PSP support component integrated into a web project management environment

Antonio Marcos Neves Esteca
rogeria@ibilce.unesp.br

Rogéria Cristiane Gratão de Souza
adriana@ibilce.unesp.br

Adriana Barbosa Santos
valenicio@ibilce.unesp.br

Carlos Roberto Valêncio

Vanessa dos Anjos Borges
vanessaborges123@gmail.com

Computer Science and Statistics Departament
UNESP – São Paulo State University
São José do Rio Preto, São Paulo, Brazil

Abstract – Studies show that the unsatisfactory results in software projects are often related to the lack of training and commitment of human resources. In this context, this paper presents a proposal that integrates the employment of the practices proposed by the Personal Software Process (PSP) into the context of project management, aiming to support the self-improvement and commitment of each human resource, helping to manage their individual goals, which contributes to the success of projects as a whole. For this purpose, a PSP support component was created and integrated into a web project management environment, called System to Aid Project Managing (SAPM). The results were evaluated in two steps: a comparative analysis between the new version of SAPM and PSP support tools available in the market and an evaluation by a group of 22 participants, including project managers and software developers. The results revealed that the component built is broader than other tools, with the differential of bringing benefits in the context of project management.

Keywords - project management; personal software process; web-based tool

I. INTRODUCTION

The quality of software development process helps in getting a quality product, besides allowing the repetition of good results [1]. Therefore, it is necessary to incorporate a set of best practices to the software process, among which is project management [2].

Despite the continuous improvement of managerial techniques, the results obtained in software projects are still far from the expected. According to the CHAOS study [3], 42% of completed projects show changes concerning initial estimates, 21% are aborted, and only 37% are successfully completed and within initial expectations. Among the reasons given by the CHAOS study for the low success rate of software projects are the lack of training and commitment of human resources involved [4].

System to Aid Project Managing (SAPM) is a project management support web environment initially conceived to support management practices [5]. Seeking to contribute to the improvement of human resources maturity by encouraging them to maintain greater discipline and self-improvement during the achievement of their activities, this paper presents a component that was integrated into SAPM to support the employment of Personal Software Process (PSP), which aims at helping software developers to control, manage and improve their work [6]. As a result, besides SAPM enables the use of best management practices by project managers, now each human resource counts on mechanisms to manage themselves, which assists in meeting individual goals, contributing to the success of projects as a whole.

The rest of this paper is organized as follows: Section II shows an overview of PSP and related works; Section III shows the results obtained with SAPM expansion through the integration of the PSP support component; Section IV shows the results evaluation; finally, Section V shows the concluding remarks and proposals for future works.

II. PERSONAL SOFTWARE PROCESS

PSP was created by Watts Humphrey to be applied at a personal level by software developers aiming at their continuous improvement. For this, PSP proposes the division of software process into phases and the use of forms to record measurements carried out. Additionally, PSP is divided into six maturity levels to be achieved in a gradual manner, namely [6]: PSP 0 – it maintains the current development process and incorporates the registration of the time spent on each task and the defects generated; PSP 0.1 – it adds the measurement of size of the programs designed and the registration of process improvement proposals; PSP 1 – it adds the generation of size and time estimates to develop the programs and the creation of test reports; PSP 1.1 – it incorporates the creation of schedules to aid in the accomplishment of tasks based on estimates; PSP 2 – it incorporates reviews of codes and of the project; PSP 2.1 – it adds standard templates to design programs.

Empirical studies show that PSP provides several personal and organizational benefits, such as [6, 7]: decreased rate of defects; increase in the productivity; improvement in the estimates generated; reduced costs for software tests; increased chance of project success as a whole due to the proper achievement of the goals established at an individual level.

Given the benefits from PSP, several related works are being developed. One of the research strands conducted aims to propose tools to support the improvement of specific
aspects related to the increase of human resources productivity [8, 9] or improvement of the quality of software products developed [10, 11]. Despite the potential of these works, none of them enables the integration of PSP into a project management environment as SAPM, contributing to the success of software projects.

In addition to research with automation propositions of specific activities, several studies have aimed at improving the personal software development process as a whole. In this context, various studies have proposed computational systems to support PSP employment [12, 13, 14, 15, 16, 17]. However, most of these proposals do not provide support at all PSP levels, or when they do, no implementation support of important elements is offered, such as time measurement of defects correction and process indicators generation.

In this scenario, it is possible to note the importance of this work, which presents a component to support the implementation of all PSP elements with the important advantage of being integrated into a project management environment, the SAPM. This integration enables human resources to act according to their duties in the projects, contributing to the achievement of the project goals as a whole.

III. PSP SUPPORT COMPONENT

The development of the proposed component started with the study of the main PSP guides [6, 18] in order to identify the elements and techniques to be supported. Next, the requirements were classified and grouped into a new component to be incorporated into the SAPM software architecture [5]. Finally, after restructuring the SAPM software architecture, the identified requirements were implemented and integrated with SAPM environment, using the following computing resources: PHP, JavaScript and HTML; database manager MySQL; Apache web server.

The model established for SAPM operation is shown in Figure 1. As presented in the illustration, the system was divided into two access areas. The Managerial Area is dedicated to the execution of managerial activities under the responsibility of managers and project team members. Its functions are presented by Ref [5]. The Personal Area is provided by the component proposed by this work and it is dedicated to the employment of PSP techniques at a personal scope. This model was established in order to avoid the overburdening of the software process. Since each team member is responsible for the personal application of PSP techniques, the project managers remain responsible only for management activities. Thus, according to the model, project managers will assign activities to team members through the Managerial Area. In turn, the team members will receive these activities in the Personal Area, where they may manage them using the resources provided.

By accessing the system, a user can select a registered project to which it is allocated. At the first access to each project, users must define if they want to use PSP support. Figure 2 (a) illustrates the main page of the Managerial Area, shown after the selection of a project. In this figure is highlighted the main menu item related to PSP. Figure 2 (b) illustrates the Personal Area main page accessed by the link "Personal Area" in the main menu of SAPM. In the figure example, was selected a project in which the user Bob is at the last maturity level of PSP. The users achieve a new PSP level when they complete a project applying the previous level, which helps them to mature gradually.

Next, the functions provided in the Personal Area are briefly presented according to each PSP level.

- **Level 0**

The main functions concerning PSP level 0 are: registration of programs to be constructed; registration of tasks involved in the construction of each program; use of Time Recording Log; use of Defect Recording Log; sending programs to test; record of completion of tasks and personal programs.

The register of tasks involved in the construction of programs can be quickly performed, since the system allows the registration of standard tasks, which can be subsequently imported to other programs without the need for typing.

For the employment of Time Recording Log, the user must record the time spent on each task and interruptions occurred.

The use of the Defect Log Recording, in its turn, enables to record the type of each defect found in programs developed and the development phase in which such defects were injected and removed.

Figures 3 and 4 show the devices made available by SAPM to support employment of Time Recording Log and Defect Recording Log, respectively, which operate in the background and record the necessary information by means of stopwatches, which can be controlled by shortcut keys. In the case of Time Recording Log, the only information that the user needs to type is a word that describes each interruption. In turn, for the Defect Recording Log, the user needs only to select the phases of injection and removal of defects, besides the type of defect.

From the data recorded by Time Recording Log and Defect Recording Log, are generated various personal indicators of productivity and quality, such as the percentage of time spent on interruptions and the rate of generation of defects, contributing to developers’ self-knowledge, which is one of the main goals of PSP 0.
Also at level 0, users can send for test the files containing the source code of their programs, which are made available to all testers allocated to the project on a page of Managerial Area referring to Quality Management. When a tester accesses the program, he has at his disposal the device for the employment of the Defect Recording Log. Data on the tests performed are stored for the program developer, in order to maintain registers that will be used to generate indicators. Finally, users can record completion of each task and program. Furthermore, all information collected and generated indicators can be viewed on Project Plan Summary, which is available at all PSP levels. At each level, new information is added to the Summary, as proposed by the PSP.

- **Level 0.1**

The main functions on PSP level 0.1 are: registration of a coding standard for each programming language; registration of data obtained from software measurements; registration of proposals for improving the process.

The system provides a page where users can register a coding standard adopted for each programming language, being possible to upload a file with the existing standard.

The registration of data obtained from program measurements should be done by filling out a form. In the specific case of counting the lines of code in a program, despite the fact that SAPM does not perform automatic classification, the system indicates internet addresses for downloading free programs that perform this process.

Finally, at level 0.1, the system provides a form to record process improvement proposals, which are also stored as lessons learned at the Managerial Area.

- **Level 1**

The main functions concerning PSP level 1 are: automation of the PROBE method, which aims at generating estimates of size and time required for developing programs; record of estimated time to accomplish each task; registration of tests.

Although PROBE method generates estimates very close to reality, its implementation requires complex procedures and calculations. Therefore, the solution implemented in SAPM automates most of the execution of the method, making its complexity transparent to the users. The only entries that the user needs to provide the system with are: definition of a proxy for each programming language, which consists of a program component whose size is proportional to the time of development [6]; registration of proxy size in programs already constructed using PSP levels 0 and 0.1; registration of actual size for programs whose estimates were performed using the PROBE method. From the record of the actual data size, the degree of accuracy of PROBE method increases with each new program.
From the total estimated time for the development of programs, users can establish the estimated time to develop each task related to the program.

Finally, the user can also record program test reports or send these programs to be tested by software testers.

- **Level 1.1**

The main functions concerning PSP level 1.1 are: distribution of total estimated time for program development between PSP phases; generation of schedule for personal tasks, presenting the Cost Performance Index (CPI) and project progress; presentation of comparison between allocations of human resources made by project managers, and the work plan established by such resources.

The distribution of the total estimated time between PSP phases is automatically performed based on the historical percentage of time spent in each phase. This distribution helps users of the system to define the time of each personal task, based on the estimated duration of PSP phases.

From the sequencing of programs and its activities that must be performed by the user, the system automatically generates a personal schedule for each PSP user, which is represented by a Gantt chart. Along with the schedule, it is also presented the CPI value and the percentage of project progress, calculated by the Earned Value Management (EVM) technique [6].

Finally, at level 1.1 is presented a comparative between human resource allocations made by project managers and by their own human resources, as shown in Figure 5. This comparison can direct managers and team members to review the estimates made, in case of significant differences.

![Figure 5. Comparative: estimates from project manager vs. personal estimates.](image)

- **Level 2**

The main functions concerning PSP level 2 are: record of checks performed by means of design review checklist and code review checklist; automatic calculation of indicators of quality and productivity.

Users can apply the checklists proposed by PSP for each program, recording the items checked in the design review and code review phases.

Finally, the system provides a dashboard to display the following indicators on quality and productivity [6]: Defect Removal Yield; Defect Density; Defect Rate; Cost of Quality (COQ); Defect Removal Leverage; Process Quality Index; CPI; Lines of Code/Hour; Time Spent on Interruptions. In addition, the system also allows you to view the evolution of these indicators along the projects, which helps to assess the evolution of personal maturity. In Figure 6, it is illustrated, as an example, the COQ and its evolution.

![Figure 6. Example of indicators in the Personal Area.](image)

- **Level 2.1**

The main function concerning PSP level 2.1 is the registration of the functional, operational, logical and state specification templates for each program, which represent in tables, respectively, the class diagram, the use case diagram, the pseudo-code and the state diagrams [6].

**IV. EVALUATION**

Results evaluation was performed in two steps: comparative analysis between the new version of SAPM and PSP support tools available in the market, revealing a greater breadth of SAPM; evaluation of the results by a group of 22 participants, including project managers and software developers, which demonstrated the benefits offered by the proposed approach, as well as detected points to be improved, guiding future works.

These two steps are presented in the following subsections.

A. **Comparative analysis with other PSP support tools**

This step consisted in the comparative analysis between SAPM and six other PSP support tools available in the market, namely: Process Dashboard [12]; Jasmine [13]; Hackystat [14]; PSPA [15]; PSP.NET [16]; Eclipse Plugin [17]. The selection of these tools include the most cited in articles consulted during the study of related work. After selection, the requirements to be analyzed were defined, which sought to evaluate the scope of the support to the elements and practices of each PSP level. Such requirements were assessed using an observational analysis that considered a range of 0 to 100%, ranging from 25 to 25%, depending on the coverage offered by the tools to the items considered. Thus, if one criterion is assessed with 75% in tool A and with 50% in tool B, for example, tool A permits an improved approach to this aspect.

Table III shows the results of the evaluation. As it can be seen, in general, SAPM presents coverage equal to or better than the other tools evaluated, except for the requirement “Measurement and Classification of lines of code”, for which the system indicates internet addresses for specific tools that may be easily used together with SAPM.

One of the outstanding points identified is that most tools offer little or no support for PSP 1, which is one of the most important levels, since it allows to obtain very accurate estimates of size and time for software development. Moreover, only PSP.NET tool approaches level 2.1. Finally, it is worth highlighting that SAPM is the only tool analyzed that is integrated into a project management environment.
software developers. In Figure 7, it is shown the profile of the development enterprises, being 8 project managers and 14 professionals from small and medium sized software programs built, making it necessary the use of other tools to obtain measurements; does not measure and automatically sorts the lines of code of registration of other types; only the types of defects proposed by PSP, not allowing the main shortcomings observed were:

3. Registration of defect type standard: SAPM considers only those items proposed by PSP in the review checklists, not allowing the registration of additional.

B. Vision of SAPM users

The objective of this stage was to analyze the vision of users concerning the contribution of this work. With this aim, it was performed a course that counted on the participation of 22 professionals from small and medium sized software development enterprises, being 8 project managers and 14 software developers. In Figure 7, it is shown the profile of the participants as to the length of professional experience, revealing the extent of the selected sample. Despite the difference in the level and length of professional experience of participants, the answers received during the course presented no significant differences.

The course was divided into two parts, performed in 4 consecutive days with 4 hours a day.

The first part took place in the first 3 days and consisted of the application of lectures that addressed all the PSP elements and practices, with the goal of ensuring that all participants had knowledge of the process.

The second part took place on the last day of the course and consisted of presenting SAPM, addressing both the Managerial Area and the Personal Area, with a focus on PSP support functions.

At the end of each stage of the course, a questionnaire was administered to participants. It is observed that the participants were not identified in the questionnaires, ensuring an unbiased assessment. Also, when starting the second questionnaire it was not possible to alter the responses of the first.

In the first questionnaire, the 14 software developers made an overall assessment of the PSP, judging three assertions as to the degree of agreement, measured on a scale of 1 (strongly disagree) to 5 (strongly agree). The assertions were: Learning (L) - A developer who does not know PSP can learn the practices in a short time; Simplicity (S) - PSP simplicity encourages its use; Motivation (Mo) - I intend to use PSP in my routine.

Now, the second questionnaire was divided into two parts. The first part consisted of the same questionnaire used in the first step, in order to verify any changes in the vision of the software developers after presenting SAPM. The second part sought to determine the view of project managers as to the contribution of PSP to the project management, therefore, it was answered only by the 8 project managers that participated. The following items were evaluated on a scale of 1 (worst evaluation) to 5 (best evaluation): Estimate (E) - Aid for time estimation of activities; Progress (Pr) - Aid for project progress control; Quality (Q) - Aid for project quality control; Indicators (I) - Importance of indicators generated.

Moreover, at the end of the first and the second part of the second questionnaire, open fields were left to describe the strengths and weaknesses of SAPM.

Figure 8 shows the comparison between the evaluation of the statements before and after the presentation of SAPM, considering the average of the scores given by the participants. Observing the graph, it is clear that there was a strong increase in agreement as to the assertions, which allows to affirm that participants felt more interested to use PSP due to computational support that met their needs.
needs for changes in estimates generated.

Although the results obtained were quite favourable, future studies will be developed to improve the proposed solution concerning weaknesses. Furthermore, in order to contribute to the advancement of research in the context of project management, data collected through the individual employment of PSP will be analyzed to identify indicators that can reflect the behavior of the team involved in the project, which should be made available to project managers to guide their decisions.

ACKNOWLEDGMENT

This research was partially supported by the State of São Paulo Research Foundation under projects number 2008/07094-2 and 2010/13478-8.

REFERENCES