AN INTEGRATED MODEL FOR MANAGERIAL AND PRODUCTIVE ACTIVITIES IN SOFTWARE DEVELOPMENT

Daniel Antonio Callegari, Maurício Covolan Rosito, Marcelo Blois Ribeiro and Ricardo Melo Bastos
Fac. Informática, PUC-RS, Av. Ipiranga 6681, Porto Alegre, Brazil

Keywords: Project Management, Software Development Process, Integrated Model, Project Planning, PMBOK, RUP.

Abstract: Software organizations are constantly looking for better solutions when designing and using well-defined software processes for the development of their products and services. However, many software development processes lack for more support on project management issues. This work proposes a model that integrates the concepts of PMBOK and RUP, helping process integration and assisting managers in the decision making process during project planning. We present the model and the results from a qualitative exploratory evaluation of a tool that implements the proposed model, conducted with project managers from nine companies.

1 INTRODUCTION

The increasing concern on software development quality drives organizations towards the adoption of software engineering models. Some of the most desirable characteristics of these models include the ability to capture software development best practices, flexibility to handle many types of projects, and good management support. The lack of specific methodologies for software project management and the increasing number and complexity of projects in organizations contribute to an increase in project management problems (Kerzner, 2000; Pressman, 2001).

The development of software products requires the planning and execution of activities defined in accordance to the scope of the project, where it is necessary to deal with both managerial and technical issues. However, most management models or guides are not software-specific. In addition, those management models are generally more related to industrial and manufacturing activities. The majority of software development processes, by their turn, generally provide just an adequate set of practices that supports the suggested activities and associated workflows.

Previous work in the area has presented interesting outcomes, but a tight integration of management and software processes with practical results is still an open question (Henderson-Sellers et al., 2000; Henderson-Sellers et al., 2001; Rehman & Hussain, 2007; Schwalbe, 2002).

This paper presents a model that integrates project management to software development processes named **Software Planning Integrated Model (SPIM)**. The model includes a set of rules for the integrated planning of managerial and productive activities in the context of software development. SPIM joins the concepts of the Project Management Body of Knowledge Guide (PMBOK) (PMI, 2000) and one of the most well-known software development processes, the Rational Unified Process (RUP) (Kruchten, 2000) to extend the proposal presented in (Callegari & Bastos, 2007). While the PMBOK provides a management perspective of the solution, the technical view is obtained from RUP. Hence, when applying project management knowledge together with the appropriate software process for a given organization, we can obtain a more complete and integrated flow of activities and their dependencies.

This paper is organized as follows: we first present an overview and the motivation for the research. A brief introduction of the base models is available in section 3. The SPIM model and its evaluation are discussed in sections 4 and 5, respectively. Finally, section 6 presents the conclusions and future work.

Study developed by the Research Group of the PDTI 01/2008, financed by Dell Computers of Brazil Ltd. with resources of Law 8.248/91.

Antonio Callegari D., Covolan Rosito M., Blois Ribeiro M. and Melo Bastos R. (2008).
AN INTEGRATED MODEL FOR MANAGERIAL AND PRODUCTIVE ACTIVITIES IN SOFTWARE DEVELOPMENT.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 25-32
DOI: 10.5220/0001683200250032

2 OVERVIEW AND MOTIVATION

As in (Kerzner, 2000), according to many empirical studies the effectiveness of an organization depends, in part, on the success of its projects.

Project management means applying knowledge, skills, tools and techniques to the project's activities, in order to meet or exceed the needs and expectations of the interested parties (stakeholders). Project management has the goal of finishing a project inside schedule and within the defined budget, according to a previously arranged set of specifications. These elements characterize the triple constraint of project management: scope, time and cost. A well succeeded project, thus, means fitting these three objectives and satisfying the sponsors. In addition, a project is a temporary endeavor with the goal of producing a unique product or service. Generally, a project is directed to a specific result and involves the coordinated execution of inter-related activities. More than that, projects are planned, executed and controlled by people, and they are constrained by limited resources (PMI, 2000).

Project management in a software development environment is defined as the management of people and other resources by a project manager in order to plan, analyze, design, build, test and maintain an information system (Schwalbe, 2002). In order to accomplish these goals, a project manager needs some kind of support, generally based on a project management methodology to handle many singular project variables, responsibilities and tasks.

From the point of view of software, a software development process (SDP) is a set of activities and related results that lead to the software production (Jacobson, Booch & Rumbaugh, 2001). The importance of having a standard SDP relies on the fact that it becomes the guide for the execution of all projects inside an organization. Therefore, many processes or guides such as RUP, Extreme Programming (XP) (Beck, 2004), Microsoft Solutions Framework (MSF) (Hundhausen, 2005) and OPEN Process Framework (Graham, Henderson-Sellers & Younessi, 1997) are being used as a common ground when designing standard software development processes in organizations.

Despite that, RUP, for instance, does not cover essential project management needs such as people management and subcontract management. In the other hand, OPEN presents a set of activities and techniques that address areas such as quality, cost estimates and management metrics. Nevertheless both models were found to lack enough support on essential knowledge areas of project management, namely procurement, communication and human resources.

Fundamentally, past and present work on the literature indicates the importance of using well defined software processes in organizations. Meanwhile, there seems to be not enough work in fulfilling the lack of project management issues in those processes. Software development processes generally provide just a set of practices that deal with certain activities and workflows related to management.

2.1 Related Work

In (Henderson-Sellers et al., 2000), RUP and OPEN are examined from a project management point of view and evaluated whether they meet acceptable standards in process support, project management guidelines, and full lifecycle description for object-oriented software development. The authors conclude that both processes are deficient in some of the standard project management areas of knowledge, like procurement management, communications management and personnel management. In order to support the full suite of project management techniques, further extensions to these areas of knowledge seem to be desirable.

A qualitative evaluation is performed on the public domain component of RUP and on OPEN in (Henderson-Sellers et al., 2001). The authors focus their comparison on aspects of the process architecture and underpinning metamodel, on the concepts and terminology used and on the support for project management. The authors conclude that the metalevel architecture of RUP leads to some dilemmas in terms of the lack of support for a truly iterative development. OPEN offers more extensive support in the area of cross-project suites of application development and maintenance and also more extensive support in metrics and quality considerations.

In (Rehman & Hussain, 2007), four important project management methodologies or frameworks – PRINCE2, RUP, Agile Development Methods and MSF – were compared to PMBOK. The authors conclude that all the methodologies or frameworks discussed have some common tools and procedures. They also propose a combined approach to achieve better results.

2.2 The Integrated Model Approach

Some types of managerial activities in software projects are inherent to the process and they do not appear at the moment we are planning the project. These activities (or their dependencies) are just the activities that most of the time cause an important slip in schedule and do not necessarily add another input in the project's risk list.

When planning a software project, it is likely that the manager does not have all relevant information up front, which forces him to interact with other departments in the organization. Hence, the flow of activities in an individual project is usually related to other common activity flows of the organization (here named **enterprise workflows**). Both flows are executed in parallel and may interfere in project schedule and cost. As a consequence, we need a solution that can provide a greater level of integration among the concepts and the models from these two areas. More than that, the desired solution should allow the development of tools to support the decision making process regarding technical and managerial planning.

By analyzing how project management knowledge can help improving current software development processes we can derive new tools for supporting different levels of automation in the planning and execution of activities inside a software project.

The proposed model (SPIM) defines three different types of activities. The first two belong to the project's workflow: Activities directly related to the construction of the product are called **productive activities**. Activities that are only necessary to coordinate the construction of the product are referred to as **managerial activities**. Finally, any other activities that do not belong to an individual project's activity workflow (and may be else shared by other projects) are called **management supporting activities** (the latter are part of the Enterprise Workflow component of the model, detailed in section 4).

Following this definition, we can find potential dependency relations between the activities in both workflows. As an example, the activity of deploying a system's database (which fits in the project's workflow) may depend on the acquisition of the server by the responsible department (this activity fits in the shared workflows of the company). As a consequence, the project manager needs continuous support in order to keep track of these kinds of dependencies (here, deploy the database is a productive activity, while acquisition is a management supporting activity). Since each project is unique, it is not feasible to build a universal software project "template"; instead, we can provide tools that help performing and constantly validating a plan, based on the three types of activities, on the resources and on related work products. Hence, the final goal of the SPIM model is to help managers to solve problems related to the inadequate definition and inter-relation of activities in a software project.

3 INTEGRATING PMBOK AND RUP MODELS

The Project Management Body of Knowledge Guide provides the best practices on project management that are applicable to the vast majority of the projects in many areas (PMI, 2000). Despite being a well accepted guide, the PMBOK is not a process in the strict sense, as it does not define actions nor it states how they must be followed and executed for the correct development of a project. The PMBOK Guide also does not include a metamodel. Nevertheless, a base model was presented previously in the form of a UML class diagram in (Callegari & Bastos, 2007) and covers concepts from general structures such as Organization, Program and Project, as well as the most important ones for our current problem, such as Activities, Stakeholders, Roles, Deliverables, and associated classes.

The Rational Unified Process (RUP) is referred to as an iterative software development process based on the SPEM meta-model (Kruchten, 2000; Jacobson, Booch & Rumbaugh, 2001). Its models cover concepts such as Artifacts, Roles, Disciplines, Activities, Phases, WorkflowDetails and ToolMentors.

According to (Henderson-Sellers et al., 2000), RUP does not attempt to cover all aspects of project management and it does not cover issues such as managing people, managing budget, and managing contracts. This is understandable because RUP evolves from the unification of methods for software development, and not from project management processes. Because we can express PMBOK's concepts in the same representation as the RUP models, it is possible to compare and semantically integrate both models, which originated the initial PMBOK+RUP model. Figure 1 presents part of the model. More information can be obtained in (Callegari & Bastos, 2007).

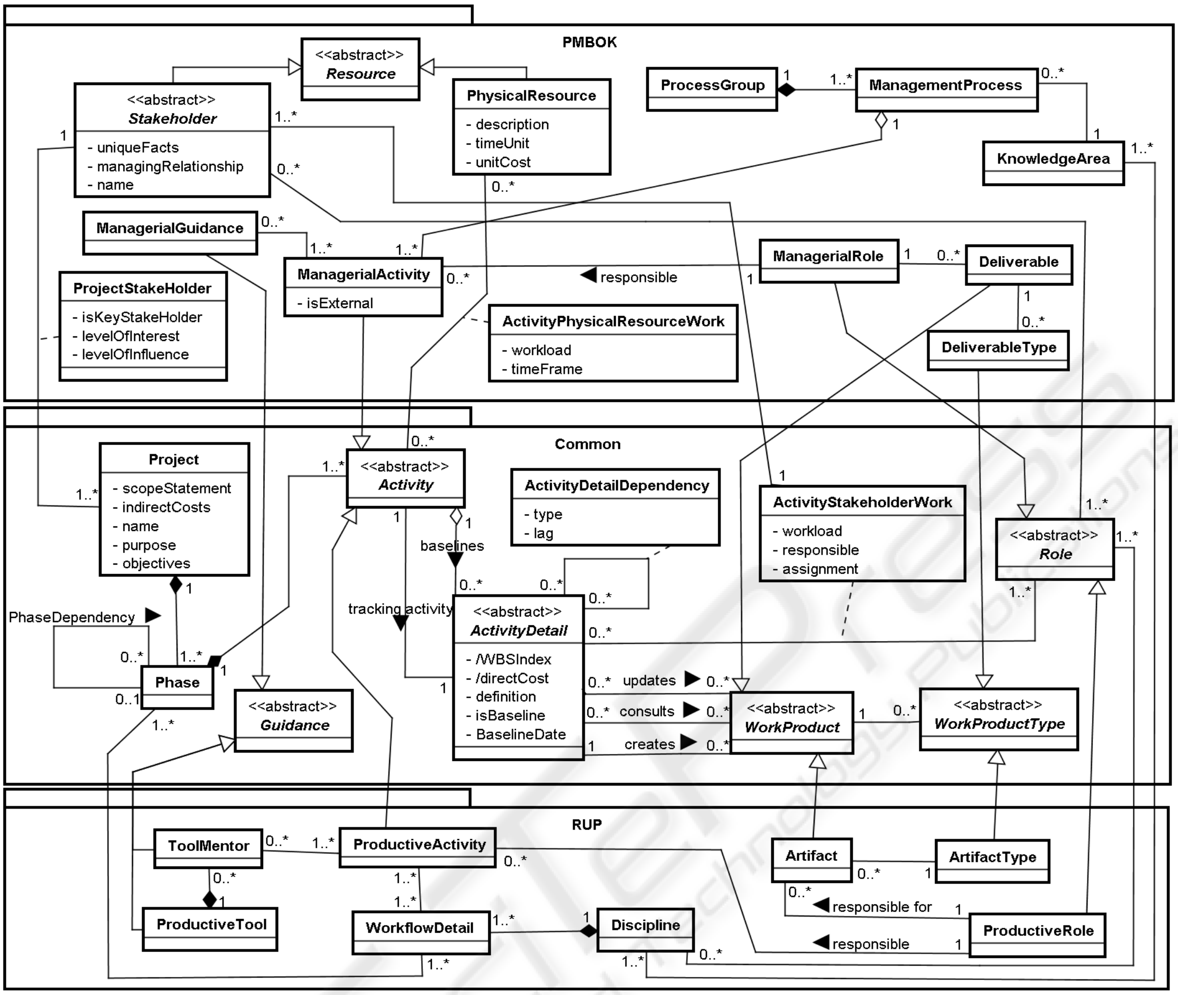


Figure 1: Part of the PMBOK+RUP model, from which SPIM is derived (more details not shown due to space reasons).

The detailed analysis of the PMBOK and RUP models allowed us to identify relationships among the elements of each model. When designing the integrated model, some of the main goals were: (a) to allow the integrated planning of the product and management of the project; (b) to distinguish the types of activities and work products; (c) to allow the integrated scheduling of managerial and productive activities; and finally, (d) to distinguish the possible relations between an activity and an artifact (create/update/consult).

It is important to note that when performing an integration of two models, the following conditions can occur: (a) an **overlapping of concepts** (two classes with the same concept on each model), in this case it is possible to transform them into a single concept inside a "common" package; (b) a **relationship of concepts** (a class of one of the original models relates to some other class on the other original model, but they do not represent the exact same concept), in which case we must create an association between them; and (c) classes with **independent and distinct concepts** from each original model, in which case we must leave each class in its own package.

As a consequence, the PMBOK+RUP integrated model is composed of three packages: one for the project management concepts, one for software development processes concepts and finally a common package that holds the concepts that cross both models.

The model covers nearly 50 classes. It maps the software **Lifecycle** concept as a set of **Phases**. The **Disciplines** split the process elements in subject areas. The **Roles** are played by the **Stakeholders** (a kind of **Resource**) in order to produce, consult or modify **Artifacts** of a project. The **Activities** are supported by tools or guides (generally called activity **Guidance**) and keep references to the artifacts they handle.

The model also has two specialized classes **ProductiveRole** (in the RUP package) and **ManagerialRole** (in the PMBOK package) indicating roles with productive and managerial responsibilities respectively. The base **Role** class is located in the Common package.

This intermediate model served as an important input for the SPIM model. Its analysis has originated implicit and explicit verification rules. The implicit rules can be directly obtained from the integrated model by means of the UML diagram semantics. The explicit rules were added to assure the consistency of the SPIM model and some are described in the next section.

4 SPIM - SOFTWARE PLANNING INTEGRATED MODEL

The PMBOK+RUP model was used as the base conceptual structure for the development of the Software Planning Integrated Model. It is also necessary to cover specific issues regarding the process of planning the activities for a software project. Figure 2 synthesizes the elements of the SPIM model.

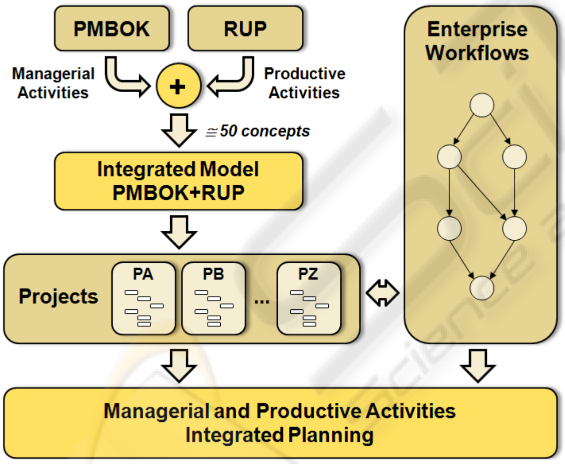
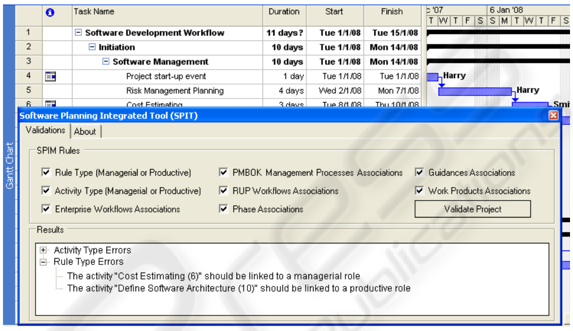


Figure 2: Projects are based on the PMBOK+RUP model, and SPIM integrates them to other workflows.

The idea behind the SPIM model comes from the need to solve the problems listed in sections 1 and 2. SPIM was conceived to reduce the complexity in visualizing the interdependencies of both enterprise workflows and the individual project's workflow of activities. Activities inside the enterprise workflow component are shared by some or all the projects the company is running, and they consume resources other than those already allocated to the projects.

One of the benefits to use such integrated approach is the ability to anticipate needs coming from different sources (e.g. projects and a support department itself, for instance) and reschedule the affected activities automatically.

From all rules of SPIM, eight of them were evaluated in this work. The rules were implemented in a tool called SPIT (**Software Planning Integrated Tool**) as an Add-in for a commercial project planning software.



		Task Name	Duration	Start	Finish
1		⊟ Software Development Workflow	11 days?	Tue 1/1/08	Tue 15/1/08
2		⊟ Initiation	10 days	Tue 1/1/08	Mon 14/1/08
3		⊟ Software Management	10 days	Tue 1/1/08	Mon 14/1/08
4	▦	Project start-up event	1 day	Tue 1/1/08	Tue 1/1/08
5		Risk Management Planning	4 days	Wed 2/1/08	Mon 7/1/08
6	▦	Cost Estimating	3 days	Tue 8/1/08	Thu 10/1/08

Software Planning Integrated Tool (SPIT)

Validations | About

SPIM Rules

☑ Rule Type (Managerial or Productive)
☑ PMBOK Management Processes Associations
☑ Guidances Associations
☑ Activity Type (Managerial or Productive)
☑ RUP Workflows Associations
☑ Work Products Associations
☑ Enterprise Workflows Associations
☑ Phase Associations

Validate Project

Results

⊞ Activity Type Errors
⊟ Rule Type Errors
The activity "Cost Estimating (6)" should be linked to a managerial role
The activity "Define Software Architecture (10)" should be linked to a productive role

Figure 3: The SPIT tool working as an Add-in for a commercial project planning software.

All information needed to perform the validations in SPIM is stored in custom fields inside the commercial product. Figure 3 presents a snapshot of SPIT in action. As mentioned before, the management supporting activities are part of the so called enterprise workflows. Each enterprise workflow is a set of activities that can be consumed by (and run in parallel to) one or more projects.

SPIM allows each instance of an enterprise workflow to be registered as a management supporting activity in software development projects. Thus, the proposed model aims to assist the identification of dependencies between the activities in both workflows (Figure 4). A workflow engine must constantly update and inform projects about the duration of each instance of the enterprise workflows. The example in Figure 4 is represented in PERT networks notation (Burke, 2001): arrows represent activities and circles represent events between activities.

In this example, the activity D in a software project depends on both the activity C (which can be either productive or managerial) and one of the enterprise workflows. The curved arrow is a management supporting activity which indicates that all the activities on the specific enterprise workflow (W1, W2, W3 and so on) must be finished before we can move on to activity D (assuming activity C has also finished). Therefore, a management supporting activity is actually a kind of virtual activity that represents an entire execution of one of the enterprise workflows (isExternal attribute of the ManagerialActivity class in Figure 1).

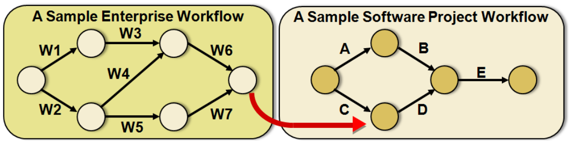

A Sample Enterprise Workflow
W1
W3
W6
W4
W2
W5
W7
A Sample Software Project Workflow
A
B
E
C
D

Figure 4: Interdependency between an enterprise workflow and a software project workflow.

The company can define a set of reusable enterprise workflows, such as “hardware acquisition”, “hiring new people”, “work environment setup”, and so on. Each reference to an enterprise workflow represents new instances of the corresponding “W” activities. Other departments update the activities’ information, and the affected projects are then rescheduled. All the rules as well as the concepts behind SPIM were evaluated in a series of interviews with experienced software project managers from 9 distinct companies. The next section describes this evaluation in detail.

5 MODEL EVALUATION

In order to evaluate the concepts behind the SPIM integrated model with respect to its acceptance and applicability, we have conducted a qualitative exploratory research involving 12 experienced project managers who work for a total of nine software development companies (detailed information is omitted for confidentiality).

All participants received a brief training on the SPIM model and were given the chance to ask prior questions. Then they were presented the same project description and were asked to plan the corresponding project by using the SPIT Add-In. After that, they answered a survey. The results are being used to refine the model.

Data acquisition was taken during November 2007 by means of a questionnaire produced in accordance to Rea and Parker (2005).

The survey had 33 questions where the first 8 captured the organizational profile, the following 10 were focused on the managers individual knowledge mapping and the remaining were used to estimate the SPIM model’s contributions in the planning process from the project managers’ point of view. The subjects in the survey had an average of 12.25 years of professional experience (min=7; max=20). The involved companies were distributed in different information technology segments (three software factories, two mobile technology development companies, two web development companies, one government institution and one research and development center). Together these companies sum up to 1620 IT-related professionals, in which 90 are project managers and 823 are software developers. In addition, these companies have an average IT market experience of more than 13 years. As we can see in Table 1, an initial analysis reveals a wide disparity in size and organizational structure, resulting in a standard deviation value of 155.34, which characterizes an adequate sample for our analysis.

Table 1: Criteria regarding the profile of the 9 companies.

Criteria	Avg.	Std. Dev.
IT market time (years)	13,33	8,25
No. of IT-related professionals	140,22	155,34
No. of project managers	10,00	8,06
No. of developers	91,44	125,85
--	--	--
Software process is RUP-based	67%	-
Management is PMBOK-based	100%	-

Table 2: Perceived benefits in performing the integrated planning of managerial and productive activities.

Question	%
Reduction in time during the project’s elaboration process.	58%
Identification of the dependencies between the management supporting activities and the production activities.	100%
Identification and measuring of the indirect costs of the project, due to the management support activities.	67%
Access to enterprise workflow information.	40%
The capacity of avoiding distortions during planning when support activities are involved.	100%

5.1 Survey Results

An initial analysis of the questions regarding the profile of the companies reveals that almost 67% of them adopt some kind of RUP-based software development process. Also, all of them adopt (fully or at some level) the concepts found on the PMBOK Guide in their projects. This confirms another aspect of the questionnaire, which reveals that 100% of the subjects received formal project management training and, thus, indicates all subjects are qualified to use the SPIM model.

In addition to the professional experience of the subjects, the average experience on project management was 5.04 years, ranging from 1 to 12

Table 3: A set of validation rules from the SPIM model and their evaluation by the managers.

	Rule	Average	Std. Dev.
1	Any given activity cannot create, modify or consult one same artifact at a given time. These operations must be made by distinct activities.	3.75	0.75
2	Managerial and management supporting activities cannot produce or modify productive, but only managerial work products. They can still consult productive work products, though.	4.42	0.79
3	Productive activities cannot produce or modify managerial work products, but only productive work products. They can still consult managerial work products, though.	4.42	0.79
4	Any given activity is only allowed to consult or modify a work product that has already been created by a preceding activity.	4.08	0.90
5	The role of the stakeholder associated to an activity must be compatible to the type of the activity (productive or managerial).	4.25	0.87
6	The value of informing the related guidance, whether productive or managerial, for each activity.	3.83	1.03
7	Managerial and management supporting activities must have at least one management-related stakeholder fulfilling their roles.	4.50	0.67
8	A productive activity must have at least one productive-related stakeholder fulfilling its roles.	4.42	0.67

(3.43 standard deviation; half of the sample being superior to 7 years). This indicates a wide range of experience regarding project management of the subjects. Together with the information of formal project management training, this data seems to explain the fact that 58% of the sample declared their experience in project management as “advanced” while the remaining 42% declared it as being “moderate”. Besides, 83% of the subjects measured their knowledge on RUP as “moderate” or “advanced”.

We begin the analysis of the SPIM model with the respondents’ evaluation of the direct benefits in performing the integrated planning of managerial and productive activities in a software project. The results are shown in Table 2.

According to the second and the last rows in Table 2, all managers found that the integrated planning allows the identification of the hidden dependencies between the management supporting activities and the production activities, while avoiding frequent distortions in the planning of the projects due to the uninformed use of resources from the management supporting activities.

Moreover, some interviewees mentioned the possibility of keeping compliancy with the development and organizational processes, as well as the advantage of prior identification of the timing some activities demand for avoiding schedule slip of dependent productive activities.

Respondents evaluated each of the 8 selected rules according to the following ordinal scale: 1-none, 2-low, 3-moderate, 4-high and 5-very high (Table 3). The general average value for the selected rules was 4.20, which suggests a great level of acceptance of the rules proposed in the SPIM model. In addition, this number increases to 4.47 if we consider only managers who have more than 7 years in project management experience.

In spite of the high scores, rules #1 and #6 were considered as providing the lowest benefits among all the rules (respectively, 3.75 and 3.83, which can be interpreted as “near to high”). In the other hand, rules #2, #3, #7 and #8 had the highest average score for all the rules, near to 4.5 points, so as being recognized as the most favorable ones. The rules that have shown the lowest disparity in the answers where #7 and #8, resulting in a standard deviation value of 0.67. The rule associated with the highest divergence in opinions was #6, which considers the activity’s guidance. These numbers reveal the relative importance of all the issues that must be considered when performing the verification of the project’s plan in accordance to the model and should indicate which items demand more specific concern.

The visibility of management supporting activities together with the activities in the software project (whether productive of managerial) was also identified as a strong benefit of SPIM by 58.33% of the interviewees. When questioned whether or not they agreed on the distinct nature of the three types of activities, all of the respondents answered they considered the distinction of the three activities a very important aspect. Besides, all respondents also agreed that the obscurity in identifying management supporting activities during project planning can negatively affect the project. It is important to note that all questions were answered without the intervention of the interviewer.

In reply to an open question, some managers mentioned the ease of validating some of the rules directly from the project planning tool as an important factor in SPIM. In reply to another open question, nearly 60% of the respondents mentioned the fact that many times the project manager only perceives the need to have asked another department some information earlier in time just at the very moment the team must execute a project's activity that depends on that other department (e.g. acquiring new equipment or hiring a new programmer are common examples). These observations confirm the relevance in building this integrated model.

As a final consideration, all the interviewed project managers found that the SPIM model contributes in identifying the dependencies of the activities between the project flow and the support management flow, which allows the prediction of the needs that come upon the organizational support areas during the planning of the project, resulting in a more accurate plan and schedule.

The results collected in the survey reaffirm the benefits the SPIM model provides in solving the problems related to the inadequate definition of tasks (increase in cost and delays in projects, for instance) due to the obscurity in visualizing the interdependency between the organization's and project specific workflows.

6 CONCLUDING REMARKS AND FUTURE WORK

This paper presented a proposal to integrate the main concepts of the PMBOK to a model of software development process (in this case RUP). First, we have identified the importance of project management activities during a software development project. Then we noticed the lack of information on management practices in most software development processes used nowadays. After an individual analysis of each base model, we proposed a model that covers both perspectives into a single integrated model. Later, we have analyzed the results from a series of interviews with experienced project managers, based on a real tool.

This work contributes with some interesting findings that reaffirm the goal of designing a support tool that help software project managers in planning software development projects. We believe that is possible to extend this integration model to other software development processes because, in accordance to (Sommerville, 1995), different models of software development processes share fundamental concepts. A previous study of integration involving the OPEN process framework (in place of RUP) also reiterates the applicability of the model. The next steps include the generalization of this approach to other software process models, as well as the development of a multi-criteria resource selection mechanism for software projects.

REFERENCES

Beck, K., 2004. *Extreme Programming Explained: Embrace Change.* Addison-Wesley, 2nd edition.

Burke, R., 2001. *Project Management: Planning and Control Techniques.* John Wiley & Sons, 3rd edition.

Callegari, D., Bastos, R., 2007. Project Management and Software Development Processes: Integrating RUP and PMBOK. In *ICSEM - International Conference on Systems Engineering and Modeling, IEEE, pp. 1-8.*

Graham, I., Henderson-Sellers, B., Younessi, H., 1997. *The OPEN Process Specification*. Addison-Wesley.

Henderson-Sellers, B., Dué, R., Graham, I., Collins, G., 2000. Third generation OO processes: a critique of RUP and OPEN from a project management perspective. In *ASPEC - Seventh Asia-Pacific Software Engineering Conference*, pp. 428-435.

Henderson-Sellers, B., Collins, G., Dué, R., Graham, I.M., 2001. A Qualitative Comparison of Two Processes for Object-Oriented Software Development. In *Information and Software Technology*, vol. 43, no. 12.

Hundhausen, R., 2005. *Working with Microsoft Visual Studio 2005 Team System.* Microsoft Press.

Jacobson, I., Booch G., Rumbaugh, J., 2001. *The Unified Software Development Process.* Addison Wesley.

Kerzner, H., 2000. *Applied project management: best practices on implementation.* USA: Wiley & Sons.

Kruchten, P., 2000. *The Rational Unified Process: An Introduction.* Addison-Wesley, 2nd edition.

Pressman, R., 2001. *Software Engineering: a practitioner's approach.* McGraw-Hill, 5th edition.

PMI - Project Management Institute., 2000. *PMBOK Guide: A Guide to the Project Management Body of Knowledge*, 3rd edition.

Rea, L. M., Parker, R., 2005. D*esigning and Conducting Survey Research: A Comprehensive Guide.* Jossey-Bass, 3rd edition.

Rehman, A., Hussain, R., 2007. Software Project Management Methodologies/Frameworks Dynamics "A Comparative Approach". In *ICIET – Internatinal Conference on Information and Emerging Technologies*, Karachi, Pakistan, pp. 1-5.

Schwalbe, K., 2002. *Information Technology Project Management*, Thomson Learning. Canada, 2nd edition.

Sommerville, I., 1995. *Software engineering*, Addison-Wesley. 5th edition.