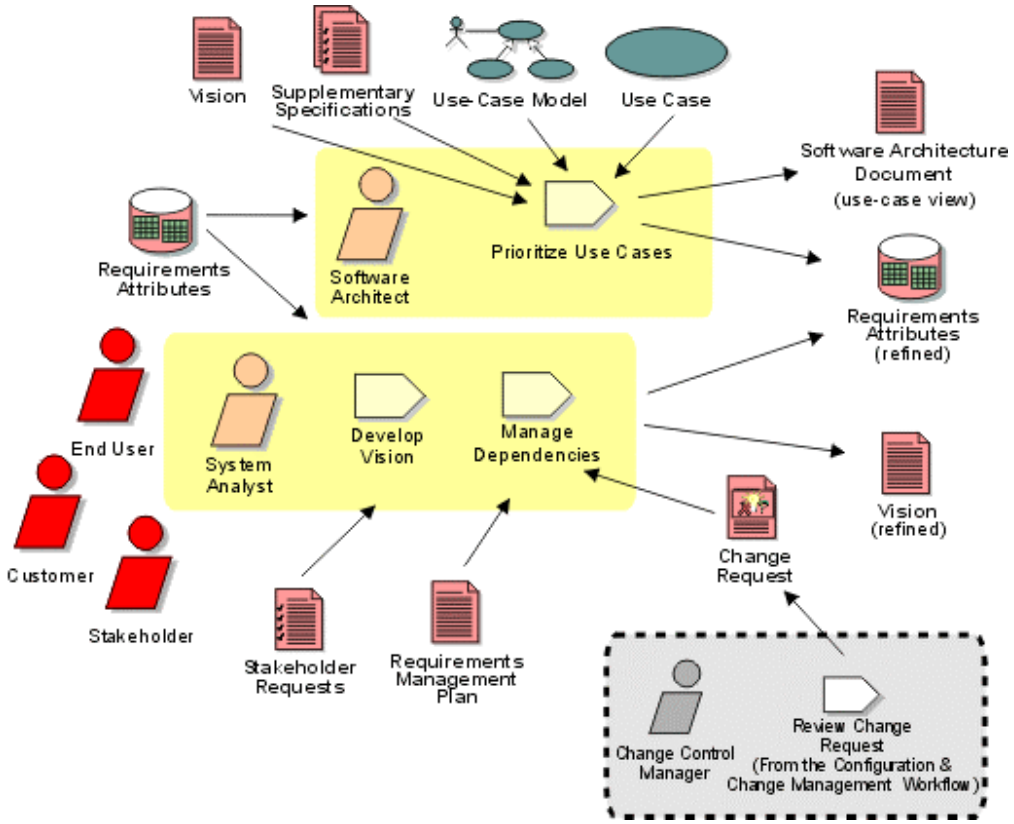

Figure 6 : Workflow Detail : Manage the Scope of the System

3.5 Refine the System Definition

The refinement of the requirements is the primary goal of this workflow. The use case's flow of events should be explained in detail. The actors should be briefly described and the use cases detailed. The final Vision document that is believed to be achievable by fairly firm budgets and dates should reflect, in its features, the realistic project scope. The supplementary specifications document should be completed and detailed. Last but not least, models and prototypes of the user interface should be designed by the user interface designer. A *use case storyboard* is a logical and conceptual description of how a use case is provided by the user interface, including the interaction required between the actor(s) and the system. The output of this workflow is a more in-depth understanding of the system functionalities.

If necessary a *software requirements specification* may be developed. It focuses on the collection and organization of all requirements surrounding the project. The check list associated with it makes sure that the team won't forget a crucial requirement and that basic issues such as functionality or performance are addressed.

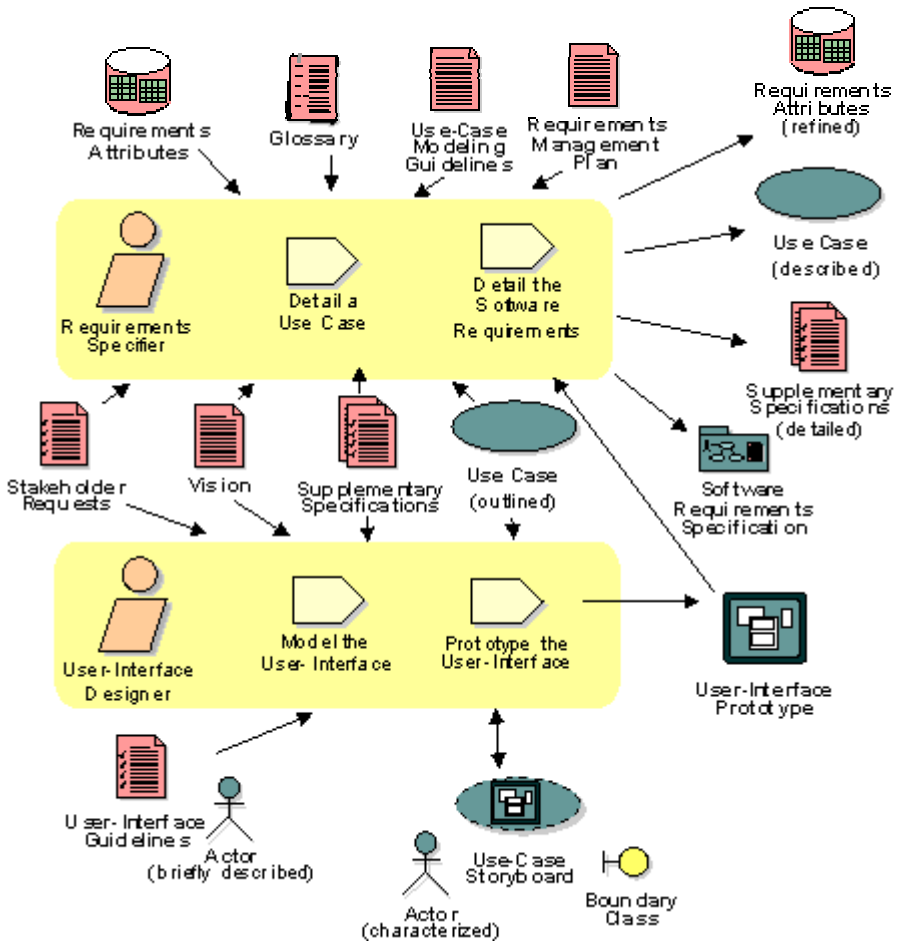

Figure 7 : Workflow Detail : Refine the System Definition

3.6 Manage Changing Requirements

The goal of this workflow is to structure the use case model, evaluate formally submitted change requests and determine their impact on the requirement set, set up appropriate requirements attributes and traceabilities, and formally verify that the results of the requirements workflow conform to the customer's view of the system. A *review record* may also be created to capture the results of the review of a project artifact.

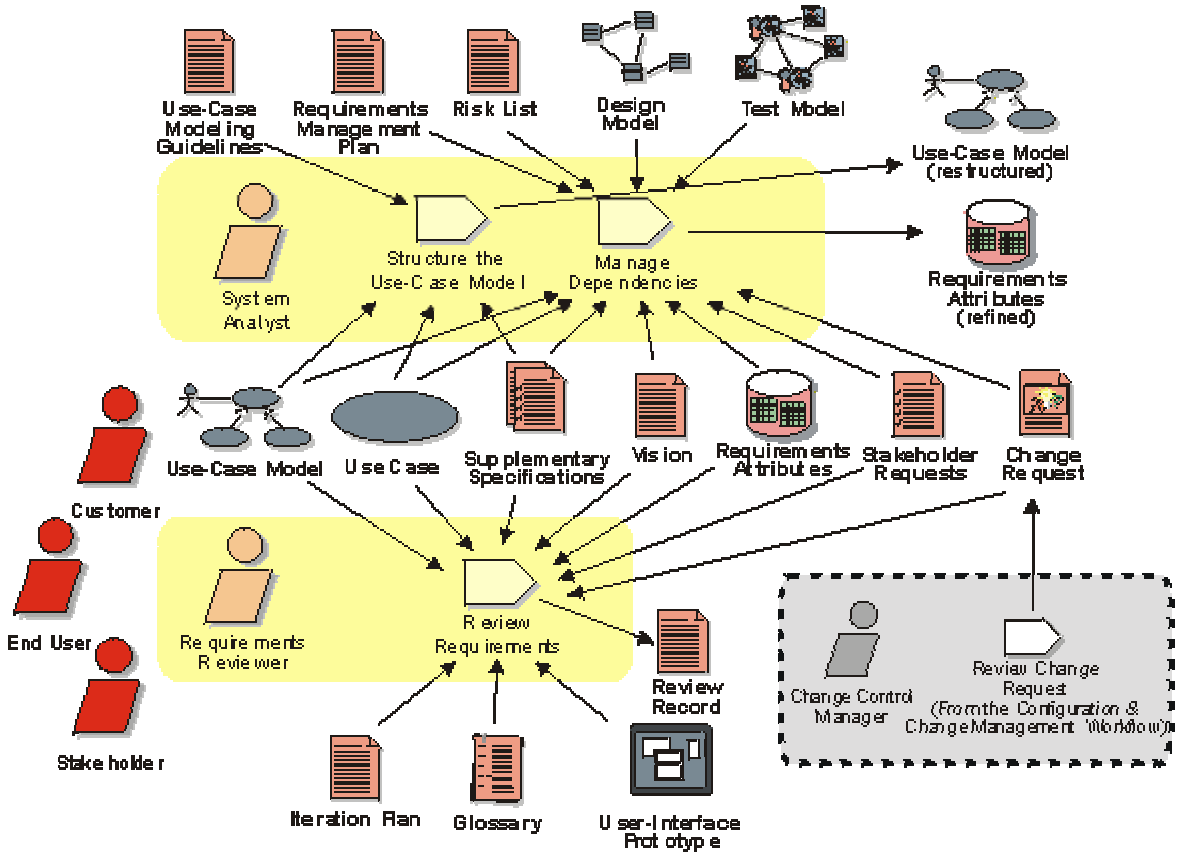

Figure 8 : Workflow Detail : Manage the Scope of the System

4 RUP in the Practice : Two Projects

4.1 Project 1 : Java-GUI for Profile Generator AMAT developed by Zühlke Engineering

This section describes the way the specification of requirements is managed using the RUP at Zühlke Engineering AG in Schlieren, Switzerland. Zühlke is an entreprise specialized in process, business and software engineering. The observations and comments resumed in this paper result from an interview held on January 16, 2001, with Micheal Hirsch, business unit leader at Zühlke. We agreed that we could define general patterns that concern every project managed with RUP when we want to specify its requirements. We will then have a closer look at the artifacts used for the specification of the requirements for the project Java-GUI for Profile Generator AMAT.

4.1.1 General patterns

First of all, Zühlke does not make use of formal methods for the specification of requirements. Michael Hirsch argues that formal methods are too expensive and that it makes the communication with the clients more difficult. For the specification they use essentially the vision document, the use case model, the supplementary specifications document and a glossary. The management of Zühlke stipulates that

the vision and the use case model are mandatory for each project, while the last two ones are only recommended.

Although the requirement core workflow is linear, the company develops the product iteratively, that means that the client and the firm agree upon a plan of the project. This plan is divided in a set of iterations. These iterations are not fixed at the beginning of the project, they evolve according to the clients' wishes. Each iteration defines a set of features (functionalities) that must be satisfied. Basically the client receives very often (every 3-8 weeks) an executable of the product that satisfies the features of the last iteration. Based on the executable he received, the client can express the features or the desired corrections for the next iteration. That means that the requirements will change at every iteration.

4.1.2 The project : Graphical User Interface for Profile Generator AMAT

The objective of the project is to develop a graphical user interface (GUI) for a profile generator named AMAT. A profile generator is a tool that enables its user to design meshes of technical objects, these meshes can then be covered with a texture giving a visual representation of the object the user wants to produce. The GUI in the project shall provide the steam turbine designer with a state-of-the-art user interface enabling him to design different objects more efficiently than this was the case before. The GUI shall interface with the software environment of the firm CCC (the real name of the client firm has been neutralized according to the client's wish). Before the development of the GUI by Zühlke, the designer did the steam turbine design by directly using the Paetra CAD tool, which is a very sophisticated and therefore slow tool. This is not very efficient since there are complex and lengthy calculations involved to generate all the data required for the manufacturing of steam turbines. During the design phase all this data is not needed anyway. For this reason, CCC developed a prototype of a profile generator tool (AMAT), which supports the design of steam turbines and analysis process without the costly overhead of dealing with complex manufacturing data. To speed up the design process even more, the existing GUI of the AMAT prototype needs to be replaced by a GUI which complies with modern GUI design principles. The strict separation of the user interface client part from the data processing server part will allow the use of Remote Method Invocation (RMI) or Corba in the future. In addition the use of Java for the user interface will result in a platform-independent implementaiton. The new GUI will also aid CCC to reach the high level goal of reducing the cycle time for the development of new steam turbines from the current 15 months down to 1 month !

Zühlke is specialized in developing solutions to such problems. Therefore it was not necessary to define the terms used in a glossary for this project. They only used two artifacts to specify the requirements for the GUI : the vision document and the use case model.

4.1.2.1 Vision Document

As mentioned earlier, the elaboration of the vision document is mandatory for every project. The definition of this document was already given in section 3.2. The

requirements and features in terms of the needs of the end user fo the Java-GUI for the Profile Generator tool. It is divided in 12 sections. While the first four sections that express the objectives and the scope of the project were already resumed in the introductory words of section 4.2, we will now handle the next sections of the vision document.

The fifth section (User Description) gives the typical profile of the end user of the GUI. Here the end user is typically defined as a steam turbine designer with a profound technical background and with some experience with software tools. The user environment is also described : the operating system supported (Windows NT), the standard monitor used (17''), etc. The key user needs are expressed in 3 points that must be fulfilled and that express the desire of the steam turbine designer to rely on (1) an intuitive and easy to use state-of-the-art user interface which complies with the CCC standards, (2) simplified analysis and (3) rapid parameter based design.

The sixth section (Product overview) presents an overview of the product. We won't treat this part in detail because of the technical complexity of such a product. The section describes the different formats acceptable to the GUI that can be read or written from/to local or remote files. The different interfaces to other products like Matlab, Altruist or Paetra already used by CCC to do the analysis and the design of objects are also sketched in a graph. The licensing and the installation through a wizard are also stated.

The seventh section (Product Features) comprises the functionalities that the GUI shall satisfy. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. These features are organized in 3 sub-sections that reflect the general, the design and the analysis features. All features are identified by an original ID (for instance Gen1 for the first feature in the 'general features' sub-section). The general features describe the general panel layout, the design options, the input and output data supported and the different startup and user modes available. These startup and user modes depend on the right of the user. The design and the analysis sub-sections precisely specify the different tools needed by the designer. A deep and thorough understanding of the specific techniques involved are necessary to be able to cope with these required features.

The eighth section (Constraints) enumerates in a straightforward fashion the constraints that shall apply to the GUI. For instance it is mentioned that the GUI must be designed in a way to allow for later RMI/Corba operation, that the Java classes interfacing the Matlab module will be provided by CCC and that the components shall be developed with reusability in mind.

The ninth section (Precedence and Priority) gives an overview of the priorities (1 to 3) and precedences between the different features.

The tenth section deals with the applicable standards, the system and performance requirements. The Java programming guidelines and the CCC general guidelines for graphical user interface have to be obeyed, the GUI shall run on Java 2 Standard Edition. It will be developed and tested on Windows NT and Sun JDK 1.2-2.

The eleventh section (Documentation Requirements) stipulates that the GUI shall be documented according to the CCC standards and that an online help shall be realized using the JavaHelp framework.

The twelvth and last section is an appendix that records the different versions of this document with their dates, authors and a description of the changes made between each version.

4.1.2.2 The Use Case Model

In the Java-GUI project the use-case model is organized in 7 sections. The scope of the document is first defined. It expresses one more time the purpose of the GUI. The second section lists the different references used to elaborate the use-case model. Beside the vision document which is the first reference for the development of the use-case model, the version of the RUP used and some books or guidelines from Zühlke and CCC are mentioned.

The third section is an overview of the use-case model, we can already recognize the actor (steam turbine designer, described in the fourth section) and the 5 use cases that compose the model.

The fifth section (Use Cases) describes in details the 5 use cases. The 5 sub-sections begin with a brief description of the use case. A normal flow of events is then enumerated. This flow describes the steps followed by a designer for the accomplishment of the assignment described by the use-case. It would be recommended here to make use of an activity diagram, which is a flow chart used to show the workflow of a system, as described by the UML notation in [3].

The sixth section is an appendix. It is a collection of screen shots and sketches of the graphical user interface. At the beginning of the elaboration of the use-case model, all the user interfaces are represented with sketches. These sketches are then removed and replaced by the real graphical interface when they are available.

The seventh and last section records the different versions of this document with their dates, authors and a description of the changes made between each version.

It is to notice that for the use-case model, Zühlke Engineering doesn't make use of the usual notation with sketches as prescribed by the UML notation.

4.1.3 Evaluation by Michael Hirsch

4.1.3.1 Pros

Hirsh estimates that the main advantage of the RUP is that it comes out of the practice. That means that it is well suited for the management of real projects. The price of RUP (CHF 1500.- per capita) can be seen as reasonable if we compare it with the costs associated with the development of an own project management

guide. Moreover, the requirement core workflow is well documented and the different artifacts are well suited to the needs of the firm Zühlke.

4.1.3.2 Cons

The documentation is poor for some core workflows like deployment, implementation and testing. But Hirsch thinks that the next versions of RUP will overcome this weakness.

4.2 Project 2 : Marconi by ISNet

This section describes the way the specification of requirements is managed using RUP at the firm ISNet and Icare at Sierre, Switzerland. ISNet is a center of competence where professors of the HES-SO (Pool of specialized schools situated in the French-speaking part of Switzerland) lead projects in various fields of computer sciences in collaboration with partners of the industry. The observations and comments resumed in this paper result from an interview held on January 19, 2001, with Yves Rey, director of the Hochschule Wallis at Sierre and member of ISNet. We will first give general patterns on how the specification of requirements is managed and achieved at ISNet. We will also introduce the reader to the project Marconi and depict the way the specification of requirements was achieved for that project.

4.2.1 General Pattern

ISNet doesn't make use of formal methods for the specification of requirements. Yves Rey thinks that these methods are not necessary for the type of project ISNet works for. Yves Rey confesses that ISNet only uses some parts of RUP like the artifacts or the general organization of the core workflows. Because of the lack of time in nearly every project, they don't have the possibility to follow all the steps of RUP. UML is used for all projects and constitutes the base notation. ISNet makes use of 3 tools to specify the requirements of a classical project : (1) Rational Rose for the use case model and the sequence diagrams, (2) Sybase PowerAmc for the conceptual database model and (3) MS Visual Basic for the generation of exemplary graphical user interfaces.

4.2.2 The Project : Marconi

The goal of this project is to develop an application for the management of technical documents for an industrial company. The client (Bobst SA, Mex, Switzerland) has specialized into the development, the manufacturing, the sale and the service of machines dedicated to the folding and glueing of flat and corrugated cardboards. As a world leader it needs an application that manages all the documents (assembly plans, different versions, etc.) concerning assemblage of technical machines. A second internal goal of the project for ISNet was to gather enough information and know-how from Bobst SA to be able to develop a meta-model for the management of documents in an industrial company. ISNet used a range of artifacts to specify the

4.2.2.1 The Vision Document

The vision document for the project Marconi is subdivied into 8 sections. The first section gives a brief introduction to the objectives of the project, the tools used and the partners involved. At the beginning of the project it was obvious that a deep understanding of the way technical machines were assembled by Bobst SA was necessary. A template of a typical document and the understanding on how and who would write or read a document had to be obtained. The different sections of the vision document reflect these needs.

The second section defines which actors are involved in the creation and writing of a document, and which actors use them. The redactor, the foreman (contremaître), the team head (ChefEquipe), the assembler (monteur) and finally the worker responsible for the fixing of the machine (SAV) are defined. The information flow between these actors and the application is also sketched in the following figure :

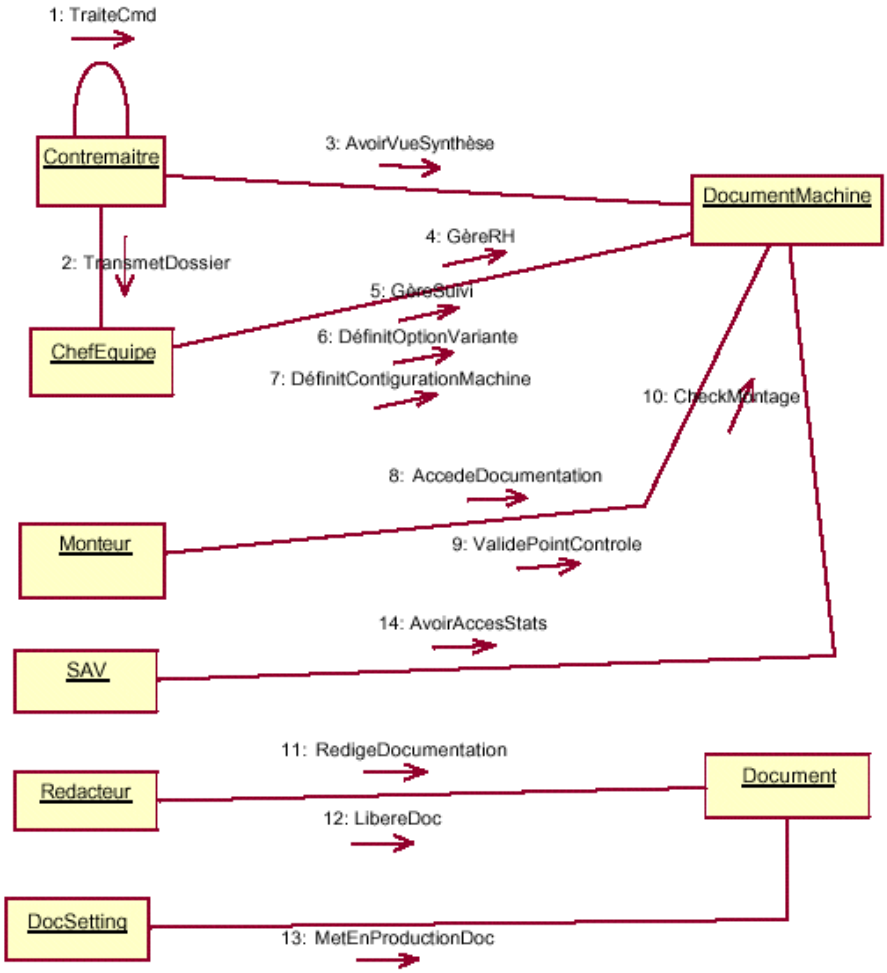

Figure 9 : Global Vision of the Interactions between Actors and Application

The third section is an analysis of the structure of a document and the structure of a machine. A meta-model of the components that compose a machine is sketched. It

the identification of the different machines had to be taken into account. Some user interfaces are inserted to give an idea of the look-and-feel of the application for the actors. The definition and a meta-template of a document is given in this section. A document for Bobst SA is defined as a structured information detailling the different phases of the assemblage for a machine. It is also specified that the assemblage of all the machines made by Bobst SA must be based on a related document. Because of the international significance of Bobst SA, a translation of the documents created may be necessary for some countries. As the translation will need external help, the use of XML to send the content of the documents is mentioned.

The fourth section describes how the commands from clients are managed and how the application should enable the foreman to follow the advancement of the production of a specific command or of his entire group. The assembler working on a machine follows the steps for the assemblage of this particular machine. These steps are described in a related document on a graphical user interface. When a step is done, the assembler shall check it on the graphical user. An aggregation of the advancement of the production can therefore be observed and the foreman has a 'live' vision of the production, what was not the case before.

The fifth section concerns the conceptual database model done with Sybase PowerAmc. Since this model is confidential, we won't show it in this paper.

The sixth section gives an overview on the modules that compose the application and that will have to be implemented. Figure 10 gives an overview of these modules.

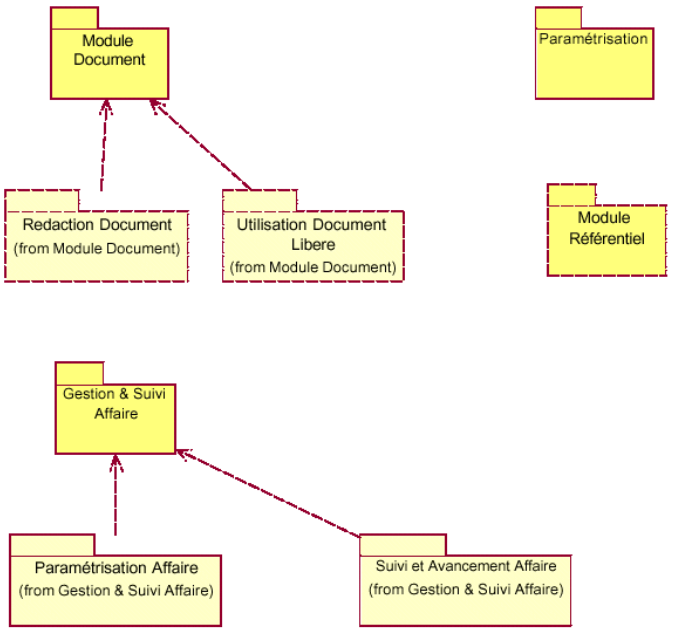

Figure 10 : Definition of the Modules

The seventh section details the functionalities that the application shall satisfy. The functionalities are expressed as use cases. A use case model for each module is sketched using the UML notation [4]. Figure 11 illustrates the use case model for the module Document which shall implement all the functionalities for the management

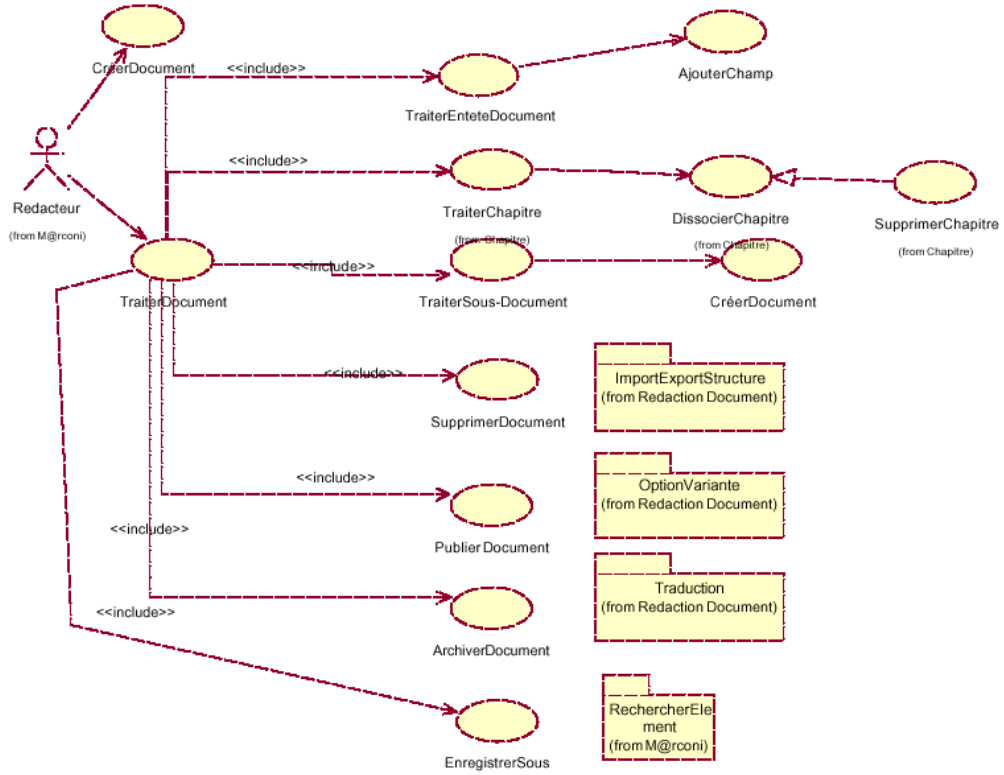

Figure 11 : Use case for the Module Document

Then each use case is detailled with a sequence diagram. Figure 12 illustrates the sequence diagram for the creation of a document.

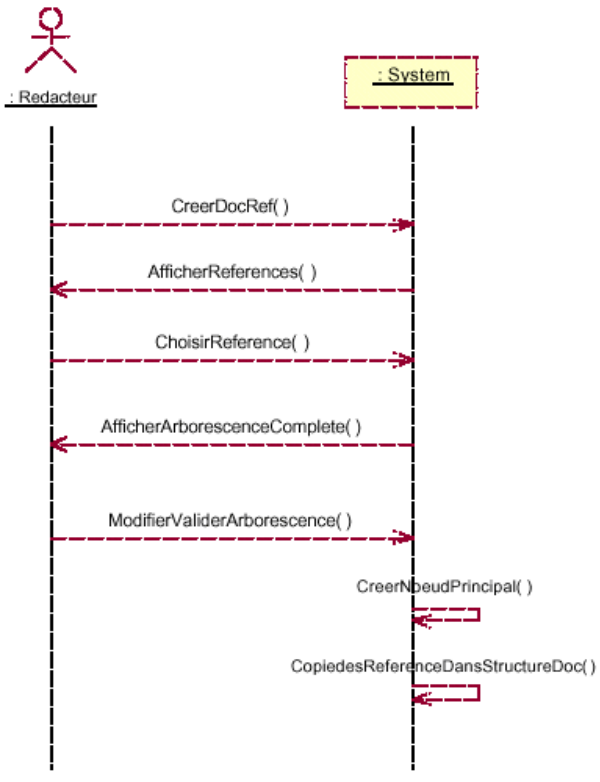

Figure 12 : Sequence Diagram for the Creation of a Document

The eighth and last section of the enhanced vision document is a glossary of all the technical words used in the vision document.

4.2.3 Evaluation by Yves Rey

4.2.3.1 Pros

Yves Rey thinks that RUP is very easy to introduce in a company and that the proposed artifacts are easy to use and boost the development time of a project after adaptation. The cost of RUP is definitely not a problem for ISNet because the members can take advantage of their relationship to the Hochschule Wallis to pay a reasonable price for it. For the specification of requirements, RUP is a very interesting process for projects that are not of strategic importance.

4.2.3.2 Cons

The first projects managed with RUP take much more time than those managed with a known process because of the time needed to read all the documentation and to fill all the artifacts out. Yves Rey thinks that with some experience many artifacts are not very usefull in some projects, but it takes some time to get used to it and time is a crucial factor for every projet. Therefore Yves Rey understands the firms that do not want to invest time and money in RUP when they already have a know-how in other methods or processes for the specification of requirements.

5 Conclusion and Evaluation

5.1 Strengths of RUP and of the Requirement Core Workflow

RUP is used by a growing number of companies in the software development branch. The Rational Software Company has made, and continues to do so, a significant investment in their Rational Unified Process product. Its web-based online e-coach can easily be introduced in the company and is documented with some universally available books (see the Literature). RUP provides several mechanisms, such as a working prototype at the end of each iteration and the decision point at the end of each phase, which provides management visibility into the development process. Since it relies on a widely used notation (UML), it renders possible a more easy dialog with the stakeholders and it is therefore an interessant help in the specification of requirements.

5.2 Weaknesses of RUP and of the Requirement Core Workflow

RUP does not cover the entire software process: it is very obviously missing the concept of maintenance and support. It is only a development process and it does not explicitly support multi-project infrastructure development efforts such as organization-wide architectural modeling, missing opportunities for large-scale reuse within the organization. RUP is currently weak in areas such as reuse management, people management, and testing. The biggest weakness of RUP, especially for the specification of requirements, is that it is too easy to interpret it as cookbook. That means that for the same project a complete different way to specify the requirements can be chosen depending on the habits of the project leader. If Zühlke Engineering

Zühlke Engineering and ISNet use only some parts of RUP, especially some artifacts, and that they don't follow all the steps and that they don't create all the artifacts defined by RUP. This observation means that RUP, as a whole, is not used in these organisations.

6 Synopsis

The purpose of this paper was to give an overview of the Rational Unified Process in general and its requirement core workflow in particular. We first recalled the basic concepts of RUP and then detailed the entire requirement core workflow. After this theoretical part, we gave an overview of two projects: the Java-GUI for Profile Generator AMAT project by Zühlke Engineering and the Marconi project by ISNet. Then we showed how the specification of requirements was managed for these two projects and which artifacts were used by the firms Zühlke Engineering and ISNet. Each project director gave his opinion on the strenghts or weaknesses of RUP. Finally the author gave its own opinion not only on the quality of RUP for the specification of requirements but also on the way Zühlke Engineering and ISNet make use of it in this particular domain.

7 Literature

[1] The whitepapers of the Rational Company : *Rational Unified Process Whitepaper: Best Practices for Software Development Teams;* http://www.rational.com/products/rup/whitepapers.jsp

[2] The whitepapers of the Rational Company : *The Ten Essentials of RUP: The Essence of an Effective Development Process;* http://www.rational.com/products/rup/whitepapers.jsp

[3] Kruchten, Philippe; *The Rational Unified Process : An introduction*; Addison-Wesley Pub Co; Essex, England; 1998.

[4] Quatrani, Terry; Visual Modeling With Rational Rose 2000 and UML; Addison-Wesley; Reading, Massachusetts, USA; 2nd printing; 2000.