

Using RUP for Process-Oriented Organisations*

João M. Fernandes¹ and Francisco J. Duarte²

¹ Dep. Informática, Universidade do Minho, Braga, Portugal

² Blaupunkt Auto-Rádio Portugal, Braga, Portugal

Abstract. In this paper, based on a proposal of a reference framework for process-oriented software development organisations, we discuss some issues related to the usage of the RUP's disciplines as instantiations of those processes. The proposals made in this paper were consolidated based on experiences from real projects. The respective lessons learnt from two of those projects are reported.

1 Introduction

Organisations are moving from a hierarchical structure, sub-divided by departments, to a model where multi-disciplinary teams run horizontal business processes that cross all the organisation. Thus, to develop software applications for this type of organisations two topics must be taken in consideration: (1) the software development process must be appropriate and controlled; and (2) the organisational platform where the processes run must be modeled and considered. The structure of the software development organisation has a major impact on the final quality of the software system being created.

This paper uses a reference framework for process-oriented software development organisations, presented in [1], that instantiates processes as RUP's disciplines. This framework was used in 2 projects and we show the lessons learned from them.

The structure of the paper is the following. In section 2 a generic reference framework for process-oriented organisations that focus its activities towards the needs of the clients is introduced. In section 3, based on that generic framework and in the RUP's disciplines, we detail it in order to describe the specific case of organisations that develop software. The Business Process Modeling is explained in section 4. Finally, on section 5, 2 case studies are presented, including the results analysis and lessons learned.

2 Generic Framework for a Process-Oriented Organisation

Business processes can be seen as a set of tasks executed to achieve a given business goal. Within the business process, human resources, raw material, and internal procedures are combined towards a common objective. An enterprise can be viewed as a group of competence centres, which share the same goal, whenever a process is instantiated [2].

In this type of organisations, the members are no longer dependent on an hierarchical chain of command. Each member must have expertise in one area where he performs

* Work partially supported by Fundação para a Ciência e a Tecnologia (FCT) and Fundo Europeu de Desenvolvimento Regional (FEDER) under project "METHODES: Methodologies and Tools for Developing Complex Real-Time Embedded Systems" (POSI/37334/CHS/2001).

his functions. For that, he is helped by a coach to improve his capacities/skills and by a process owner to co-ordinate his activities with the other team members.

Generically, an organisation exists to supply a set of products or services to its clients. To achieve this, the organisations need to execute a set of internal activities. Actually, organisations do not exist isolated, but are part of markets, where other organisations can be their competitors, suppliers or clients. The creation of wealthy, within private organisations, or the supplying of products or services with social impact, within non-profit organisations, follow the same underlying principle: the fulfillment of the clients' needs. Thus, these needs and the expectations of the clients are vital for defining the internal structure of an organisation.

A process inside an organisation can be viewed as a set of activities that has as inputs a set of services and/or materials. It also has as outputs a set of services and materials. This view must be oriented towards the necessities of the clients and the creation of added-value. This implies that the clients' requirements must always be considered, both during the development and the performance of the system.

In an organisation, there are other processes rather than those that provide added-value to the clients. Within an organisation, the management by processes requires a structure that differs from the typical functional hierarchy. It is mandatory to synchronise the processes among them and to fulfil the strategic objectives of the organisation. Thus, for a process-oriented organisation, a structure with the following components should exist [1]:

Process Management Top Team: This team includes the top managers and all processes owners.

Process Godfather: For each process one godfather, which must be a top manager, should be designated.

Process Owner: For each process, an owner is needed. He must have know-how on managing processes and persons, but also competency in the areas associated with the process.

Multi-Disciplinary Team: These teams must be created for each added-value process. Considering the organisation size and its strategic objectives, multi-disciplinary teams may also be established for management and support processes. This decision must be taken by the process owner and godfather.

Execution Teams and Team Leaders: These teams and their leaders represent the instances of a given process. Therefore, during the execution of a process, some teams will use it with a specific focus. For example, for a given production process, one team may be responsible for producing parts for industrial clients, while other team may produce them for individual clients. This partitioning must be managed by the process' multi-disciplinary team, but the team leader is supposed to suggest its composition. Each leader must obey the definitions and use the indicators of the process which he is responsible for.

To align a process-based organisation with its strategic objectives, it is crucial that the goals are based on the organisation's mission and vision, and also on its principles and values. Based on those strategic objectives and in the business plan, the priority when deciding the key processes within the organisation can be perceived.

After completing these stages, highly connected to the top management, we propose the following pragmatic procedure to introduce the processes in an organisation: (1) Define the processes: identify the collection of processes in an organisation and decide on the relative importance of each one; (2) Define the sub-processes and tasks: each process is decomposed into sub-processes, until we reach a level of detail where only primitive activities remain; (3) Identify the interfaces among processes; (4) Assign the owner and godfathers to the processes and the members to the multidisciplinary teams; (5) Define the goals of the processes; (6) Define the key indicators; (7) Define the processes' team leaders and executing teams; (8) Measure and monitor the processes; (9) Execute corrective actions; (10) Review and continuously improve the processes.

The identification of the key processes within an organisation must be accomplished with the help of criteria. Some of those criteria are: (1) Identify if the process traverses several functions; if this is not the case, we might have an activity or a task; (2) Verify if the process is measurable and is integrated in the business plan; otherwise, the process must be terminated, since it is not aligned with the organisation's strategy; (3) Verify if there is a focus towards the clients and their requirements; when this does not happen, the process may be inefficient; (4) Determine the importance of the process for the final quality of the product; (5) Determine the importance of the process for the fulfilment of the organisation's mission; (6) Verify if the process is multi-disciplinary; if that is not the case, we may have an activity or a task; (7) Find out the importance of the process to the success of the organisation.

Based on these criteria, we can select which processes of a given organisation will have a process-oriented management and the activities and tasks they will be composed of. Some processes, activities, and tasks will be eliminated since they do not add any value to the clients, neither to the organisation. These eliminated (or redefined) processes, activities, and tasks and their respective consequences in terms of reorganisation and impact in human resources are the essence of re-engineering [2].

Besides this general framework, inside each process category (management, support, and added-value), there are several processes that define in more detail the types of activities that exist inside organisations (fig. 1.a). The business processes are designed with the aim of bringing value to the organisation. This value must be quantified to serve as a basis to trace the process state and possible improvements. Some general indicators for each process were proposed [3]: productivity, added value, cycle time, queue size, quality indices.

In the management category (top of fig. 1.a), 3 business processes are proposed:

- Global Strategy (GS): Within this top-level management process, potential clients are evaluated, the innovations are planned, and the general policy for the enterprise, such as its mission, its vision, its principles, its values, and its long-term objectives, is defined.
- Policy Deployment (PD): The organisation's policy and short-term objectives, usually defined in an annual basis, are unfolded and delivered to the competence centres and processes.
- Business Plan (BP): The plan and the budget of the activities and their distribution among processes and competence centres are defined. Normally, these activities are

planned with an annual scope; the activities for controlling the execution of the plan are also defined.

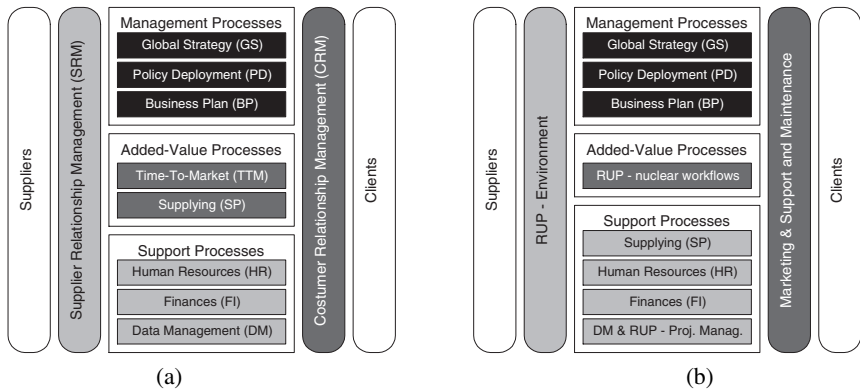


Fig. 1. (a) General composition of the processes within a generic organisation; (b) General framework for the processes of organisations that develop software.

In the added-value category (center of fig. 1.a), there are 3 business processes:

- Time-To-Market (TTM): This represents the development process associated to a new product and the environment needed to produce it.
- Supplying (SP): This process occurs normally after the TTM for the same product, and starts with a request from the client, or with a decision of production. It ends with the deliver of a product to the client with the fulfilment of the required schedules and quality levels.
- Customer Relationship Management (CRM)¹: This process includes all activities that allow clients to be satisfied with products or services supplied to them.

In the support category (lower part of fig. 1.a), 4 business processes are suggested:

- Supplier Relationship Management (SRM)²: This includes activities such as the selection of the materials' suppliers, the execution of audits to verify the quality, or the establishment of agreements for just-in-time delivers.
- Human Resources (HR): This process includes all issues associated with the well-being and qualifications of the organisation's collaborators.
- Finances (FI): All the activities related to fiscal issues belong to this process.
- Data Management (DM): This process includes activities associated to data quality and the indicators of the organisation, the security of data, the priority of computer related projects and their budget control, data-warehousing and data-mining.

¹ Although the CRM process is mainly seen as an interface process, it was also included in the added-value category, because client-oriented organisations must address the customer needs.

² The SRM process is mainly classified as an interface process, but it was also included in the support category, because it generates inputs for added-value processes.

The activities related to the continuous improvement, to the quality and environment management including certification tasks, are all examples of activities that tend to be confused with processes. These activities are part of all processes, such as those of fig. 1.a, and should not have a separate management.

3 Framework for Organisations That Develop Software

Since the framework presented in fig. 1.a describes a generic organisation, it can also serve as a basis for modelling an organisation that develops software. Being a more specific type of organisation, we can add more details and propose more specific processes to the processes presented in the generic framework.

We propose the instantiations of the business processes within the organisations that develop software (fig. 1).

Therefore, the processes named Global Strategy (GS), Policy Deployment (PD), and Business Plan (BP) are equivalent to those of any other organisation.

Since software is an intangible product, it is obvious that no raw materials are needed to produce it. For organisations that develop software, the SRM process is instantiated by the RUP's Environment discipline, since it furnishes the working environment (e.g. development tools), which will be used by the teams, and the development guidelines to be followed within the organisation.

The Supplying (SP) process consists essentially in creating copies of an application. In contrast with more traditional industries, where it represents probably the most important process, in software, due again to its intangible nature, this is not an added value process. Usually, the kind of materials used to support it, DVD and printed manuals, imply that external suppliers are hired.

The Time-To-Market (TTM) process is mapped into the set of RUP's core disciplines: Business Modeling, Requirements, Analysis and Design, Implementation, Test, and Deployment. This set of activities, or sub-processes, run in parallel for the same development project [4]. In our opinion, this is the most critical process for an organisation that develops software.

The Human Resources (HR) process for software factories is the same as for other types of organisations. We must however point out that software development requires highly-specialized people, being their hiring a critical issue for the success of the organisation. It is impossible to produce quality software without skilled people.

The Finances (FI) process is the typical fulfillment of the fiscal obligations, which is common to all types of organisations.

The Customer Relationship Management (CRM) process is decomposed into the sub-processes: Marketing, and Maintenance and Support. This ensures that, when a software application is delivered to the final clients, its life-cycle does not end, but instead continues with this process, incorporating changes and corrections and providing training to the users, until the application is no longer used by the client. The Marketing assumes, in this case, a form similar to that observed in other types of organisations.

The Data Management (DM) process, in what concerns the data related to the RUP's core disciplines, is mapped into the RUP's Project Management discipline. In this discipline, some activities lead to the production of indicators of the project status. Its

existence is the foundation to take decisions based on facts, related to the progress of the project and also to adjust and improve the software development process.

To reach the highest CMM levels [5], the continuous improvement of the development processes must be part of each process, instead of being a single autonomous process.

RUP's core disciplines, that in fig. 1.b implement the added-value process, are subdivided in activities, which can be viewed as sub-processes. The description of those sub-processes is made with UML activity diagrams, complemented optionally with other type of diagrams, such as interaction diagrams and business object diagrams. This representation is also valid for all other processes of a generic organisation, as illustrated in fig. 1.a. Whenever an organisation that develops software executes a development process, the TTM process will be executed. Since we are proposing this process to be implemented by the six RUP's core disciplines, it implies that the Business Modelling discipline can also be executed. This discipline strongly recommends the usage of activity diagrams (and other complementing diagrams) for modelling purposes. Therefore, the target organisations, as proposed in fig. 1.a, will be modelled with a collection of these diagrams. In parallel, within the organisation that develops software, since the Business Modelling discipline is a sub-process of the TTM process, it can also be modelled by activity diagrams.

4 Business Process Modeling

In this section we concentrate on 'Business Modelling', because of the six RUP's core disciplines (Business Modelling, Requirements, Analysis and Design, Implementation, Test, and Deliver), it is the one most directly related to the client's needs. During the development of software, all the stakeholders must have a common understanding of the business processes that exist in the target organisation. This reality is not circumscribed to the obvious organisational information systems, but can also include real-time embedded systems if they affect the business processes.

Within RUP, the business processes model is obtained in the Business Modelling discipline. The main activities in this discipline are centred around the identification, refinement, and realisation of the business processes and in the definition of the roles of people associated to the business. Each role in this RUP's discipline has under its responsibility the execution of several activities that will have as deliverables several artifacts (tab. 1). For example, the activity 'Refine Business Processes' includes the activities 'Structure the business use case model', 'Review the business use case model', and 'Detail business use cases'.

Among all activities and their respective artifacts, only some of them are mandatory. This flexibility permits the configuration of RUP, so that it can be adapted to a specific project executed in a specific organisation. Thus, taking into account the options made when choosing the artifacts, the following set allows modelling business processes [4].

Business Vision: This artifact captures the goals of a particular business modelling activity. It should be stated 'Why' and 'What' are to be modelled. It also serves as an oracle to all future high level decisions.

Business Glossary: In this artifact all business terms and expressions are kept.

Business Rules: The business rules correspond to policy statements and conditions that should be fulfilled, from the business perspective. They are similar to systems' requirements, but they focus on the business core, expressing rules related to business, but also its architecture and style. Its modeling must be rigorous, being one possibility the usage of the Object Constraint Language (OCL) as specified in UML [6]. Alternatively, using a natural language in a structured form is also admissible, since even though there are clear advantages in using a more formal approach, the need to allow its understanding by all the stakeholders is usually paramount.

Business Use Case Model: The main goal of this artifact is to show how the business is being used by all stakeholders. This is achieved by modeling the business processes and their interactions with external parties, based on use case diagrams (with stereotypes for business use cases and business actors) [7]. The business processes are associated to a discipline that needs to be modeled, and that specify how added-value is created for the business actors. This modeling can be supported by activity diagrams, possibly extended with the representation of organisational units interfering in the business process and with the distribution of the activities by those organisational units.

Table 1. Roles, activities and artifacts for business modeling in RUP.

Roles	Activities	Artifacts
Analyst of the Business Process	Verify target organisation Establish and adjust objectives Capture the vocabulary of the business Find business actors and use cases Maintain the business rules Structure the business use case model Define the business architecture	Business rules Business use case model Business Glossary Business object model Business vision Supplementary business specification Target organisation verification Business architecture
Reviewer of the Business Model	Review the business use case model Review the business object model	
Designer of the Business	Detail business use cases Find business workers and entities Define the automation requirements Detail business entities Detail business workers	Organisational units

5 Case Studies

The proposals made in this paper are being introduced in the second author's organisation, in what concerns the organisation's structure and the adoption of a RUP-based process and UML. At the moment, two software projects were conducted following the approach suggested here with promising results, and we next report on some of the lessons learned from these projects.

The first project (Travel Management) allows to validate the importance of using adequate methods for capturing requirements and for modelling them using a notation that clients can easily read. It also helps to understand the challenge of adopting standard software. In the second project (Premium Wage), the selected development process is more extensively used and we evaluate the capacity of the process to cope with complex organisations. The final application requires the complete re-engineering of some business processes, the extension of functionalities into new business processes, and finally the design and implementation of a new business process and its supporting application.

5.1 Travel Management

The first project was the development of a travel management system. The need and motivation for implementing a computer-based system to manage the travels, mainly their planning and the payment of the respective expenses, is based on the fact that previously this process occurred quite often, but had no automatic support. These business processes, although frequently executed, do not belong to the set of core processes.

This project had a requirement from the target organisation of being implemented in the SAP R/3 ERP platform. This imposition derives from the strategic decisions made by the top managers of the target organisation:

- Minimization of the number of distinct computing platforms within the same organisation.
- Reduction of the technical knowledge needed to accomplish maintenance activities.
- Reduction of the number and types of tools that the final users need to know.
- Improvement of the specialization level, and consequently the performance, of the IT personnel and final users.
- Economic advantages obtained by negotiating a big quantity of software licenses.
- Support advantages. The organisation is not a small client of several suppliers, but is instead a big client of only one supplier. This advantage must be weighted against the risk of dependency.

This project was chosen by the organisation that developed the software to introduce the UML graphical notation, with the following motivations:

- To improve the quality of the documentation from previous projects (incomplete textual descriptions in scope and detail).
- To improve the inter-communication among all the stakeholders.
- To improve the rigor of the documented information.
- To augment the quality of the final product, being this understood as the degree of fulfillment of the client's requirements.
- To ease the support for the final application.

Typically, ASAP [8] is adopted as the development process in SAP projects, but in this project we decided to adopt RUP. Although ASAP is adequate for SAP-based solutions, it has some weak aspects that are not desirable in modern processes, namely the fact that the current (As Is) and the future (To Be) situations are mostly described in a free textual form, and also the difficulty in managing the requirements change.

Results Analysis. We next analyze the way RUP and UML were used in this project. The main positive points that were identified on the execution of this project are the following:

- It is mandatory to assess the risks and the respective correction and contention actions. This activity reduces the occurrence of the potential problems and also minimizes their negative impacts when they do occur.
- The effort to design and implement the architecture of the final solution was small, since it was already available and documented.
- The introduction of RUP, in an organisation with a maturity level of its software development process not higher than CMM-3 [8], requires developers to understand that developing software is not only writing code, but includes also other activities that have an important impact on the quality of the final product.
- The quality of the documentation improved in quantity, detail, and rigor, being also an example for the rest of the organisation and allowing a better maintenance and support.
- The stakeholders meetings improved in productivity. Quicker results (requirements definition) were achieved in less time.
- The client's requirements were satisfactorily implemented.

As negative points, we identified the following ones:

- It is difficult to map the requirements on the functionalities already available in the standard ERP software.
- It is necessary to have a deep knowledge about the standard ERP software's process reference model.
- Some members of the development team try to resist to the introduction of a new process, such as RUP.
- The lack of a study about the economic impact of an application (i.e. its return on investment) induces some clients to just evaluate the costs, when the application is delivered. This fact is even more evident whenever the target architecture is standard software.
- If the quality assessments are informally conducted and without the participation of all stakeholders, the project status may be imprecisely perceived.
- RUP business modeling was not conveniently executed. The AsIs description was made in text (based on a already available guideline) and the ToBe situation was described only with use cases and their realisations in the final system (not in the business).

Lessons Learned. The main lessons learned with this project were:

- The adoption of standard software solutions (like ERP) implies an extra activity of conflict negotiation between the client's needs and the functionalities offered by the solution. Although it is possible and not difficult to implement changes to a ERP, usually this is not a good idea since it implies that in the next upgrade of the product all these customised solutions have to be evaluated to be transported, discarded, or

changed. The conflict negotiation can only be conducted by software development team members, because typically the clients do not have a deep knowledge about the ERP's functionalities and customisation.

- RUP introduction should be made in a project where the high level artifacts (e.g. business use case) have an easy and understandable transposition to lower level artifacts (e.g. source code). Without this attribute, RUP can be misunderstood either by clients and development team, and be wrongly seen as a paper generator.
- RUP introduction should not be made by hierarchical (from top to bottom) imposition, but through presentations explaining its potentials. This helps the development team to have a proactive behaviour during the usage of RUP.
- It is impossible to introduce RUP without a training effort both in UML and RUP.
- The target organisation has a deep influence on the software development organisation. Problems like the replacement of key members and the inexistence of complete definitions for requirements, which are under clients' responsibility, may have a negative impact on the software development organisation.
- Quality verifications (like the milestones for phase transition inside RUP) should be formally executed and documented, and being carried out with the presence of all stakeholders.
- The Business Modelling discipline is vital for the success of the final application and should be executed in all projects that involve business applications.

5.2 Premium Wage

The second project was the development of a software system to calculate the payment of extra money to employees, based on their productivity. This project was classified as critical, since it may have important social and behavioral impacts on the organisation, if the amount is badly calculated or if it is impossible to explain how it was obtained. The adoption of this system is to improve the organisation's overall productivity.

Besides its criticality, the business process is also complex due to its dependency from other processes. In the present case, the payment of a premium depends on three main factors: individual absenteeism, quality of the products produced in his line, and individual performance. The first two sub-processes were extended in order to support new functionalities required. The third, a complete reengineering was carried on. Finally, to the premium wage values calculation a completely new process was designed, modeled, and implemented.

In the project the selected development process is extensively used and we evaluate the capacity of the process to cope with complex organisations. The final application requires the complete re-engineering of some business processes, the extension of functionalities into new business processes, and finally the design and implementation of a new business process and its supporting application.

Business Modeling. The technology-planning horizon for big companies is now a synthesis of software engineering and process engineering [9]. Additionally, the success of a project depends heavily on the correct perception of the business process to be modeled. Taking into account these two aspects, the RUP's Business Modeling discipline

assumes a critical role in the software development process and therefore a special attention is paid to it in this section.

This discipline can generate the artifacts presentd in table 2, and has as objectives the following:

- To understand the structure and dynamics of the organisation where the system will be executed.
- To comprehend the current problems of the target organisation and to identify potential improvements.
- To assure that clients, final users, and developers have a common understanding about the target organisation.
- To capture/deduct the requirements of the system necessary to support the target organisation.

The procedure taken for configuring this discipline in this case study was was also adopted for describing the other RUP's disciplines.

One way of parameterising RUP is by choosing which artifacts to use and their level of detail. On table 2, we show which artifacts on the Business Modeling discipline were used in the project and the arguments considered to decide on its usage. This choice was validated by the quality assessments. Both the subset of used artifacts and also its degree of detail can not be anticipated with rigor, but must be selected based on experience and knowledge of the development team in relating the characteristics of each project with the functionalities offered by the artifacts.

Table 2. Artifacts in business modeling.

Artifact	Used	Motivation
Business Rules	yes	Rigor in describing a new business process
Business Use Case Model	yes	First description of the functionalities and the organisation's business actors
Business Glossary	no	The business terms are common to the target and software development organisations (they are sub-organisations of the same organisation)
Business Object Model	yes	Realization of the Business Use Cases
Business Vision	no	The business vision is common to the target and software development organisation
Supplementary Business Specification	no	Business Use Case Model and the Business Object Model are sufficient
Target Organisation Verification	no	The target organisation is perfectly known by the developers. The current are modelled by the Business Use Case Model - Current Situation
Business Architecture Document	no	The details present in the Business Use Case Model and in the Business Object Model are sufficient
Organisational Units	yes	Mapping the business process functionalities on the target organisation's structure

The criteria to fulfill this choice are related with:

- Characteristics of the project itself (e.g. criticality of the modelled business processes, type of target organisation).
- Characteristics of the organisation that develops software (e.g. team size, level of knowledge about internal rules).
- Temporal restrictions. Since resources are limited in engineering projects, it is always necessary a balance between the quantity and detail of the produced artifacts and the deadlines for implementing the project.

The produced artifacts result from a set of activities that occur inside those disciplines. In this case study, we identified the need for the artifacts to represent two distinct situations in terms of business: one part of the project represents reengineering activities of some business processes, while the other part represents the introduction of a new business process.

In several diagrams (e.g. Business Use Case Model), the standard UML is augmented with the stereotypes defined by RUP, thus allowing the creation of RUP-like artifacts. Other techniques not included in RUP, such as the classification of requirements FURPS+ [10] or the planning of maintenance teams [11] are also used.

Other Relevant Artifacts. Besides the artefacts suggested by RUP, we believe that it is also important to document other issues associated with the development of a software application:

- List of verification to assess the project status in the Quality Verifications that evaluate the transitions between the process phases (inception, elaboration, construction, transition).
- Planning and analysis of the required capacity to support and maintain the system when it is in the productive phase.

For this case study, we estimated the following values:

STEP 1. Estimate the requests volume (RV) with the table 3: Requests Volume $(r/d) = 1.46 \times 0.56 = 0.82$. We estimate that each request is served by one single person.

Table 3. Estimation of Requests Volume.

Organisational unit	Requests / day	Resolution Time (days)
Productivity by chain	1	1/8
Quality VQ	1/3	1/16
Presences in Post	1/22	1/8
Individual Performance	1/22	1/8
Premium Wage Calculation	1/22	1/8
TOTAL	1.46	0.56

STEP 2. Estimate the adequate level of preparation (LP): For the application under consideration, it was agreed to use the value of 80%.

STEP 3. Define the base number of persons: Using the suggestions in [11], with $RV = 0.82$ and $LP = 80\%$ we obtain a value less or equal to 2 persons.

STEP 4. Add extra capacity. For this case study, we consider that 2 persons is enough, since the organisation has already experience in dealing with similar systems.

STEP 5. Establish a strategy to diminish the number of requests in the queue. For the present case study, the following issues were considered relevant: (i) the requests from the organisational units 'Productivity per Line' and 'Calculate Prize Money' have higher priority; (ii) requests from other organisational units have a normal priority; (iii) high-priority requests are always served in first place.

Queues with separate priorities for each subsystem can also be created, but this alternative was not followed, since we intended to have a global estimation for the support and maintenance of the application responsible for the complete business process.

Results Analysis. We now analyse the positive and negative points associated with the parameterisation of RUP to implement the application under development. The main positive issues were:

- RUP, but especially the artefacts of the Business Modeling discipline, had a good performance in modelling the business processes, since they served as a basis for the artefacts at lower levels.
- The UML artefacts, with the stereotypes suggested by RUP, were a very efficient communication medium among all the stakeholders.
- The artefacts generated by RUP, namely the source code, have an easy utilisation for future systems that need them, allowing thus the reuse of code.
- The development process standardisation in the organisation that develops software was initiated to adopt the usage of RUP.
- RUP proved that it can adapt to the needs of organisations that develop software and specific projects.
- The set of RUP's disciplines allows to cope with the complexity and ensures that no important aspect related to the software development is forget, diminishing the risk of failure.

We also identified the following main negative aspects:

- The usage of RUP and UML must be accompanied with training for the developers.
- The need to create new roles to execute the software development process, such as the business process designer or the tools expert, forces internal reengineering.

Lessons Learned. The lessons learned with this second project were the following:

- It is very important that all stakeholders be aware of the tasks related to the use of a new software product. In this way all stakeholders, as suggested by RUP, should know and validate all the requirements.

- The transformation of use cases into class diagrams was accomplished via the following methodology [7]: (1) Transform system use cases into classes of 3 categories: Interface, Control, Data; (2) From the full set of classes generated, only the meaningful (depending of the description and use case realisation) are kept; (3) Aggregate the similar remaining classes (e.g. database-related classes); (4) Create associations between the groups of classes. From the remaining class groups, a mapping into a 3-tier architecture was inferred. The class groups, Interface, Control, and Data, were distributed into the three tiers, Presentation, Business, and Data, respectively. This method provided a mechanisation that lead to a time improvement in reaching the final architecture for the system.
- In matrix organisations, some problems, external to the clients and to the software development teams, arise just because of the existence of departments. When the departments are among the stakeholders for a project, they block the software development, because it will model business processes that can threaten the existence of tasks inside departments or even the departments.
- The RUP usage, even in the first projects, is not itself a delay factor to reach the final solution.

6 Conclusions and Future Work

The main conclusions for our work, based on the two case studies are:

- For an organisation represented by processes, RUP proved useful for modelling it and to mapping its business processes into software applications to support them.
- RUP can be used for systems already in execution, developed without a process-oriented perspective, allowing their integration into a new system, by reengineering the business processes, through the business use case diagrams for current and future situations.
- A good understanding and documentation of business processes is essential to build quality software systems and RUP recommends and supplies support for this discipline to be properly conducted.
- RUP configuration for a given project is essential for its success. This is a task mainly based on experience from past projects.
- The quality of personnel is the most important issue for the final quality of the software product.
- The Business Modelling discipline in RUP can be the basis for the business processes reengineering. The comparison of current and future situation on Business Modelling and the implementation of future situation into software systems are the basis for that. Also, as future situation tends to be implemented into software system, besides the business reengineering an improvement of processes performance can be achieved via the usage of software systems where manually processes existed.
- The structure and support during the project phase are key factors for the correct software development process and for the success in its utilisation.
- The main added value for a software development organisation is the correct business modelling of target organisations. The remaining tasks (such as programming the applications or testing) tend to be more easily automated and executed with supporting tools.

- The maintenance and support planning for a software system should be planned during its development process.
- The software development organisation must provide guarantees to its clients. Its own structure, the software development process used, and the quality of its collaborators are fundamental factors to achieve it.

As future work, we foresee the following. Instantiated specifications of software development organisation model (e.g. for CMM-2 or higher levels) can be provided. This way, the transition of existing organisations into reengineered ones could be performed more easily using to a current Business Modelling situation and a future situation described by the already available templates. The RUP parameterisation for a specific project, running in a specific software development organisation, and for a specific target organisation is based mostly on empirical decisions. An auxiliary process to parameterise it would provide benefits to all stakeholders as well permit its use by organisations that do not have expertise to configure it. Finally, the existence of a simulation environment to run current and future situation business processes, to test them, to evaluate them through business performance indicators, and to preview the impacts of reengineering would provide a valuable framework for business modelling decisions on future situations.

References

1. João M. Fernandes and Francisco J. Duarte. A Reference Model for Process-Oriented Software Development Organizations. In C. Gonzalez Perez, B. Henderson-Sellers, and D. Rawsthorne, editors, *OOPSLA 2003 Workshop on Process Engineering for Object-Oriented and Component-Based Development*, pages 31–42. COTAR, October 2003.
2. Michael Hammer. *Beyond Reengineering: How the Process-Centered Organization Is Changing Our Work and Our Lives*. Harper Collins, 1996.
3. C. Marshall. *Enterprise Modeling with UML: Designing Successful Software through Business Analysis*. Object Technology. Addison-Wesley, 2000.
4. Rational Software Corporation. Rational Unified Process: Product Overview, August 2003. <http://www.rational.com/products/rup>.
5. M. Paulk, B. Curtis, M. Chrissis, and C. Weber. Capability Maturity Model for Software, Version 1.1, 1996. Software Engineering Institute, Carnegie Mellon University.
6. Object Management Group. OMG Unified Modeling Language Specification version 1.4, 2001. version 2001.03.00.
7. João M. Fernandes and Ricardo J. Machado. From Use Cases to Objects: An Industrial Information Systems Case Study Analysis. In *7th International Conference on Object-Oriented Information Systems (OOIS '01)*, pages 319–28. Springer-Verlag, August 2001.
8. SAP. Training Manuals ABAP/4, 2000. SAP Portugal.
9. Howard Smith and Peteringar. *Business Process Management: the Third Wave*. Meghan-Kiffer Press, 2002.
10. R. Grady. *Practical Software Metrics for Project Management and Process Improvement*. Prentice Hall, 1992.
11. R. Ramaswamy. How to Staff Business-Critical Maintenance Projects. *IEEE Software*, 17(3):90–4, 2000.