A Body of Knowledge for Executing Performance Analysis of Software Processes

Natália Chaves Lessa Schots, Ana Regina Rocha
COPPE/UFRJ
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
{natalia, darocha}@cos.ufrj.br

Gleison Santos
PPGI/UNIRIO
Universidade Federal do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
gleison.santos@uniriotec.br

Abstract—The periodic execution of process performance analysis represents an important step for an organization to achieve quality in their processes and products. However, it is possible to notice that few software development organizations can establish and maintain this practice effectively. Many of the difficulties in executing performance analysis are related to the lack of knowledge in the field and difficulties of learning that are due to limitations and dispersion of currently available literature. In this sense, this paper presents the definition of a body of knowledge in process performance analysis, which will be made available through a knowledge-based system, aiming to support the execution of performance analysis in software development organizations. Particularly, we highlight the description of which knowledge the system must provide, as well as the ways how this knowledge is being captured and presented. For helping the understanding of the body of knowledge, an example of use is illustrated.

Keywords- Process Performance Analysis; High Maturity Practices; Body of Knowledge; Knowledge-based Environment; Knowledge Management

I. INTRODUCTION

In software development organizations that are looking for improvement in their processes and products, the adoption of techniques for process performance analysis is essential, since these techniques allow them to learn about their process, as well as control and predict them through the process metrics and statistical techniques [1].

There are many works in the literature that discuss the application and benefits of using process performance analysis in software, e.g. [2][3][4]. However, it is possible to see that few software development organizations adopt this practice in their organizational culture. An evidence of this scenario is the low number of evaluated software organizations in CMMI levels 4 or 5; according to [5], only 8% of the organizations evaluated are/were on these levels. Furthermore, few works in the literature show the effective implementation of process performance analysis in the software field [6].

The main difficulties pointed by the organizations for not implementing process performance analysis are: (i) lack of planning and collection of adequate software metrics; (ii) lack of knowledge on the techniques and methods to execute process perform analysis; and (iii) lack of knowledge about the data required to adequately perform the analysis [3][6][7].

As it can be seen in the literature reviews and in practice through consulting experience, process performance analysis is a human-dependent activity, as it requires several decision-making actions that cannot be completely solved by a computer. Due to this, knowledge becomes a critical success factor [8]. These decision-making actions require deep technical knowledge of performance analysis and a good understanding of the process being analyzed.

In fact, for an effective process performance analysis, it is mandatory for those responsible for this activity to have technical knowledge of the concepts and techniques to carry out performance analysis; one should also have knowledge of both the process and the organizational context in which the analysis will be performed, as otherwise the benefits of the analysis may not be achieved. As it can be seen, knowledge is one of the most important assets in an organization that seeks for excellence in the development of its products and services [1]. Thus, it must be properly managed, so that such organization can take advantage from its several benefits.

In this sense, knowledge management practices can play an important role to support this kind of activity by collecting, storing and applying the right knowledge to the right people at the right time. In order to support performance analysis by using knowledge management practices, a knowledge-based environment named SPEAKER (Software Process pErformance Analysis Knowledge-based EnviRonment) is being developed. This environment aims to provide a body of knowledge on the concepts, tasks, and techniques of process performance analysis, designed to support software development organizations to perform this analysis properly.

Thus, this paper aims to describe the body of knowledge that will be made available through the proposed environment, as well as how this environment must deal with the management of such knowledge. To this end, the paper aims to answer the following questions: (Q1) which knowledge should be made available for supporting the execution of process performance analysis? and (Q2) how this knowledge should be documented in SPEAKER?

The remainder of this paper is structured as follows: Section II presents basic concepts about performance analysis and knowledge management. In Section III, the body of knowledge for software process performance analysis is described, and a hypothetical example of its use is presented in
Section IV. Some considerations on how the proposed knowledge-based environment will manage this knowledge are presented in Section V. Section VI presents some related works. Finally, Section VII provides some final remarks.

II. BACKGROUND AND MOTIVATION

Process performance analysis involves a set of methods and techniques first targeted at understanding the process and then analyzing its stability [1]. The execution of process performance analysis is one of the requirements defined by maturity models (such as CMMI-DEV [9] and MR-MPS-SW (the Reference Model of the Brazilian Software Process Improvement Program for Software) [10]) and quality standards (such as ISO 15504 [11]).

In order to use measures to predict future results or as a basis for process improvement, it should be ensured that the process behavior is stable [12]. A process is stable when its variability is due only to common causes, i.e., caused by inherent factors that are part of the process. If the process is unstable, variability is caused by special causes (also called assignable causes), i.e., caused by events that do not usually occur during the process execution, and thus should be prevented [1][12]. When variability is due only to common causes, the process behaves between known limits and thus it becomes possible to predict it [12].

According to [1], a process performance analysis aiming at continuous process improvement can be accomplished by executing the following tasks: (i) clarify business goals; (ii) identify and prioritize process issues; (iii) select and define measures of processes or products; (iv) collect, verify and retain data on process execution; (v) analyze process stability; and (vi) analyze process capability.

The main tools used for analyzing process stability are control charts [12]. From them, it is possible to verify whether the process variability is due only to common causes or not. There are many types of control charts, and each one can be more appropriate for a specific context. For an adequate control chart, one needs to check the question to be answered, the type of data measured (variables or discrete), the size of subgroups (if applicable), the data distribution model, among others [1].

Although each type of control chart has some rules for detecting if a process is stable, they are difficult to analyze when the responsible for the analysis is not aware of both the process and the organizational context. Moreover, the wrong type of control chart can lead to mistakes in the analysis (e.g., indicating that a process is stable when it is not). In such cases, only an expert may be able to notice this issue and analyze the data with another type of control chart.

In order to support a non-expert person in the execution of performance analysis activities, a knowledge-based system (KBS) can be used. It aims at imitating human reasoning in problem-solving for assisting decision making [13], with a combination of artificial intelligence techniques and a database of specific knowledge. The concept of KBSs is widespread amongst academics and researchers [14]. Particularly, many knowledge management initiatives have been proposed in software engineering, such as [15] and [16]. These studies illustrate the synergy between these two areas.

Basically, knowledge-based systems seek to implement knowledge management activities in an organization. Usually, the main activities are knowledge identification, knowledge acquisition, knowledge development, knowledge dissemination, knowledge utilization, and knowledge maintenance [17]. However, such activities may vary depending on strategic goals and the organizational context.

Knowledge management deals with two kinds of knowledge: explicit and tacit. Explicit knowledge, also known as “codified knowledge”, can be easily described in a textual or symbolic form. Tacit knowledge, in turn, is on people’s minds and is hard to express [17]. In this work, both kinds will be captured and made available in the body of knowledge.

III. A BODY OF KNOWLEDGE FOR SOFTWARE PROCESS PERFORMANCE ANALYSIS

Based on the execution of a systematic mapping study (briefly described in the Section VI), in addition to literature reviews, it could be noticed that there is a lack of works that support process performance analysis by knowledge management practices. Moreover, there are only a few papers in the literature that describe process performance analysis in software, as pointed out in [6]. This scenario makes the learning and adoption of process performance analysis by software organizations more difficult, even for those organizations that already have achieved intermediate maturity levels (e.g., CMMI level 3 or MR-MPS-SW level C).

The knowledge base is a major component of a KBS. Therefore, for developing SPEAKER, the knowledge to be made available was identified and organized in a body of knowledge. Such body of knowledge is presented in the following subsections, highlighting the knowledge required for executing properly the performance analysis (answering Q1 showed in Section I), as well as how this knowledge is being documented in SPEAKER (answering Q2).

A. Knowledge required to process performance analysis

Based on the technical literature about performance analysis in software (e.g. [1][9][10]) and also in non-software area (mainly in manufacturing area) (e.g. [12]), a set of activities and tasks was identified. They served as input for the identification of the required knowledge to execute software process performance analysis. The main activities and tasks are listed in Table 1. This knowledge involves the breakdown of activities and tasks, and the description of techniques or methods that can be used for supporting the execution of performance analysis. It also involves the requirements expected by maturity models and quality standards. The description of each technique or method has the following structure: name, description, application context, data characteristics to which the method applies (variable or attribute, sample size, data distribution, etc.), and a practical example in the software area.

For composing the body of knowledge, both types of knowledge (explicit and tacit) are intended to be captured. In the context of this work, explicit knowledge is that spread in books, papers, journals and academic theses, while tacit knowledge is collected from interviews with experts.
The explicit knowledge involves technical knowledge about quantitative and statistics methods, and is present in the literature on process performance analysis in both software and not related to software areas. In the body of knowledge, when such knowledge comes from an area not related to software, an analysis is made to verify whether it has been already applied to software or not, and if some adjustment is necessary. In order to verify if the knowledge to be stored in SPEAKER is correct and adequate, peer reviews with experts in the corresponding key areas are planned. Thus, it is expected to minimize the risk that the knowledge provided by the environment does not achieve its purpose.

Tacit knowledge, in turn, is rarely described in books or papers and it is difficult to capture [17]. However, tacit knowledge can be partially converted to explicit knowledge and then be managed by a knowledge system [8]. To this end, in this work, tacit knowledge is collected and enriched by performing successive interviews with experts, who describe how one executes each performance analysis activity and in which context. After that, the collected knowledge is included in the body knowledge of the system, becoming an explicit knowledge. These experts are involved in the implementation of high maturity level in software organizations or are certified appraisers to lead appraisals for organizations aiming to attain high maturity levels. Therefore, this knowledge will allow SPEAKER works as an assistant, allowing the practitioner to think about aspects that usually would be overlooked.

Table 1. Activities and Tasks for Performance Analysis

<table>
<thead>
<tr>
<th>Activities</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparing for Performance Analysis</td>
<td>- Identify quantitative goals of process quality and performance</td>
</tr>
<tr>
<td></td>
<td>- Identify critical processes</td>
</tr>
<tr>
<td></td>
<td>- Assess the adequacy of the process for performance analysis</td>
</tr>
<tr>
<td>Verifying Process Stability</td>
<td>- Group data of similar projects</td>
</tr>
<tr>
<td></td>
<td>- Select appropriate typ of control chart</td>
</tr>
<tr>
<td></td>
<td>- Construct the control chart</td>
</tr>
<tr>
<td></td>
<td>- Apply stability and trends tests</td>
</tr>
<tr>
<td></td>
<td>- Identify and execute corrective actions to stabilize the process (if necessary)</td>
</tr>
<tr>
<td></td>
<td>- Confirm process stability (if necessary)</td>
</tr>
<tr>
<td></td>
<td>- Establish performance baseline</td>
</tr>
<tr>
<td>Verifying Process Capacity</td>
<td>- Determine process capacity</td>
</tr>
<tr>
<td></td>
<td>- Compare capacity results with quantitative goals of process quality and performance</td>
</tr>
<tr>
<td></td>
<td>- Identify and execute corrective actions to make the process capable (if necessary)</td>
</tr>
<tr>
<td>Establishing Performance Model</td>
<td>- Identify possible independent variables</td>
</tr>
<tr>
<td></td>
<td>- Select appropriate analysis method according to variable types</td>
</tr>
<tr>
<td></td>
<td>- Develop the regression equation, probabilistic model, or simulation</td>
</tr>
<tr>
<td></td>
<td>- Calibrate and test the performance model</td>
</tr>
<tr>
<td>Monitoring Process Stability</td>
<td>- Collect new measures</td>
</tr>
<tr>
<td></td>
<td>- Update control chart</td>
</tr>
<tr>
<td></td>
<td>- Check need to recalculate performance baseline</td>
</tr>
<tr>
<td></td>
<td>- Apply stability and trends tests</td>
</tr>
<tr>
<td></td>
<td>- Confirm process stability</td>
</tr>
<tr>
<td>Monitoring Process Capacity</td>
<td>- Monitoring Process Stability (activity)</td>
</tr>
<tr>
<td></td>
<td>- Verifying Process Capacity (activity)</td>
</tr>
</tbody>
</table>

B. Knowledge documentation

Besides identifying which activities and tasks must be supported and which knowledge must be comprised in the body of knowledge, it is necessary to identify the format this knowledge must be organized and displayed to practitioners.

The way this knowledge is documented and presented influences the ease of its understanding and use. Therefore, the knowledge must be documented in a format that is suitable for practitioners. In this sense, SPEAKER must allow different ways of documenting and presenting the knowledge related to performance analysis activities, as follows:

- **Process**: The knowledge about activities and tasks related to process performance analysis should be documented in the form of a process, so that they can be described with their goals, input and output criteria, their responsible members etc. Besides, the sequence and dependencies between activities must also be evidenced.

- **Recommendations**: The suggested techniques for supporting the execution of each performance analysis activity should be presented through descriptions and examples of use. Tips and points of attention related to the execution of those activities must be presented as well.

- **Rules**: A result of a previous activity can affect or limit the options to execute the next activity. This knowledge is presented through rules.

- **Best practices**: Describe the consensus between experts in executing certain aspects of process performance analysis.

- **Lessons learned**: After executing performance analysis activities in the organizational context, it should be possible for the practitioner to describe lessons learned about this execution. This way, the initial knowledge base of the environment can be constantly fed back.

IV. Example of Use

Aiming to present how the body of knowledge can help a software development organization in executing performance analysis, a fictitious scenario of use is given. It was derived from professional experience of the authors and findings from the literature. Due to space limits, only the activity “Preparing for Performance Analysis” (shown in Table 1) is described.

The Alpha organization develops corrective and evolutive improvements of its products. It has achieved CMMI level 3, and the top management decided to start the implementation of level 4. As the organization does not have a consultant for supporting the necessary improvements, it was decided that Alpha’s software engineering process group (SEPG) should study the necessary concepts for applying the performance analysis and implement the process. For supporting the SEPG, the body of knowledge (to be provided by SPEAKER) is used.

The activity “Preparing for Performance Analysis” aims to identify which critical subprocesses (and their attributes) must be subject to performance analysis. The body of knowledge must then comprise all the tasks that must be executed to this end. Such tasks are listed in Table 2. The necessary knowledge to perform each task (whose samples are also described in
Table 2) provides some definitions, a description of applicable techniques, some examples and the lessons learned related to the task at hand. Further information on a concept or a specific technique will also be provided for helping its understanding.

Table 2. Tasks and related knowledge

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Corresponding Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define/clarify the organization’s vision and mission</td>
<td>Definition and examples of vision and mission</td>
</tr>
<tr>
<td>Identify strategic goals and their indicators</td>
<td>Balanced Scorecard (description and example) Goal-Question-Metric (GQM) (description and example)</td>
</tr>
<tr>
<td>Analyze external environment (market, clients, competitors etc.)</td>
<td>SWOT analysis (description and example) Porter Forces (description and example)</td>
</tr>
<tr>
<td>Analyze internal environment (organization’s strengths and weaknesses)</td>
<td>SWOT analysis (description and example)</td>
</tr>
<tr>
<td>Identify medium term tactical software goals</td>
<td>Definition and example of medium term tactical goals</td>
</tr>
<tr>
<td>Identify short term tactical software goals</td>
<td>Definition and example of short term tactical goals</td>
</tr>
<tr>
<td>Identify quantitative goals of process quality and performance</td>
<td>Definition and example of quantitative goals of quality and performance</td>
</tr>
<tr>
<td>Identify critical processes</td>
<td>Examples of selection criteria Recommendations from maturity models</td>
</tr>
<tr>
<td>Establish selection criteria for subprocesses</td>
<td>Examples of selection criteria Recommendations from maturity models</td>
</tr>
<tr>
<td>Evaluate subprocesses regarding the degree of compliance with the criteria</td>
<td>Weighted Multi-voting (description and example) Pareto Analysis (description and example)</td>
</tr>
<tr>
<td>Prioritize critical subprocesses</td>
<td>-</td>
</tr>
<tr>
<td>Assess the adequacy of the process for performance analysis</td>
<td>-</td>
</tr>
<tr>
<td>Identify attributes related to each critical subprocess</td>
<td>-</td>
</tr>
<tr>
<td>Evaluate the suitability of each attribute to performance analysis</td>
<td>Requirements needed for performance analysis</td>
</tr>
<tr>
<td>Identify and execute corrective actions for adjusting measures to performance analysis (if applicable)</td>
<td>Recommendations of corrective actions Results of corrective actions previously adopted by the organization</td>
</tr>
<tr>
<td>Define list of processes for performance analysis, with their related attributes and priorities</td>
<td>-</td>
</tr>
</tbody>
</table>

From the information provided through the body of knowledge, the SEPG establishes along with top management the following organization’s vision and mission.

Mission: “Develop high quality software in the medical field, providing clients with more agility in their processes, and providing resources for a more effective decision making. This way, we hope that our clients can have a high productivity in their consultations and value the preservation of their patients’ health with confidence”.

Vision: “Be recognized as a company of excellence in the development of software in all sectors of the medical domain, reaching a continuous and sustainable growth”.

Proceeding with the tasks set out by the body of knowledge, the SEPG identifies strategic and tactical goals (both short and long term) from successive decompositions. An example of these goals is shown in Figure 1 (for illustration purposes, only one requirement is derived at each step). For decomposing the short term software goal in quantitative goals of quality and performance, the SEPG needs to identify which subprocesses are related to this goal, as recommended in the body of knowledge. In the example shown, the subprocesses Software Coding and Software Testing are identified.

![Figure 1. Alpha’s Goals](image)

In order to assist the organization in properly defining quantitative goals, the information necessary for their completion are presented with an example indicating how they should be filled out. The quantitative goals of quality and performance identified by Alpha are presented in Table 3.

Table 3. Quality and performance quantitative goals for the Alpha organization

<table>
<thead>
<tr>
<th>Subprocess</th>
<th>Action</th>
<th>Value</th>
<th>Measurable Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Coding</td>
<td>decrease</td>
<td>10%</td>
<td>rework effort</td>
</tr>
<tr>
<td>Software Testing</td>
<td>decrease</td>
<td>5%</td>
<td>rework effort</td>
</tr>
</tbody>
</table>

From the information provided, the SEPG establishes a set of criteria to be adopted, and chooses to use the Weighted Multi-voting technique [18]. After applying the technique, the subprocesses Software Coding, Software Testing and Requirements Specification are identified as critical. In order to assess whether the critical subprocesses and their attributes are suitable for performance analysis, a checklist is presented for evaluating their attributes. Table 4 presents the application of the checklist to the subprocess Software Coding by the SEPG.

Table 4. Results from the evaluation of the subprocess Software Coding

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Result</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding effort</td>
<td>Suitable</td>
<td>N/A</td>
</tr>
<tr>
<td>Rework effort</td>
<td>Suitable</td>
<td>N/A</td>
</tr>
<tr>
<td>Schedule</td>
<td>Not suitable</td>
<td>Data are collected in a non-standard way</td>
</tr>
</tbody>
</table>

From the results, the SEPG cannot execute performance analysis with the attribute “schedule”, but can establish corrective actions, so that such attribute can be properly collected (hence properly analyzed) in future projects. The body of knowledge comprises recommendations that can guide the SEPG in this task. Examples include (i) a proper training
focused on the responsible for the data collection and (ii) the
definition of clearer collection procedures.

Thus, at the end of the activity “Preparing for Performance
Analysis”, Alpha has a list with critical subprocesses identified
and prioritized. From this list, the organization can begin
performance analysis with the verifications of stability of the
first subprocess (as shown in Table 1), with regards to an
attribute considered a priority.

V. TOWARDS A KNOWLEDGE-BASED ENVIRONMENT FOR
SOFTWARE PROCESS PERFORMANCE ANALYSIS

Aiming to minimize the difficulties faced by software
organizations for implementing process performance analysis
(as pointed in the previous sections), we are developing
SPEAKER, a knowledge-based environment for supporting the
execution of software process performance analysis. By using
this environment, we expect to help software organizations in
analyzing their process properly, taking into account the
organizational context and correctly applying the concepts and
techniques of process performance analysis. Through
SPEAKER, the presented body of knowledge must be accessed
and maintained effectively by practitioners.

The main module of SPEAKER is the Knowledge-Based
System (KBS), which aims to provide the knowledge about
performance analysis and perform the knowledge management
activities. Such activities involve the capturing, storage,
recovery, and maintenance of the knowledge and rules
described in the body of knowledge. In the context of the KBS,
the main knowledge management activities will be performed
as follows:

- **Knowledge identification and acquisition**: This activity
  involves both explicit and tacit knowledge, and also covers
  the knowledge identification, when the required knowledge
to execute the process performance analysis is identified.
  For the construction of the initial knowledge base of the
  environment, this activity has been performed through
  successive technical literature reviews and interviews with
  experts. After the environment is set up and used by an
  organization, the practitioners will be responsible for this
  activity by registering their understanding of performance
  analysis execution, or as a result of the knowledge
  management activity;

- **Knowledge organization and storage**: The knowledge is
  organized and stored so that it can be available and
  understandable by practitioners. To this end, different
  knowledge formats will be adopted depending on the
  performance analysis activity being executed at the time.
  These formats include lessons learned, best practices, case
  studies, examples and stories;

- **Knowledge retrieval and utilization**: This activity allows
  the knowledge to be used by practitioners to support them
  when a performance analysis activity is executed. The
  knowledge related to this activity is displayed to better
  assist its execution. Thus, only the knowledge that is
  relevant at the time is presented to the users, in order to
  avoid information overloading. Besides, one can search for
  a particular knowledge that is not being presented;

- **Knowledge maintenance**: While the knowledge is used to
  execute performance analysis, users can evaluate the
  usefulness and accuracy of such knowledge. Thus,
  knowledge with low scores can be reviewed or excluded
  from the knowledge base. This evaluation may also
  indicate the lack of a specific knowledge; in such case, a
  new round of the knowledge identification and acquisition
  activity can be conducted.

SPEAKER also consists of two other modules (not detailed
in this paper): a process components library and a tool for
process instantiation and execution. They are responsible for
defining and storing the execution history of the process
performance analysis and establishing the communication
between the KBS and the user.

VI. RELATED WORKS

In order to explore how knowledge management practices
are used in software process performance analysis, a systematic
mapping study was conducted. Such study, also known as
Scoping Study, is designed to provide a wide overview of a
research area [19]. The main goal of this study was to identify
papers that (i) report difficulties, problems or challenges
regarding knowledge issues in the process performance
analysis, and (ii) suggest or adopt knowledge management
methods or techniques to deal with these issues. The planning
and results of this study are described in [20].

Through this systematic mapping study, only one approach
that deals with some knowledge management practices to
support process performance analysis was identified. This
approach, called “SPC-Framework”, proposes to adapt the
original statistical process control concepts for application in
the software development scenario [21].

To achieve this purpose, this framework comprises [21]: (i)
Test set: the selection of a test set, amongst those presented in
the SPC literature, along with its rearrangement in logical
classes; (ii) Test interpretation: the interpretation of every test
in each class, from the software process point-of-view; (iii)
Investigation process: a process to guide software process
monitoring and stability investigation; and (iv) Experience
Base: a framework of an experience base for collecting
experience in the use of the SPC, based on Decision Tables.

Based on the few results of the study, non-structured
reviews in the literature were made to better ground the
proposed system. These reviews took into account areas non-
related to software, and some works were identified, especially
in the manufacturing area. One of them is an expert system [22]
which aims to support pattern recognition in a control chart.
According to the authors, this system allows plotting control
charts, identifying instabilities patterns, calculating the
capability index, identifying possible root causes, and
suggesting improvement actions.

Another work in the manufacturing area is an advisory
system aimed at capturing and disseminating knowledge to
support the selection of an adequate control chart in a specific
situation [23]. This system is based on pre-set rules that drive
the user to make the selection.
The few identified ones have some significant drawbacks that can harm the execution of process performance analysis, namely: (i) they generally consider only one type of control chart; (ii) they do not deal with the whole performance analysis process, but only some tasks; and (iii) they do not take into account the context information during the analysis.

VII. CONCLUSIONS

This paper presented how a body of knowledge on process performance analysis was organized. By integrating it into SPEAKER, software development organizations can be properly supported in carrying out performance analysis.

After the complete organization and structuring of the body of knowledge, the next research steps are the development of SPEAKER, based on the infrastructure provided by CORE-KM (Customizable Organizational Resources Environment with Knowledge Management) [24], a customizable environment that supports the definition, customization and execution of a knowledge management system that is specific for each organization. This environment was developed by our research group at COPPE and was already used in academic contexts and in Brazilian organizations for creating knowledge-based environments that support their business processes.

A survey was conducted with some experts in software process performance analysis, aiming to verify the activities and tasks identified as required for executing performance analysis. 4 out of 10 participants answered the survey. All of them stated that the proposed activities are indeed necessary. They also agreed with the sequence of these activities. However, according to them, the representation of iterations between activities should be emphasized, and the verification of the process capability should be better explained. One respondent indicated that statistical tools should be recommended as well. Some of these improvements are already being incorporated into the body of knowledge.

In addition to the evaluation of the knowledge identified and organized in the system, the SPEAKER environment will be evaluated as a whole. To this end, it is planned to use SPEAKER in the context of a real software development organization that is preparing to adopt performance analysis in their processes. The evaluation will include aspects such as (i) the sufficiency of the knowledge provided by SPEAKER when supporting non-experts in executing process performance analysis, and (ii) the adequacy of the knowledge representation format with respect to the stakeholders’ needs.

ACKNOWLEDGMENT

The authors would like to thank CNPq for the financial support to this research.

REFERENCES


