International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 6 (2014) pp. 160 - 170

SPIM: An Integrated Model of Software Project Management and Organizational Workflows

Maurício Covolan Rosito[1] and Ricardo Melo Bastos[1]

[1] Department of Computer Science, Pontifical Catholic University of Rio Grande do Sul,
Ipiranga Avenue, 6681, Porto Alegre, Brazil
mauricio.rosito@acad.pucrs.br, bastos@pucrs.br

***Abstract*: Software projects are very dynamic and are often full of uncertainties and problems that may cause schedule delays, cost increases or unsatisfactory product delivery. To perform these adjustments on the project plans, managers must deal with a set of aspects: the technological resources that will be affected, human resources allocation and other design elements. Yet, during the planning and execution of a software project, the integration of specific activities in the project with the activities that take part in the organization's common activity flow should be considered. In this sense, this article presents the results of an experimental study related to the integration of project management with organizational flows. A software tool was built to demonstrate and evaluate the results.**

Keywords: Empirical Evaluation, Organizational Workflows, Software Project Management, PMBOK, OPEN, RUP**.**

I. Introduction

The development of a software product involves handling resources and activities, among other things, to produce the desired results [1]. Usually, a project is intended to achieve a specific result and involves the coordinated implementation of interrelated activities. More than that, projects are planned, executed and controlled by people and are constrained by limited resources [2].

During the planning and execution of software projects, different types of tasks are assigned to resources in order to meet the goals related to time and costs of these projects. In response to new information or estimates, the project manager may have to modify the project plan, such as reallocating resources or canceling tasks [3]. These adjustments which arise during the execution of software projects give rise to the term reconfiguration of projects. In this research, any effort made on a pre-existing planning of activities or on the resources allocated to them is understood by dynamic reconfiguration.

Research in this area has traditionally addressed this issue by focusing on aspects related to the initial planning of projects (static perspective). More recently, this problem has been treated in a dynamic perspective, in which there are adjustments to the project during its implementation. Such changes may impact the time and costs previously defined for the project. Yet, project managers must be aware of the software projects inherent risks. In general, companies are organized in order to manage multiple projects simultaneously [4]. Additionally, it seems that, unlike the traditional model of project management (which describes the projects individually), models of multi-project management must deal with interdependencies between the various projects with a set of constraints on time and available resources [5].

Also, during the execution of the project, the manager may need to interact with other departments of the organization in order to obtain relevant information to the project (contact the human resources department about the need for staff training, for instance). Thus, the distinction between the specific activities in a project with activities that take part in the organization's common activity flow can be observed. This way, a project manager must deal with this decoupling amongst the activity flow of a software project and other activity flows of the organization which provide some kind of support to the project. Both types of workflows are executed in parallel, have their own resources and may influence the timing of activities and project costs. Therefore, the project manager needs some kind of support to help in the process of decision making taking into account the integration of these different streams of activities during the simultaneous execution of software projects.

Considering the presented situation in order to contribute to the solution of the noted difficulties, a computational model, called Software Planning Integrated Model (SPIM) was proposed. The SPIM allows the dynamic reconfiguration of software projects' support considering the planning and replanning of their activities. To evaluate the model and embodiment of the proposal, a software tool, called Software Planning Integrated Tool (SPIT), was developed. This tool was used in an experimental study, allowing the evaluation of the proposed model.

This paper is organized as follows: Section II starts with an overview of software project integration with organizational workflows to establish a background in this field. Section III presents the SPIT tool, considering all the functionalities implemented in this research. Section IV presents related works to this research. The planning and execution of the proposed experiment are described in the Section V. An analysis of results is presented in Sections VI and VII. In section VIII the findings and future works are described.

II. SOFTWARE PLANNING INTEGRATED MODEL

The SPIM model was developed considering the integration of project management concepts provided in Project Management Book of Knowledge Guide (PMBOK) [2] with the concepts of software development provided in Rational Unified Process (RUP) [8] and in Object-oriented Process, Environment and Notation (OPEN) [9]. The detailed study of the PMBOK, RUP and OPEN metamodels helped to identify how their classes are organized and which are the valid relations between the elements of each model.

In a previous work [21] we developed an integrated metamodel between PMBOK and RUP classes (called PMBOK+RUP). This metamodel is composed of three packages: one for the project management concepts, one for the concepts of software processes, and finally, a common package holding the concepts that occur in both models. This study allowed the development of a methodology to integrate models of project management with models for software development processes. After that, an integrated metamodel between PMBOK and OPEN (called PMBOK+OPEN) was developed (see [19]) with a structure similar to the PMBOK+RUP model (replacing the package for the software development process). The two software development processes, however, have particular characteristics that are reflected in different classes and different relationships with the PMBOK and Common packages. The PMBOK+RUP and PMBOK+OPEN metamodels provided the conceptual framework necessary to develop a unique model to assist in project planning considering the concepts arising from the software development processes. To demonstrate the feasibility of proposed concepts, we developed an integrated model called SPIM. Figure 1 presents the stages of development of the integration model SPIM.

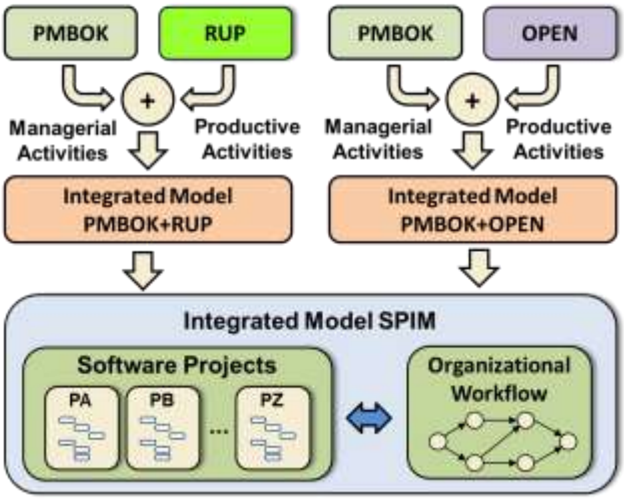

Figure 1. Stages of development of the model SPIM

The SPIM was designed considering the need of project managers to access information from other departments of the organization during the software project planning. To support this functionality, this model defines three different types of activities: (a) productive activities: activities directly related to the construction of the software product; (b) managerial activities: activities that are only required to coordinate the construction of the software product; and (c) management supporting activities: any other activities that do not belong to an individual project's activity workflow (and may be else shared by other projects).

The proposed distinction between the types of activities that occur on a software project allows project managers to identify potential dependency relationships between the activities of the organization's workflows and activities of a specific software project workflow. As an example, the activity of developing a particular web site (which fits in the project's workflow) may depend on the acquisition of the web server by the responsible department (this activity fits in the organization's workflows). The difficulty in visualizing this interdependence of workflows during the planning of activities can adversely affect the project, resulting, for example, increased costs and delays in project timelines. As a consequence, the project manager needs continuous support in order to keep track of these kinds of dependencies.

Following this definition it is possible to distinguish which activities should be updated by other sectors of the organization (using a mechanism such as a workflow) and which should be updated directly by the project manager. Each organizational workflow is a set of activities that can be consumed by one or more software projects. The management supporting activities are part of organizational workflows. SPIM allows each instance of an organizational workflow to be registered as a management supporting activity in software development projects. Thus, the SPIM model enables the integration of software management tools, such as Microsoft Project [10], with workflow systems (assuming that a workflow engine must constantly update and inform projects about the duration of each instance of the organizational workflows).

In the SPIM model (see Fig. 2), the Organization class represents a company that is organized by programs. Programs are groups of projects designated to reach a strategic objective. The organizations, usually, divide projects in several phases aiming a better managerial control. A necessary resource for the project, such as people, equipment or place, is represented by the Resource class. These resources are divided into active resources (Stakeholder class) and non-active (PhysicalResource class). The PhysicalResource class represents a physical resource in a project, such as a necessary material to accomplish an activity, a necessary equipment to accomplish an activity or a physical place.

The ProjectStakeholder class represents the relationship of the stakeholder with the project, for example: if it is a key stakeholder of the project, his level of interest in the project and his level of influence in the project. The ActivityPhysicalResourceWork class associates zero or more physical resources to zero or more activities. It establishes the physical resources work load in that activity.

The proposed model defines three different types of activities. Activities directly related to the construction of the product, such as coding or database modelling, are called productive activities (ProductiveActivity class). Managerial activities (ManagerialActivity class), however, may belong to the software development workflow (attribute isExternal = false) or belong to the business organization workflow (attribute isExternal = true). Each activity can belong to one or

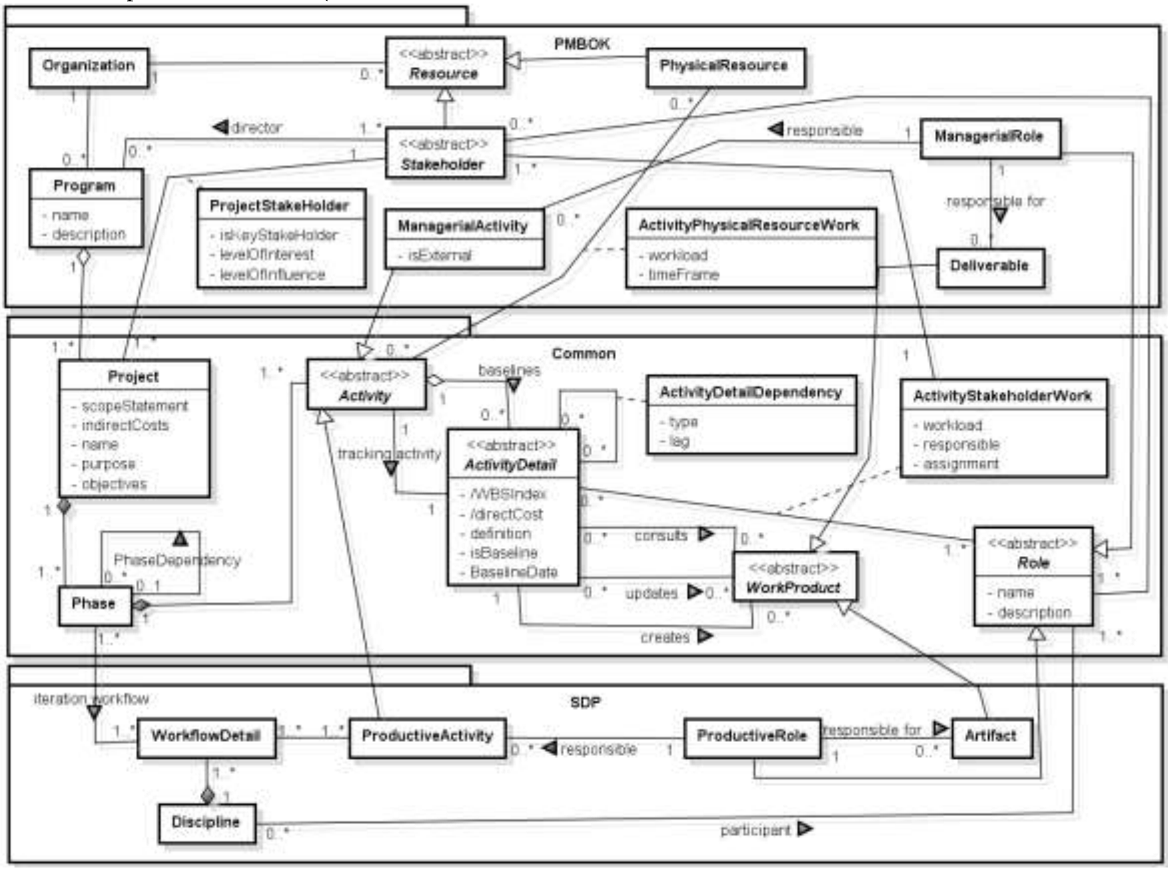

Figure 2. Main classes of the SPIM model

more baselines. In each baseline generation, an activity should maintain the relationships with the roles and work products (WorkProduct class). Thus, the ActivityDetail class was defined as responsible for maintaining these relationships, while the Activity class was defined as an aggregation of one or more ActivityDetail classes. Also, the ActivityDetailDependency class defines if one or more activities can be executed in parallel, and if two activities can be overlapped.

Stakeholders can play several roles during the execution of project activities. For each association between a role and activity (ActivityStakeholderWork class) there must be an association of this activity with a stakeholder who's able to play that role. Then, managerial activities are performed by managerial roles and productive activities are performed by productive roles.

In the software development process (SDP) package, we define a discipline as the division of process elements into areas of interest. Each discipline is formed by one or more work flows. Workflows define how productive roles must work together through their activities The Artifact class represents something that is produced, consumed or modified (such as documents, models or source codes) during the execution of activities in a software process.

During the SPIM model definition, a set of 19 rules (see [7]) were established to ensure consistency of the model. The rules were implemented in a tool called Software Planning Integrated Tool (SPIT).

III. SOFTWARE PLANNING INTEGRATED TOOL

The SPIT aims to demonstrate the concepts proposed by SPIM model and its set of rules. It was developed in 'C #. Net' in a modular way, allowing it to absorb new features quickly and without compromising its structure. Currently, the SPIT has integrated in its environment the following modules:

- SPIM Validator: acts as an add-in for Microsoft Project and performs the SPIM model validation rules on software projects;
- BackOffice: responsible for managing the information required by the SPIM model, such as roles definition, types of activities and associated work products. This information is exported to Microsoft Project through custom field to be used by the SPIM Validator module;
- Workflow Integrator: module responsible for synchronizing the information contained in organizational workflows with those present in a specific software project;
- Compiler: a tool for modelling, compilation, generation and verification of Petri nets. It internally uses PNML format to represent a Petri net into the system.

Other studies have been integrated into the platform, such as the SPIM Simulator, which will consist of an environment for simulation and analysis of Petri nets. Each module of SPIT is presented in the following subsections.

A. SPIM Validator

The SPIM Validator acts as an add-in for Microsoft Project 2010 and performs the SPIM model validation rules on software projects. This choice allows SPIT to take advantage of the features that are already implemented in accordance with the proposed integration model.

The SPIM model, however, proposes certain concepts and constraints that are not implemented by this commercial tool. According to Chatfield and Johnson (2010), this commercial software has extra fields which enable the storage of custom information for tasks and resources. All extra information needed to perform the validations in SPIM is stored in custom fields inside the commercial product. Thus, it was possible to add to this commercial tool, for example, information about the role played by a particular stakeholder during the execution of a particular task.

The additional information proposed by SPIM that are not supported by Microsoft Project, such as assigning a role for a stakeholder, is stored in an external database. Table 1 shows some of the constraints defined for the SPIM model. The SPIM Validator module has been updated to access a database in Microsoft SQL Server 2008 [24] through stored procedures, due to safety and ease of maintenance.

#	Restrictions of the Integrated model SPIM
1	A program must have a director. Therefore, a stakeholder who is director of a program must have a managerial role;
2	A project must have only one key stakeholder;
3	An project phase cannot have itself as predecessor or predecessor
4	A managerial activity must be associated with at least one managerial role;
5	A productive activity must be associated with at least one productive role;
6	The stakeholder responsible for a managerial activity must play a managerial role
7	O stakeholder responsible for a productive activity must play a productive role;
8	Each project activity must be the responsibility of only oneindividual, even though many people come to work in that activity;
9	A managerial activity cannot produce or modify a productive work product, only a managerial work product. However, this activity may consult a productive work product;
10	A productive activity cannot produce or modify a managerial work product, only a productive work product. However, this activity may consult a managerial work product;

Table 1. Some Restrictions of the Integrated Model SPIM.

B. BackOffice

The BackOffice module is responsible for managing the information required by the SPIM model, such as roles definition, types of activities and associated work products. Thus, BackOffice module provides an interface (see Fig. 3) that allows project managers to associate a software project registered in the SPIT database with the file used by the commercial tool (.mpp file). This information is exported to Microsoft Project through custom fields to be used by the SPIM Validator module. This module offers two ways to export the information in its database for Microsoft Project:

a) through project templates: a set of tasks and their associations should be exported to commercial software at once;
b) through individual custom fields: individually select the custom fields that should be exported to a specific project.

Thus, the BackOffice module allows exporting information about roles, work products, processes, management of PMBOK, guidances and workflows of RUP and OPEN.

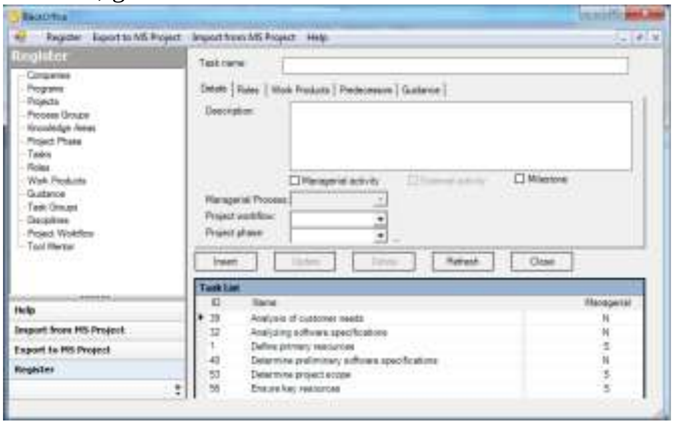

Figure 3. BackOffice screen

C. Workflow Integrator

This module is responsible for synchronizing the information contained in organizational workflows with those present in a specific software project. Currently, different technologies and solutions can be found for the development of workflows. Once the SPIT tool was developed using '. NET Framework', we chose to develop the organizational workflows through the Microsoft Visual Studio 2010 Workflow Designer tool [18]. The Windows Workflow Foundation (WF) provides the Visual Studio 2010 Workflow Designer that enables users to create their own custom workflows and workflow activities. Custom code, as well as design forms – such as web forms – may be included in the referred workflows to be used by each one of them to communicate itself with the workflow users during association and run time.

Due to the distributed nature of a business process (coordinating information coming from multiple sources across an organization) a workflow may be deployed as a distributed application. To support this functionality, the use of web services was chosen as the host platform to access distributed workflow applications. Web service definitions were designed to abstract how applications communicate with each other. These services allow the decoupling of business logic from client application code. Then, we use web services to expose interaction points modeled inside the workflow as web service methods. Interaction points are implemented in .Net Framework using web service activities. These activities are directly mapped to methods defined on the SPIT interface.

D. Compiler

The Compiler module was developed to perform the translation (in both directions) between work break down structure (WBS) and Petri nets. This module uses the Petri Net Markup Language (PNML) [25] as a XML-based interchange format for Petri nets. The Compiler module is divided into three components:

a) Parser: which contains the classes that compose the parser for .pnml and .mpp files;

b) Translator: which contains the classes that compose the C#.net representation of WBS and Petri net in SPIT; and

c) Writer: which contains the classes responsible for writing the output files.

However, this last module is still under development and, therefore, was not used in this experiment.

		Task	Start	Finish	Predecessor	Resources
7		⊟ **Software Requirements**	**05/01/2012**	**20/01/2012**		
8		Analysis of customer needs	05/01/2012	12/01/2012	6	System analyst
9		Determine preliminary software specifications	12/01/2012	17/01/2012	8	System analyst
10		Develop preliminary budget	17/01/2012	19/01/2012	9	Project Manager
11		Ensure resources required	19/01/2012	20/01/2012	10	Project Manager
12		Analysis completed	20/01/2012	20/01/2012	11	
13		⊟ **Design**	**20/01/2012**	**06/02/2012**		
14		Analyzing software specifications	20/01/2012	24/01/2012	12	System analyst
15		Develop functional specifications	24/01/2012	31/01/2012	14	System analyst
16		Develop prototype based on functional specifications	31/01/2012	06/02/2012	15	System analyst
17		Design completed	06/02/2012	06/02/2012	16	
18		⊟ **Development**	**06/02/2012**	**05/03/2012**		
19		Analyze functional specifications	06/02/2012	07/02/2012	17	Developer
20		Develop source code	07/02/2012	28/02/2012	19	Developer
21		Perform developer tests	13/02/2012	05/03/2012	20TI-75%	Developer
22		Development completed	05/03/2012	05/03/2012	21	

Figure 4. BackOffice screen

E. Example scenario

To better illustrate the use of the new features proposed in SPIT, we will consider the scenario of developing a web site. For this purpose, we will examine a common situation in software projects. In this case, the scenario under consideration is related to the acquisition of new hardware for a specific project.

At first, the manager should perform the initial configuration of the project through the BackOffice module of SPIT. Through this interface it can register all the information specified in the model SPIM, such as the definition of roles, work products and guidance. Once registered in the SPIT's database, this kind of information must be exported to a file that can be used by Microsoft Project. The BackOffice module allows the manager to select which information will be registered as custom fields in this commercial tool. Through the custom field information, the project manager may start the registration of the project plan items, such as: tasks, resources and other settings.

In the scope of this research, we will use a small project to explain the interaction of the SPIT with a workflow engine. Figure 4 shows a partial schedule developed for a fictitious software project. For this example we will assume that we are on January 19 and the project manager needs to buy a new server to host the application source code. The acquisition of new hardware is an activity performed by the company's administrative sector. This activity belongs to an organizational workflow that is executed in parallel to the project's workflow and has its own resources and deadlines. The disregard to the dependencies between the activities of these different workflows can result in software project planning distortions (such as schedule delay).

Based on this situation, the manager should identify the deadline for requesting the new server to the administrative sector. First, to accomplish this task with the support of SPIT, the project manager must identify which tasks depend on the acquisition of this server to be executed. In this example, the manager must select the properties of the activity called “Develop source code”. In the custom fields tab he must type the ID of the workflow for the acquisition of the server.

Since this configuration was performed on the project, the manager can trigger the SPIM Validator module to access the information in this organizational workflow. In this case, the workflow returns the average execution time for the acquisition of this hardware (see Fig. 5). With this information, the manager has better information to decide which actions to take avoiding project delays.

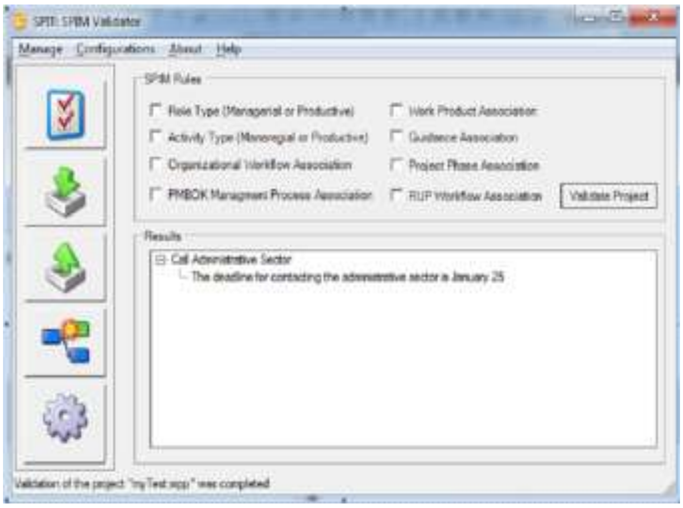

Figure 5. SPIM Validator screen

IV. RELATED WORKS

Currently, several studies have been developed by the software engineering community involving the description of approaches, methods, algorithms, or strategies related to the dynamic reconfiguration of software projects. In a recent study [see 26], a systematic review of aspects related to software projects' dynamic reconfiguration emphasizing the integration of project management with the organizational flows was conducted by these authors. Systematic review is a way to evaluate and interpret the relevant work to a specific research question [27], [28]. Also, the systematic review gives a scientific rigor to a literature review process and, as a consequence, minimizes the slants that can happen during a conventional literature review.

Approximately 130 articles were analyzed from the year 2000 to the present. After that, all articles were preliminary selected for a future analysis based on the following sequence of steps: (1) the search string was run in each of the search engines; (2) the title and abstract of each article were read; (3) when initially approved, the full texts were read for a final approval; (4) if in doubt due to lack of clarity in the abstract, a quick reading of the text was conducted; and (5) the remaining articles were selected for full reading. The implementation of this sequence of steps resulted in the selection of 24 works. Fourteen of them were later dropped due to the lack of minimal relevance expected. Thus, the final selection included 10 articles (this reduction is expected in a systematic review, mainly due to selection errors from the search engines).

By analyzing the results, it was possible to draw some conclusions based on a quick quantitative analysis. First, all selected works present solutions to the scheduling of activities on the project schedule. Also, all results show solutions that involve some kind of dynamic solution. However, only two articles present solutions that provide integration of the project's activity flow with the organizational flows of the company. Also, more than half of the selected works have some kind of tool or prototype. However, only one study includes results from a case study or experiment.

In [3], the authors proposed to adapt the algorithm developed to plan the activities of the robots on Mars (which uses simulation techniques) to be used in software projects, using a solution based on agents. According to the authors, these two areas have in common a high degree of uncertainty and the need to adapt the work plan as new information becomes available. The planning of the robot activities is based on a strategy that identifies key location points in order to select the best path to the robot, assisting in the process of decision making through simulation techniques. This same approach, able to prioritize which paths the robot must follow when directed to solve problems of project management, is responsible for prioritizing tasks and resources.

Instead of using simulation techniques, [22] proposes the use of Bayesian networks, due to its ability to model uncertainties and to provide probability estimates for problem solving. The authors state that the main difference between these two approaches is that the simulation typically provides deterministic results and Bayesian networks can produce a set of probabilistic outcomes for each execution, allowing the project manager anticipate situations, even based on incomplete information.

In [23], the author presents a dynamic model of software development process based on Markov decision processes. These stochastic decision processes are an important tool for planning and optimization in environments that involve uncertainties. However, the authors conclude that the computational effort for the optimization of a Markov decision process grows exponentially according to the number of software components.

The work presented by [29] contains a set of integrated tools for modeling, analysis and management of systems. The tool PROSIM, for example, provides mechanisms for modeling, analysis and design of business processes. In [30], ProPlanT, a multi-agent tool that allows planning of production activities and resource selection (based on mechanisms subscribe/advertise), is described.

Finally, [4] proposes to adopt the case-based reasoning to assist in building networks of software project activities, claiming that managers already do this naturally. This technique tries to solve new problems by adapting solutions used to solve previous problems.

Throughout this analysis, it is observed that the surveyed works do not have a solution that allows the planning of software projects considering the interactions of the project manager with other departments within the organization. Thus, there is a need for more study on this subject.

V. EXPERIMENTAL STUDY

The literature provides some approaches based on an experimental strategy to perform the evaluation of models and products where the human factor is considered. The following approaches to evaluate processes, products and resources are suggested by [11]: feature analysis; case studies, surveys, and experiments. Feature analysis is primary used to rate and rank the attribute of a developed software product. However, this strategy does not evaluate the behaviour of the data in terms of cause and effect. Case study is used to organize information about a case and then analyze the contents by seeking patterns in the data and by further analysis through cross comparison with other cases. Survey is a retrospective study to try to document expectations and outcomes in given situations. This technique can also be applied to perform an empirical evaluation of the results with qualitative indicators. However, there is no manipulation of input variables in this study. Experiments, on the other hand, represent a more controlled type of study, usually conducted in laboratories. In this approach, the values of the independent variables (inputs of the experiment) are manipulated to observe the changes in the values of the dependent variables (outputs of the experiment). At the end of the experiment, the results are analyzed, interpreted, presented and packaged.

In this research, the use of a formal experiment method was chosen. However, as the experiment has a quantitative approach [12], an integrated survey is also used in order to evaluate qualitative data. The proposals of [12], [13] an [14] were used as guides to conduct this experiment. The adopted experimentation process includes the following main activities: (1) Objective Definition for the experimentation; (2) Design of the experiments; (3) Execution of the experiments; and (4) Analysis of the results/data collected from the experiments.

A. Objective Definition

The Goal-Question-Metric technique (GQM) [6] was used to define the study, establishing the overall goal, the objectives of the study and measurement. At first, it was decided that the purpose of this research is to compare, in the Unified Process, the accuracy and the effort of integrated planning model SPIM compared to the traditional model of software project planning. Once defined the main objective of the experiment, two issues were identified:

1. The effort to do the activity planning of the software project using the SPIM integrated model is equal to the effort to do the activity planning according to the traditional model?
2. The accuracy in planning the schedule of software projects related to the attribution of deadlines and resources, thinking about the integration with the organizational flows, through the integrated model SPIM is equal to the accuracy to perform the planning according to the traditional model?

For the response variables (accuracy and effort) to be measured we need to have basic metrics that, related, can represent the value of the response variables. The metric associated to Question 1 corresponds to the effort measured by the ratio of spent time in minutes by each participant during the course of the software project activity planning in each method (traditional or using SPIM).

The metric related to Question 2 corresponds to the accuracy related to the attribution of deadlines and resources on the project's schedule activities using each one of the methods, thus, avoiding the occurrence of certain types of risks in the project. For accuracy, in this study it was defined as the ratio amongst the score made by the participants and the total score possible, according to a template previously developed by the researchers.

B. Design of the Experiment

In this phase the researchers should formalize the hypotheses, determine the independent and dependent variables, selection of participants, preparation of the experiment and the conceptual consideration of the validity of the experiment. The researchers selected an 'in-vitro' and 'offline' approach in which participants performed the experiment in a controlled environment. To conduct this experiment, the context involving students of a university was chosen. This approach can reduce risks and costs not covered by the scope of the research at this time. Thus, the experiment was conducted by six undergraduate and four postgraduate students of computer science. After that, based on the previous informal definition of the two issues in this research, it was possible to formalize the two hypotheses and a definition of its measures for evaluation.

The first hypothesis is related to the effort of managers in planning activities and resources for software projects. Then, the first null hypothesis (H_0) is as follows: the effort involved in planning the activities of the software project using the SPIM integrated model is equal to the effort to do the planning of activities according to the traditional model. The effort will be measured by the difference between the start time and the end time (in minutes) of each approach, where:

- Δ_{tspim}: represents the time variation spent in minutes for the planning of project activities using the SPIM;
- Δ_{ttrad}: represents the time variation spent in minutes for the planning of project activities using the traditional model.

The formula of the first null hypothesis is as follows: H0: $\Delta_{tspim} = \Delta_{ttrad}$. As an alternative hypothesis (H_1), the effort involved with the planning of project activities using the SPIM model is not equal to the effort involved in using the traditional model. That is, H1: $\Delta_{tspim} \neq \Delta_{ttrad}$.

The second hypothesis is related to the accuracy of managers to plan the activities and resources in software projects. Then, the second null hypothesis (H_0) is as follows: the accuracy in the schedule planning of software projects related to the assignment of deadlines and resources, thinking about the integration with the organizational flows, through the SPIM integrated model is equal to the accuracy to accomplish the planning according to the traditional model. The accuracy will be evaluated by the ratio of the participants' score and the total score possible, according to a template previously developed by the researchers, where:

- P_{spim}: Accuracy associated with planning using the SPIM model;
- P_{trad}: Accuracy associated with traditional planning.

The formula of the second null hypothesis is as follows: H_0: $P_{tspim} = P_{ttrad}$. As an alternative hypothesis (H_1), the accuracy in the planning of project activities using the SPIM model is not equal to the accuracy accomplishing the planning according to the traditional model. That is, H_1: $P_{tspim} \neq P_{ttrad}$.

After establishing the hypotheses of this experiment, some important characteristics of the experiment were identified. Consequently, it is crucial to be familiar with the terminology used in software experiments. Considering that experimental units are the objects upon which the experiment is run, three different scenarios of software development project were created, aiming to approach different software project risks. The first scenario is related to the assignment's compatibility of the involved stakeholder role with the type of activity (managerial or productive). The second scenario is related with the interaction among the organizational flows to the acquisition of new hardware or software during the project. The third scenario is related to the risk of identifying that the most qualified staff is unavailable at critical moments.

The outcomes of this experiment (response variables) are concerned to the effort and accuracy in planning activities of software projects. Also, any project characteristics (called factors) intentionally varied were identified during experimentation. Each factor has several possible alternatives. In this experiment, there is one factor to be analyzed (project planning methods) and two alternatives: the traditional method of project planning and the method using the integrated planning model SPIM. Some characteristics, however, would be desirable to be invariant, but they vary in an experiment (blocking variables). In this experiment, the level of experience in project planning is a blocking variable.

Considering all the characteristics of this research, it was necessary to find out which one of the two factor alternatives was better (traditional or SPIM) regarding to a given response variable (effort and accuracy). Then, the one-factor designs were chosen. This sort of design involves comparing the response variable to each alternative in a given number of experimental units.

If both alternatives are used in the same project, two similar teams are required. The definition of which participants would perform each approach of software project planning (in the traditional way or with the help of SPIM model) occurred randomly. The experimenter took ten cards (half red and half black) from the pack; the red cards would correspond to the use of the traditional project planning method and the black ones to the use of the SPIM method. The experimenter shuffled the cards and allowed each subject to take a card for each experimental unit (software development project). The balancing principle was also used so that each software project planning proposal was performed by the same amount of participants (five participants for each proposal).

C. Execution of the Experiment

This activity involves the preparation, implementation and validation of data obtained in the experiment. The realization of the experiment occurred in September 2011 and it was performed in the university's computer lab (controlled environment). Initially, all participants received an email inviting them to join this experimental study. In this invitation it was explained that this event included a presentation of SPIM model and the realization of a practical activity where participants would have the opportunity to perform exercises based on typical situations of project management. This was a charge-free event and the seats were limited (maximum 15 participants). Consequently, the experiment involved only students that had some interest in the area of project management.

As mentioned earlier, the studied problem corresponds to three scenarios that simulated situations in software development projects. At first, all participants received a brief training in the SPIM and had the opportunity to test the main features of SPIT on a sample project. Later, they had the opportunity to make the first questions about the proposed work.

The experiment began only after confirmed that all participants understood and could execute the SPIT. Then, they were presented to the same description of each scenario and were asked to perform the corresponding project planning - some using the traditional method and others with the SPIT tool. In order to avoid possible distortions in the obtained results, both in the trial of SPIT and the questionnaire's resolution phase, it didn't occur having any interaction with the interviewer.

D. Analysis

This activity describes how data collected from the experiment was examined in order to draw the conclusions of this research. The methods of analysis can roughly be divided into two major blocks: parametric and nonparametric. Parametric tests require a parametric assumption (such as normality) and non-parametric tests are often used in place of their parametric counterparts when certain assumptions about the underlying population are not met. The choice of each analysis technique depends of the scale type of the response variable. Non-parametric tests are used when the response variable is nominal or ordinal. When the response variable is measured on an interval or ratio scale, then parametric and non-parametric should be applied.

According to [15], parametric tests are statistically more powerful than non-parametric methods. However, if these parametric tests are not conclusive (or valid), then the analysis will have to resort the application of non-parametric tests. Considering these two types of analysis techniques, the drawing of conclusions was attempted by rejecting the null hypotheses with the parametric test or/and accepting them with the non-parametric test.

For the testing of hypotheses, in a context of one factor and two treatments, the literature suggests the significance test called 'T test' for two independent samples (if performed a parametric test) or 'Mann-Whitney test' (if it is a non-parametric test). This definition was taken after verifying if the distribution was normal or not (Shapiro-Wilk test) and checking the variance of the data obtained in the experiment (Levene Test). In the next section the analysis of data obtained in this experiment will be detailed.

VI. ANALYSIS OF RESULTS

According to the scope of this research, it was necessary to evaluate these two hypotheses: effort and accuracy. For the hypothesis analysis of this research, we used the T test (suitable for comparing the averages of a quantitative variable between two independent groups) or Mann-Whitney test (if the test is non-parametric). Then, the verification of each null hypothesis for each developed scenario was performed. It must be noted that the null hypothesis (H_0) is related to the randomness of the observed results, that is, if it is true, statistically, no conclusion can be drawn about the results of the experiment.

The alternative hypothesis (H_1), however, is one that will be accepted if the null hypothesis is rejected. Also, the level of the test significance (p-value) was fixed in 5%. The analyses presented in this experiment were made using the Statistical Package for Social Sciences (SPSS) [16].

A. First Hypothesis: Effort

Initially, we studied the behaviour of each sample (traditional and SPIM) in order to find outliers. An outlier is an observation that lies an abnormal distance from other values in a random sample from a population [17]. According to the boxplot graph it was observed that the effort variable does not have outliers.

After that, it should be verified whether the data has or not a normal distribution. To achieve this goal, a null hypothesis and an alternative hypothesis were defined, as follows: H_0: it is a normal distribution; H_1: it is not a normal distribution. The Shapiro-Wilk test can be used to check if the data distribution is normal (see Table 2).

Variable	Methods	Statistic	Degr. Freedom	Significance
Effort	Traditional	0.891	5	0.364
	SPIM	0.968	5	0.863

Table 2. Results of the Shapiro-Wilk test for the effort variable.

According to the results, it may be observed that the level of significance in both samples are greater than the level of significance that can reject the null hypothesis (0.05 or 5%). Thus, there is no evidence to reject the null hypothesis in which the distribution is normal. However, the T test also assumes that the variability of each group is approximately equal. With this goal, two hypotheses were defined: H_0: The variances are equal; H_1: The variances are not equal. The Levene's test for equality of variances (see Table 3) shows if its assumption of the T test has been met.

Variable	Assumption	Significance
Effort	Equal Variances assumed	0.860
	Equal Variances not assumed	0.860

Table 3. Levene's test for the effort variable.

According to the results, the significance (p-value) of Levene's test is 0.860. If this value is lower than or equal to the significance level (α) for the test (in this case 0.05), then the null hypothesis in which the variability of the two groups is equal may be rejected, implying that the variances are unequal. If the p-value is greater than the α level, then, equal variances are assumed. In this case, 0.860 is greater than α, so the fact that the variances are equal was assumed. Once it was identified that the distribution was normal and variances were equal, the T test was applied (see results in Table 4).

Variable	Criterion	T	Degr. of freedom	Sig. (2 tailed)
Effort	Equal Variances assumed	-2.070	8	0.073
	Equal Variances not assumed	-2.070	7.749	0.073

Table 4. T test for the effort variable.

This is a two-sided test, in which the p-value = 0.073 is directly compared with α = 0.05 (significance level). Since p-value = 0.073 > 0.05, H_0 is not rejected. Thus, there is no statistical evidence to reject the hypothesis that the effort average to accomplish the planning of the activities using the traditional model is equal to the spent effort with the SPIM.

B. Second Hypothesis: Accuracy

Similarly to the analysis of the first hypothesis, it was observed that the accuracy variable does not have outliers. The next step was to identify if the data has a normal distribution. To achieve this goal, the following hypotheses were defined: H_0: it is a normal distribution; H_1: it is not a normal distribution. The Shapiro-Wilk test shows if its assumption of the T test has been met (see Table 5).

Variable	Methods	Statistic	Degr. Freedom	Significance
Accuracy	Traditional	0.960	5	0.269
	SPIM	0.961	5	0.814

Table 5. Results of the Shapiro-Wilk test for the accuracy variable.

Also in this case, the level of significance in both samples is greater than the level of significance that can reject the null hypothesis (0.05 or 5%). Thus, there is no evidence to reject the null hypothesis in which the distribution is normal. We also applied the Levene's test of equal variances for the accuracy variable (see Table 6) considering the following two hypotheses: H_0: The variances are equal; H_1: The variances are not equal.

Variable	Assumption	Significance
Accuracy	Equal Variances assumed	0.173
	Equal Variances not assumed	0.173

Table 6. Levene's test for the accuracy variable.

Directly comparing p-value with α (significance level), it is observed that p-values in both samples reject H_0. Then, considering that the distribution was normal and variances were equal, the T test was applied (see Table 7).

Variable	Criterion	T	Degr. of freedom	Sig. (2 tailed)
Effort	Equal Variances assumed	-5.774	8	0.000
	Equal Variances not assumed	-5.775	5.223	0.002

Table 7. T test for the effort variable.

Since the p-value is lower than the significance level (α) for the test (0.05), then the null hypothesis may be rejected. Thus, there is statistical evidence to reject the hypothesis that the accuracy to accomplish the planning of the activities using the traditional model is equal to the accuracy obtained with the SPIM model. Based on the results presented for the accuracy variable it is understood that there is a difference between the mean effort to do the planning with the traditional and SPIM methods. Comparing the mean values of the SPIM approach (82%) with the traditional approach (57%) we conclude that the precision in making the planning model using the SPIM is larger than in the traditional model.

VII. QUALITATIVE ANALYSIS

At the end of the experiment's execution, each participant answered a questionnaire, produced in accordance to [20]. The survey had 10 questions where the first 6 were focused on knowledge mapping of project managers and the remaining were used to estimate the SPIM model's contributions in the planning process from the project managers' point of view. An analysis of the obtained results from the questions related to the profile of respondent individuals shows that these had an average of 4.88 years of professional experience (min. = 2, max. = 8). In addition to that experience of the respondents, 29% of the sample reported their experience in project management as little, while the remaining 71% declared it as moderate or advanced. In addition, 78% of subjects classified their knowledge of software development processes as moderate or advanced. This indicates the subjects sufficient range of experience regarding to project management.

We begin the analysis of the SPIM with the respondents' evaluation of the direct benefits in performing the integrated planning of managerial and productive activities in a software project. When questioned whether or not they agreed on the distinct nature of the three types of activities, most of the respondents (80%) answered that the SPIM helps managers to access enterprise workflow information. Also, 90% of the interviewed subjects agreed that the SPIM model contributes in identifying the dependencies of the activities between the project workflow and the organization workflow. The majority of the interviewed subjects found that the SPIM model contributes in the identification and measuring of the indirect costs of the project, due to the management support activities. As a final consideration, 60% of the respondents noticed a reduction in time during the project's elaboration process.

VIII. FINAL CONSIDERATIONS

This article presented the SPIM, a model to integrate software project management with organizational workflows. An experimental strategy was chosen to evaluate the proposed model. This experiment aimed to compare the accuracy and the effort of integrated planning model SPIM compared to the traditional model of software project planning, considering the characteristics and particularities involved in the Unified Process. It was first identified the need of software project managers to access information from other departments of the organization during the project planning. Then, the concepts proposed by the SPIM integrated model and the features of the SPIT tool were presented. After that, the planning and execution of the experiment analyzing two distinct project management methods (traditional and SPIM) was presented. Finally, the results of an experiment conducted with a group of undergraduate and postgraduate students were presented.

This work contributes with some interesting findings related to the software project management. The experiment reveals that the use of the SPIM model approach helps managers to create and conduct a more precise project plan than the traditional method. 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. The obscurity in identifying this kind of relationship during the planning and execution of a software project can negatively affect the project schedule. This evidence gets clear during this empirical study while analyzing accuracy variable.

An evidence related to the effort variable could also be extracted: the time for planning the activities using the SPIM model is similar to the traditional model. The idea behind the SPIM model comes from the need to reduce the complexity in visualizing the interdependencies of both organizational workflows and individual project's workflow of activities. Most of the effort when using the SPIM model is related to the filling of the extra information proposed by this model. Nevertheless, the results of the effort variable did not become favorable to the traditional method.

The results of this experiment reaffirm the benefits that the SPIM model provides in solving problems regarding to the inadequate definition of tasks due to the obscurity in visualizing the interdependency between the organization's and project's specific workflows.

References

[1] K. Schwalbe, "Information Technology Project Management", Thomson Learning, 2nd edition, 2002.

[2] Project Management Institute, "PMBOK - A Guide to the Project Management Body of Knowledge" Newtown Square, PA: Project Management Institute, 4th edition, 2008.

[3] D. Joslin and W. Poole. "Agent-Based Simulation For Software Project Planning". In Proceedings of the 37th Conference on Winter Simulation. Orlando, Florida, 2005, pp 1059-1066.

[4] B. Lee and J. Miller. "Multi-Project Management in Software Engineering Using Simulation Modelling". In: Software Quality Journal,vol. 12, 2004, p.59-82.

[5] T. Abdel-Hamid and S.E. Madnick. "Software Project Dynamics: An Integrated Approach". Englewood Cliffs, NJ, Ed. Prentice Hall, 1991.

[6] R. Solingen and E. Berghout. "The Goal/Question/Metric Method". New York: McGraw-Hill, 1999.

[7] D. Callegari, M. Rosito, M. Blois, R. Bastos. "An Integrated Model for Managerial and Productive Activities in Software Development", In: ICEIS - 10th International Conference on Enterprise Information Systems, Spain, 8p, 2008.

[8] P. Kruchten, "The Rational Unified Process: An Introduction", Addison-Wesley, 2nd edition, 2000.

[9] I. Graham, B. Henderson-Sellers and H. Younessi, "The OPEN Process Specification", Addison-Wesley, 1997.

[10] C. Chatfield and T. Johnson. "Microsoft Office Project 2010 Step by Step". Redmond: Microsoft Press, 2010.

[11] S.L. Pfleeger and J. Atlee. "Software Engineering: Theory and Practice", 4th edition, Prentice Hall, 2009.

[12] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell and A. Wesslén. "Experimentation in software

engineering: an introduction". Boston: Kluwer Academic Publishers, 2000, 204 p.
[13] N. Juristo and A. Moreno. "Basics of Software Engineering Experimentation". Kluwer academic Publishers. 2nd edition, 2003.
[14] A. Field. "Discovering Statistics Using SPSS". 2nd edition. Sage Publications, 2005.
[15] M.B. Miles and A.M. Huberman. "Qualitative Data Analysis", 2nd edition. London: Sage Publications, 1994.
[16] IBM. "SPSS Statistical Analysis". viewed July 25, 2011, <http://www-01.ibm.com/software/analytics/spss/products/statistics>.
[17] F.E. Grubbs. "Procedures for detecting outlying observations in samples". Technometrics, 11, 1969, p1-21.
[18] B. Myers. "Foundations of WF: an introduction to Windows Workflow Foundation". Apress, 2007.
[19] M. Rosito, D. Callegari and M. Blois, R. Bastos. "Project Management and Software Development Processes: Integrating PMBOK and OPEN". In: WASET - World Academy of Science, Engineering and Technology, Malaysia, 2012.
[20] L.M. Rea and R. Parker. "Designing and Conducting Survey Research: A Comprehensive Guide", Jossey-Bass. 33rd edition, 2005.
[21] D. Callegari, and R. Bastos, "Project Management and Software Development Processes: Integrating RUP and PMBOK", In: ICSEM - International Conference on Systems Engineering and Modeling, 2007.
[22] C. Fan, Y. Yu. "BBN-based software project risk management". Journal of Systems and Software, Vol. 73, Issue 2, Applications of statistics in software engineering, pp. 193-203, 2004.
[23] F. Padberg. "Linking Software Process Modeling with Markov Decision Theory". In Proceedings of the 28th Annual International Computer Software and Applications Conference,pp. 152-155 vol.2, 28-30, 2004.
[24] R. Dewson. "Beginning SQL Server 2008 for Developers: From Novice to Professional". Apress, 2008.
[25] M. Jungel, E. Kindler and M. Weber. "Towards a Generic Interchange Format for Petri Nets - Position Paper". In XML/SGML based Interchange Formats for Petri Nets, Aarhus, Denmark, 2000.
[26] M. Rosito and R. Bastos. "A Systematic Review on the Integration of Project Management with Organizational Flows". In: WASET - World Academy of Science, Engineering and Technology, Malaysia, 2012.
[27] B. Kitchenham. "Procedures for performing systematic reviews". Joint Technical Report Software Engineering Group, Department of Computer Science, Keele University, 33p, 2004.
[28] J. Biolchini, P.G. Mian, A.C.C. Natali and G.H. Travassos. "Systematic review in software engineering". Rio de Janeiro, 31p, 2005.
[29] D. Delen, P. Benjamin and M. Erraguntl. "An Integrated Toolkit for Enterprise Modeling and Analysis", In: Proceedings of the 1999 Winter Simulation Conference, pp. 289-297, 1999.
[30] A. Riha, M. Pechoucek, H. Krautwunnova, P. Charva and A. Koumpis. "Adoption of an Agent-Based Production Planning Technology in the Manufacturing Industry", In: 12th International Workshop on Database and Expert Systems Applications, pp. 640-646, 2001.

Author Biographies

Mauricio Covolan Rosito is a Ph.D. student at the Department of Computer Science, Pontifical Catholic University of Rio Grande do Sul, Brazil. He received MSc. degree in Computer Science at 2008. His current research interests are Software Project Management, Processes of Software Development and Experimental Software Engineering.

Ricardo Melo Bastos is a Professor at the Pontifical Catholic University of Rio Grande do Sul, Brazil. He obtained his Ph.D. degree in 1998 from Federal University of Rio Grande do Sul, Brazil. His current research interests are Software Project Management, Modeling and Development of Object-Oriented Systems and Agent-Oriented Software Development.