

Enterprise-Oriented Software Development Environments to Support Software Products and Processes Quality Improvement

Mariano Montoni, Gleison Santos, Karina Villela,
Ana Regina Rocha, Guilherme H. Travassos, Sávio Figueiredo,
Sômulo Mafra, Adriano Albuquerque,
and Paula Mian

Federal University of Rio de Janeiro - COPPE Sistemas
Caixa Postal 68511 – CEP 21941-972– Rio de Janeiro, Brazil
{mmontoni, gleison, darocha, ght}@cos.ufrj.br

Abstract. Software organizations have to adapt efficiently to cope with clients needs changes and new and evolving technologies in order to guarantee business success. Moreover, organizations must continuously enhance their capability to develop software in order to increase products and processes quality. These characteristics constitute dynamic environments that require specific competences from software engineers such as knowledge related to software technologies, ability to adapt software processes concerning project characteristics, and experience on product and process quality management. This paper presents enterprise-oriented software development environments that support software engineers to execute software processes more effectively and to produce products with better quality. A main feature of these environments is the support offered to organizational knowledge management. Thus the paper also presents the main characteristics of the knowledge management infrastructure integrated to those environments. The practical experience using the environments has shown several benefits, such as an increase of product and process quality, and the preservation of organizational knowledge related to software processes and the development of software products.

1 Introduction

Software organizations have to adapt efficiently to cope with clients needs changes and new and evolving technologies in order to guarantee business success. Moreover, organizations must continuously enhance their capability to develop software in order to improve their innovation ability and increase organizational processes efficiency [1]. In order to achieve this goal, organizations must be more productive, increase the quality of software products, diminish project effort and costs, and deal with the criticality of time-to-market for commercial products [2].

These characteristics constitute dynamic environments that require specific competences from software engineers such as knowledge related to software technologies,

ability to adapt software processes concerning project characteristics, and experience on product and process quality management. Globalization, increasing competition, more dynamic markets and shorter cycles in product development and innovation increase the need for a better adaptation to those environmental factors. These factors establish the need for a consequent adaptation of all business processes, including software processes, to existing and future market needs [1].

Software processes are knowledge-intensive business processes, i.e., a process that relies very much on knowledge [4]. Therefore, software processes can be considered as core processes of the organization, which executions should produce or add new knowledge to the organization's knowledge repository. Moreover, software processes executions require from software engineers innovative and creative abilities [3, 4].

Since participants of software processes executions have different background knowledge and past-experiences, different domains knowledge are applied at different levels [1]. Besides that, knowledge and software processes are integrated and should be evaluated as a whole, i.e., for knowledge management to be of use in an organization, it should be effortlessly incorporated in everyday business activities. Large parts of an organization's activities, especially on the operational level, are structured around business processes [1]. Therefore, it is imperative to integrate knowledge management with software processes, which means that a computerized system that supports business processes should also support knowledge management [5, 13].

In this context, Software Development Environments (SDE) have been playing an important role to support software engineers in the execution of software processes through the application of specific procedures that combine integrated tools and techniques in accordance to particular software paradigms. Moreover, SDE are evolving to integrate knowledge management activities within software processes aiming to support developers to produce better software products based on organizational knowledge and previous experiences more effectively [6].

This paper presents *Taba Workstation* SDEs. The *Taba Workstation* was created from the perception that different domain applications have distinct characteristics that influence in the environment from which software engineers develop software [7]. It is being used to support the deployment of software processes in small and medium size Brazilian companies as part of a larger project, named *QualiSoft*. The objective of this project is to increase the capability of software organizations and the quality of their software products through the adequate use of software engineering techniques in their software processes.

The next section presents some basic concepts related to SDE. The *Taba Workstation* infrastructure is presented in section 3. The steps executed to support software processes deployment in small and medium size companies in the context of the *QualiSoft* project are presented in section 4. The section 5 presents the *Taba Workstation* Knowledge Management tools. The section 6 presents some practical results of organizations that have been using the *Taba Workstation* SDEs. Finally, section 7 presents some conclusions and points out future work.

2 Supporting Software Products and Processes Quality Improvement Through Software Development Environments and Knowledge Management

A Software Development Environment (SDE) is defined as a computational system that supports software development, maintenance and improvements. It is supposed to support individual and group activities, project management activities, enhancement of software products quality, and increase of the productivity, providing the means for the software engineers to control the project and measure the activities evolution based on information gathered across the development. SDE should also provide the infrastructure to the development and integration of tools to support the execution of software processes. Moreover, this infrastructure should maintain a repository containing software project information gathered across its life cycle.

Recent researches have demonstrated the necessity of standardization of software development methodologies in an organization aiming to increase the control and improve the quality of products produced through the execution of software development processes [8, 9]. Therefore, SDE should not only support software engineers in the execution of software processes activities, but also provide the means to execute these processes according to organizational software development standards.

The standardization of software processes can be achieved through the definition of a Standard Process, i.e., a basic process that guides the establishment of a common process across the organization [8]. This process is the base for the definition of software development processes specialized according to the software type (paradigm and technologies), and to the development characteristics (e.g., distributed or centralized development). These specialized processes can be used to define specific processes considering particularities of the software project, a.k.a., instantiated software processes. Therefore, a SDE should be developed to guide software engineers in the execution of software processes instantiated to a specific software project.

Although it is recognized the importance of standardization of software processes, software organizations must cope with many difficulties that jeopardizes the institutionalization of standard processes, for instance, the dynamism of software processes, the necessity to apply new and evolving technologies, and the people turnover. Therefore, it is imperative for the organizations to manage their members' knowledge in an efficient way in order to guarantee the improvement of their software products and processes, and to preserve organizational knowledge [10, 11].

The identification, maintenance and dissemination of different types of knowledge related to software processes from one project to another are important to increase organization competitive advantages and improve software processes and products quality [12]. Efficient management of such knowledge supports organizational learning and initiatives for software process improvement [14].

The fact that most software development organizations are process-centered provides many benefits (e.g., process-centered knowledge management systems can be designed to associate software process activities with knowledge necessary to execute

it explicitly) [14]. Moreover, tacit and explicit members' knowledge related to software processes are valuable individual assets that must be captured and converted into the organizational level. The collected knowledge represents indicators of problems concerning the software process definition or the environment in which the software is being developed. This important knowledge can be used to learn about the software process and to provide the means for implementing organizational changes aiming to enhance business performance [15]. In order to acquire such knowledge efficiently, it is necessary to transform arbitrary experiences declarations in structured explicit representations through the execution of activities for knowledge acquisition, packaging, dissemination and utilization [16].

Lindvall *et al.* [17] present some benefits of a knowledge management program institutionalization: (i) efficient reuse of documented experiences; (ii) easiness to find solutions for problems within the organization; (iii) identification and storage of valuable experiences; and (iv) facility to propose measures to improve processes execution and increase software products quality. Basili *et al.* [18] and Ruhe [19] point out that by structuring and explicitly representing software process knowledge, it is possible to define efficient training programs that can increase employees' productivity and foster transference of innovative software engineering technology. Landes [20] also notes that knowledge management solutions efficiently support activities of organization members with poor experience in a specific area or domain.

Although most organizations recognize the importance of managing software process knowledge, the establishment of a knowledge management program is sometimes a laborious task. For instance, it is hard to convert tacit knowledge to explicit, and it is difficult to implement knowledge management solutions in a non-intrusive way. Weber *et al.* [21] point out problems with knowledge management systems, for instance, inadequacy of knowledge representation formats and lack of incorporation of knowledge management systems into the processes they are intended to support.

3 The Taba Workstation

The **Taba Workstation** was created from the perception that different domain applications have distinct characteristics that influence in the environment from which software engineers develop software [7]. During the last years, the **Taba Workstation** evolved to comply with the different levels of capability maturity models of software organizations and to support knowledge management activities integrated to software processes. Therefore, the main objectives of **Taba Workstation** are: (i) to support the configuration of process-centered software development environments for different organizations (Configured SDE); (ii) to support the automatic generation (i.e., instantiation) of software development environments for specific projects (Enterprise-Oriented SDE); (iii) to support software development using the instantiated environment; and (iv) to support the management of organizational knowledge related to software processes. Compared to similar SDEs, the fact that the **Taba Workstation**

was developed by the research group allowed the freely distribution of the system to software development organizations.

The CASE tools integrated in the environments offer automated support to: (i) adaptation of the organization standard processes for a specific project; (ii) definition of the organizational structure [6]; (iii) acquisition, filtering, packaging and dissemination of organizational knowledge [6]; (iv) planning the organization of specific projects; (v) time, costs, risks [22], human resources planning, monitoring and control [6]; (vi) planning and execution of Configuration Management activities; (vii) identification of software product quality requirements; (viii) documentation planning; (ix) supporting the planning and monitoring of corrective actions; (x) supporting measurement and analysis activities based on the GQM method; (xi) project monitoring through the generation of periodic reports and measures; (xii) controlling of the activities executed during a specific project; (xiii) requirements management; and (xiv) post mortem analysis.

One of the greatest restrictions for knowledge sharing is the use of different concepts to describe a domain for different systems. The development of ontologies facilitates the sharing of a common terminology. The use of ontologies in the *Taba Workstation* infrastructure facilitates the communication between multiple users and the retrieval of knowledge stored in the environment. Considering communication, the defined ontologies reduces terminological and conceptual mismatch in a company. When retrieving knowledge items, the ontologies' supplies vocabularies whose terms are used as links among multiple knowledge/data bases contents. Moreover, when defining synonyms and acronyms for the concepts, ontologies provide linguistic equivalents, which may occur in text documents and can be used to classify and access non-formal knowledge.

The Software Engineering Ontology defines a common vocabulary to guide the registration/distribution of a company's knowledge map and software engineering knowledge in the *Taba Workstation*. A company's knowledge map defines for each employee its level of skills, knowledge and experiences.

The Enterprise Ontology provides concepts and attributes related to the structure, behavior and knowledge owned by companies, defining a common vocabulary to guide the description of any company. The Enterprise Ontology aims to supply a common vocabulary that can be used to represent useful knowledge for the software developers on the involved organizations in a software project. It supports the development of several CASE tools in the *Taba Workstation* SDEs [6].

The ontology explicit definition supported the identification of potentially useful terms and phrases, definition of ontology semantics and sub-ontologies (depicted in figure 1) that describe concepts aiming to facilitate understanding, for example:

- how the organization is perceived in its environment,
- how the organization is structured, which are their objectives and how it behaves,
- how organization's projects have been lead and how the desired and possessed abilities have been distributed into the organization,
- who are the available resources on the organization and how the distribution of authority and responsibility in the organization are accomplished.

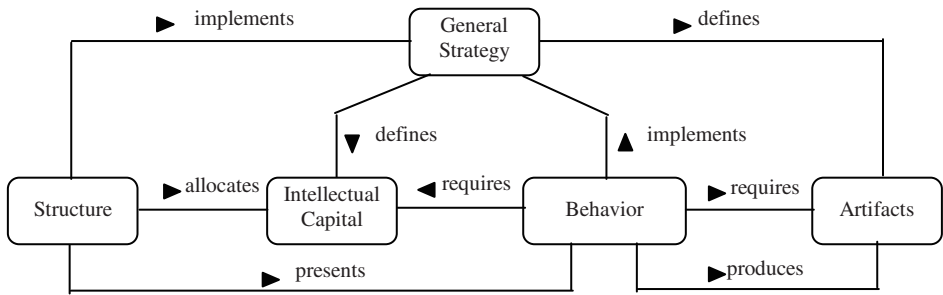


Fig. 1. Sub-ontologies of Enterprise Ontology

4 Using TABA Configured Environments to Support Software Processes Deployment in Small and Medium Size Companies

Since 2003, software engineers of several small and medium size Brazilian organizations are using the *Taba Workstation* in the context of the *QualiSoft* project to increase their capability to develop software through the adequate use of Software Engineering methods and techniques in its software processes. The *QualiSoft* project resulted from a contract between the *RioSoft* (a non-governmental organization that integrates the *Softex Program - Society for the Support of Brazilian Software Production and Exportation*) and the *Federal University of Rio de Janeiro*.

The *QualiSoft* project was constituted of the following activities: (i) definition of standard software development and maintenance processes adjusted for small and medium size companies; (ii) training in Software Engineering methods and techniques and in the software processes defined; (iii) adaptation of the standard process to each company; (iv) use of a *Taba Configured SDE* to support the use of the processes; and (v) follow-up of the companies in the execution of pilot projects.

Since the focus of the *QualiSoft* project is on small and medium size organizations, we executed the project with a pool of organizations with similar characteristics aiming to decrease the overall cost and increase the project feasibility. The first phase of the project started on August 2003 and aimed to address a pool of 10 organizations. The second phase started on January 2004 and addressed a second pool of 9 organizations.

The next sections describe the following steps executed to achieve the objective of the *QualiSoft* project: (i) definition of software process in accordance to specific characteristics of software development and strategic goals of each organization and training using the *Taba Workstation SDEs*; and (ii) deployment of the defined software processes using a *Taba Configured SDE*.

4.1 Software Process Definition and Training Using the *Taba Workstation*

The first step in the execution of the *QualiSoft* project was to be acquainted of the individual characteristics of the organizations. In order to do so, each organization filled

out a detailed form and the process specialists had to schedule regular visits on the organizations. The form contained questions related to the organizational culture, software process stages and quality management systems adopted software development practices, main problems in the current software development and maintenance processes, and organizational objectives related to software process improvement.

The following step was to define software development and maintenance standard processes adequate to small and medium size organizations. The processes defined is based on the software processes life cycle described in the international standard ISO/IEC 12207 [9] and it is adherent to the software development practices defined in the CMMI [23] Level 2 process areas.

In parallel to the processes definition activity, the members of the organizations were trained in the Software Engineering methods and techniques. During the first year, approximately 32 hours were spent on formal training. This training was performed under the form of tutorials on the following topics: Software Engineering, Software Process, Requirements Engineering, Configuration Management, Project Management and Software Products Quality. Approximately, 80 professionals were trained during this phase. After the theoretical training, project managers and software developers participated on a specific training on the standard software processes defined. Training activities during the second year included other important topics, such as Peer-review, Software Tests, Measurement and Analysis, Supplier Agreement and Knowledge Management, constituting 44 hours of formal training. During the second year of the project, more than 70 organizational members were trained.

The following step focused on the processes deployment supported by the configured environment in each organization.

4.2 Software Processes Deployment Using a Tabca Configured SDE

These steps had been carried out individually in each organization considering its specific characteristics. Initially, the standard processes defined previously had been adapted by the software process specialist to each company considering the characteristics identified in the beginning of the project, such as software types developed, documents produced and software development paradigms adopted. After the approval of the adaptations by the organization, a software development environment was configured based on the adapted processes (the organization's standard process).

The configured environment was installed in each organization followed by 20 hours of hands-on training on the software tools. After this training, a pilot project was carried out. Using their configured environment each organization instantiated a specific environment for its pilot project through the execution of the AdaptPro tool.

The objective of the AdaptPro tool is to support the institutionalization of the standard processes because it facilitates the adoption of these processes in all the projects of the organization. By using the AdaptPro tool, the software engineering can execute the following activities: (i) characterize the project; (ii) plan the process that will guide the project through the adaptation of the organizational standard process con-

sidering the project characteristics; and (iii) instantiate a SDE to support the execution of the planned process. The figure 2 presents a screenshot of the AdaptPro tool. On the left side of this figure, the system presents the activities that guide the execution of the tool. On the right side of the figure, the system presents another screen to support the execution of the selected activity; in this case, it is presented the screen that supports the definition of a life cycle model to a specific project as part of the process planning activity. A list of life cycle models and the respective level of adequacy to the project considering its characteristics are presented on the right side of the screen. Besides that, the user can consult the justification of the automatic identification of the adequacy level and can consult the software processes defined for similar projects that used the same specialized process and life cycle model facilitating the selection of an adequate project life cycle model by the user. Moreover, the user can consult knowledge related to life cycle models directly from this screen and register knowledge related to the planning process activity, such as lessons learned. These functionalities are described in the following section.

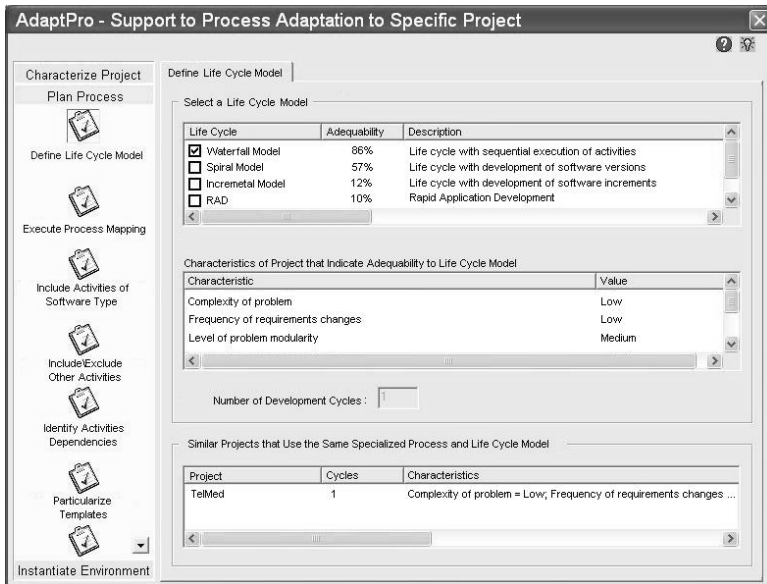


Fig. 2. AdaptPro – a tool to support process adaptation to specific projects

After planning the process, the project manager use the AdaptPro tool to instantiate the specific process to the project based on its particularities. The product of this tool is the process plan (including adaptations to support the life cycle model chosen) and a SDE to support the execution of the planned process.

The AdaptPro tool, just like the other tools of the *Taba Workstation*, is integrated to the *Taba Workstation* Knowledge Management tools. These tools are described in the following section.

5 Taba Workstation Knowledge Management Tools

The Taba Workstation is constituted of several tools integrated to Knowledge Management tools that support software engineers in the execution of software processes activities in different ways. The next sections describe the *Sapiens* and the *ACKNOWLEDGE* Knowledge Management tools.

5.1 *Sapiens*: A Corporate Yellow Page Tool to Support the Representation of Organizational Structure Knowledge

The analysis of corporate yellow pages has a great importance to human resources selection. Because usually most of the time the desired competence exists somewhere inside the organization, being, however, necessary to expend much time to identify, find and have access to who possesses it [18]. *Sapiens* is a software tool for the representation of the organizational structure with the competences required along it and is integrated to the Taba Workstation tools [6]. Besides supporting staff allocation, including the competences of each professional, it also contains search and navigation mechanisms. This way, it is possible the creation of a culture of identification, acquisition and dissemination of the existing knowledge that can be used by the organization to know itself better and take off greater advantage of its potential. It is based on the infrastructure defined for the Taba Workstation, making use of the Enterprise Ontology. Software developers can use it to find the most appropriate person to help in the solution of a problem inside the organization.

For each position in the organizational structure, it is possible to indicate which competences are necessary or relevant for its performance and to indicate which of these competences are mandatory or not. In a similar way, it is possible to indicate which competences a person owns. The association between people and competence, as well as between position and competence, is not always equal and must take in consideration a certain level of variation. This leveling of the competences allows the standardization of the different degrees of “expertise” existing for a specific ability. Predefined search options based on the ontology structure (e. g., “Who has a specific competence?”, “Which are the competences for a person?”) are available and the user can also create new ones.

The organizational structure knowledge represented in the *Sapiens* tool is used by other tools available in the environments instantiated to support the execution of processes adapted to specific projects, for instance, the RHPlan tool. The goal of the RHPlan tool is to support the human resources allocation in a software project. It also has mechanisms to help the contract order or qualification of professionals order when the necessary human resources cannot be found inside the organization. It is based on the definition of the necessary competence profiles to the accomplishment of project activities, and posterior search for organization professional that possesses similar profiles to the desired one. The project manager can search the knowledge on the existing competence inside the organization and find who possesses them. The database of professional’s capabilities is provided by the *Sapiens* tool.

Figure 3 presents a screenshot of RHPlan tool, showing an example of human resources allocation in project activities. In the left side is possible to see all activities for the staff allocation plan creation: definition of profiles needed in the execution of each process activity, selection of professionals, request of contract or training for professionals when the available professionals in the organization do not fit the desired profile, and visualization of the human resources allocation plan.

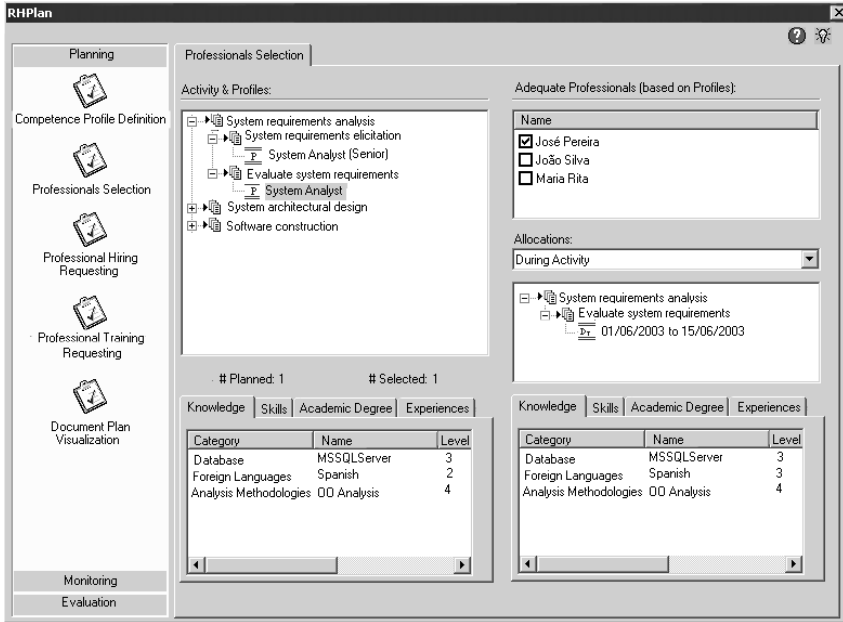


Fig. 3. Professional selection in RHPlan tool supported by organizational structure knowledge represented in the Sapiens tool

The software engineer not only can consult the knowledge competences inside the organization through the RHPlan tool, but also consult the analysis of past experiences supporting project managers in planning and controlling the human resources allocation. Project experiences are an important source of knowledge in Software Engineering and, therefore, the acquired experiences must be identified and shared [24]. Moreover, the (successful or not) lessons learned about human resources allocations in other projects are very important so previous errors can be avoided and success cases are not forgotten.

As well as Sapiens, RHPlan uses the Enterprise Ontology for its class model definition. However, while in Sapiens classes are related to staff allocation in the organizational structure, in RHPlan the allocation is carried through specific software development process activities. Both of them manipulate the same database of organization members' competences; then they are benefited by the same mapping infrastructure of ontology concepts to physical model classes. RHPlan also uses the

concepts defined in the sub-ontology of Behavior to describe projects, software processes, activities, resources and distribution of necessary competences for the accomplishment of the activities.

During the knowledge acquisition, the ontology concepts can also be used to index the organizational knowledge memory. Later, during the knowledge search this same information can be used to assist the search for a specific knowledge. This functionalities are describes in the following section.

5.2 ACKNOWLEDGE: A Knowledge Acquisition Tool

In order to acquire, filter and package organization members’ tacit and explicit knowledge related to software processes, an acquisition process was defined and a supporting tool named **ACKNOWLEDGE** was implemented and integrated into other tools in the *Taba Workstation* [6]. The main objective of this approach is to capture individual knowledge valuable for the organization, such as, domain knowledge (domain theory), business knowledge (best practices, knowledge about clients and new technologies), past experiences knowledge (lessons learned, common problems), and organization members’ knowledge acquired during processes execution.

The **ACKNOWLEDGE** tool can be accessed from two icons located under the title bar of all tools from the *Taba Configured SDE* (figure 2 and 3). Organization members’ knowledge can be acquired by clicking on the icon (📁) and all captured knowledge can be consulted by clicking on the icon (🔍). The integration of **ACKNOWLEDGE** to these tools avoids interruption of organization members normal routine during knowledge capture and reuse.

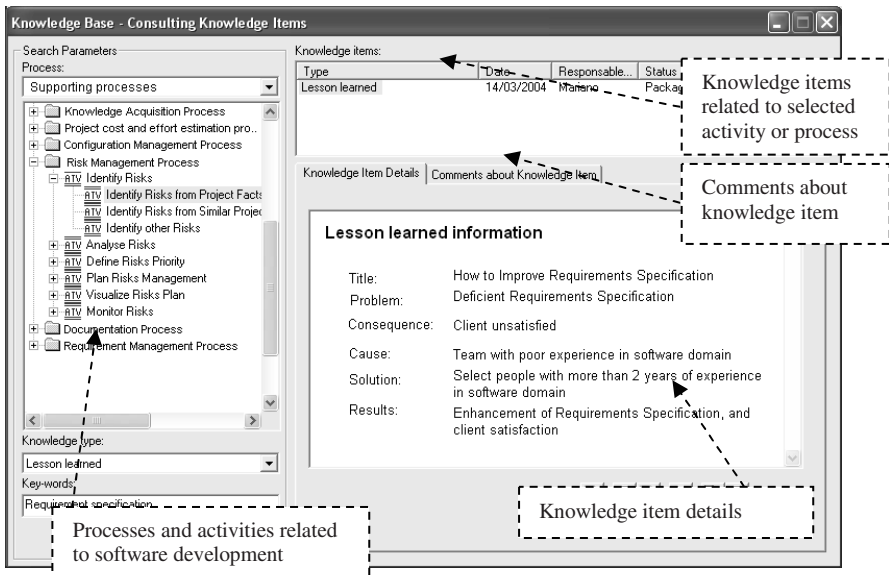


Fig. 4. Consulting interface of **ACKNOWLEDGE** tool

Figure 4 presents the knowledge consulting interface. In order to facilitate the search, some consulting parameters can be specified, such as, activity or process related to software development, knowledge type, and key words. The consulting interface also allows knowledge users to register comments about knowledge items. These comments are very useful during knowledge repository maintenance since they facilitate identification of knowledge items that have real value for the organization. Moreover, the comments add value to knowledge items turning them more reliable for knowledge users, and providing the means for establishment of a continuous learning cycle. During the knowledge acquisition, the ontology concepts can be used to index the organizational knowledge memory. Later, during the knowledge search this same information can be used to assist the search for a specific knowledge.

6 Success Cases of Software Processes Deployment Using *Taba Workstation*

The use of the *Taba Workstation* to support software processes deployment in small and medium size companies demonstrated several benefits in the context of the QualiSoft project described previously, such as increase of product and process quality, and preservation of organizational knowledge related to software processes. A direct benefit obtained from the use of the *Taba Workstation* can be exemplified by two organizations that obtained ISO 9000:2000 certification based on the software process and configured environment deployed.

The first organization to obtain the ISO certification was a software development organization that during two years was involved in software processes definition, preparation and implantation without success. After one year of the QualiSoft project, the organization obtained the ISO 9000:2000 certification. Nevertheless, the organization has obtained more practical benefits than just the ISO certification, for instance:

- decrease of rework,
- increase of software products quality,
- increase of software developers' qualification through the dissemination of software engineering knowledge,
- establishment of best practices aligned to organizational objectives,
- preservation of organizational knowledge, obtaining relative people independence during the projects. Although, the organization recognizes the importance of people, the client will not suffer from lack of quality of processes and products when people are reallocated to other projects.

The organization recognized that the *Taba Configured SDEs* facilitate and significantly reduce repetitive work during the definition of the specific process for a project, planning activities, project execution and monitoring. Moreover, the *Taba Workstation* lessons learned repository has constituted a repository of historical information to be used during other projects execution aiming to prevent the recurrence of common errors and problems.

The second organization to obtain the ISO certification is a software development organization that already had the ISO 9000:1994 certification and had to be compliant with the ISO 9000:2000 standards in order to renew their certificate. According to the software engineers of the organization, the *Taba Configured* SDE was decisive to obtain the certification renewal, because it has speeded the deployment of software processes and has facilitated the dissemination of organizational best practices.

Besides the report from these two companies, some lessons learned were identified during the regular meetings to support the process deployment in the organizations:

1. the deployment of software processes in a group of organizations in a common project and at low cost is feasible (the costs for each organization was about 40% less than a similar project conducted independently for one organization),
2. the configured environments facilitates training, deployment and institutionalization of software processes by providing case tools to assist and/or automate software development and maintenance tasks,
3. the knowledge management approach adopted in the environments is determinant for the success of the approach, because it helps the developer in executing its activities by providing useful knowledge when the developer needs it the most,
4. the knowledge acquisition approach integrated into the CASE tools enables the gradual evolution of the knowledge repository with acquisition and dissemination of lessons learned, best practices and suggestions for processes improvement,
5. the effective support of the high-level manager of each company was a critical factor to guarantee the success of the QualiSoft project.

7 Conclusions

This paper presented the *Taba Workstation* SDEs, enterprise-oriented software development environments that support software engineers to execute software processes more effectively and to produce products with better quality. The paper also presented the main characteristics of the knowledge management infrastructure integrated to those environments. These environments have shown to be useful for software engineers supporting the deployment and execution of software processes and the construction of software products with better quality. Therefore, the *Taba Workstation* SDEs guaranteed not only the increase of organizational competitive advantages, but also the enhancement of software processes and products quality.

The results of the QualiSoft project have also shown the feasibility of carrying out software process definition and deployment as a cooperative work between several companies reducing costs without diminishing the expected results. The project has also shown that it is possible to promote technology transfer between universities and software organization producing good results to all the involved parts.

The *Taba Configured* SDEs also support the execution of other Brazilian companies' software processes. The initial results regarding the use of these environments are promising. Three companies using *Taba Configured* SDEs are expected to reach CMMI Level 2 in official appraisals during 2005. A CMMI pre-evaluation was conducted successfully in one organization in January 2005, and the official appraisal is

schedule to March 2005. This organization has also recently obtained the ISO 9000 certification through the support of the *Taba Workstation*.

Further information about Enterprise-Oriented Software Development Environment and its tools can be found at <http://www.cos.ufrj.br/~Taba>.

Acknowledgement

The authors wish to thank CNPq and CAPES for the financial support granted to the project Enterprise-Oriented Software Development Environments and SEBRAE for supporting these two years of QualiSoft project. We also wish to thank K. Oliveira, R. Falbo, M. Amaral and B. Diaz for their contributions to this project.

References

1. Gronau, N., Müller, C., Uslar, M.: The KDML Knowledge Management Approach: Integrating Knowledge Conversions and Business Process Modeling, Proc. of the 5th Int. Conf. of Practical Aspects of Knowledge Management, Vienna, Austria, (2004) 1–10
2. Pfleeger, S. L.: Software Engineering: theory and practice, 2nd edition, Prentice-Hall, Inc., ISBN 0-13-029049-1 (2001)
3. Eppler, M., Seifried P., Röpnack, A.: Improving knowledge intensive Processes through an Enterprise Knowledge Medium. In: PRASAD, J. (ed.): Proceedings of the 1999 Conference on Managing Organizational Knowledge for Strategic Advantage: The Key Role of Information Technology and Personnel. New Orleans (1999), In ref: [1]
4. Hamel, G., Prahalad, C.K.: The core competence of the corporation. Harvard Business Review (1990), In ref: [1]
5. Andersson, B., Bider, I., Perjons, E.: Integration of Business Process Support with Knowledge Management – A Practical Perspective, Proc. of the 5th Int. Conf. of Practical Aspects of Knowledge Management, Vienna, Austria, (2004) 227–238
6. Montoni, M., Santos, G., Villela, K., Miranda, R., Rocha, A.R., Travassos, G.H., Figueiredo, S., Mafra, S.: Knowledge Management in an Enterprise-Oriented Software Development Environment, Proc. of the 5th Int. Conf of Practical Aspects of Knowledge Management, Vienna, Austria, (2004) 117–128
7. Oliveira, K. M., Zlot, F., Rocha, A. R., Travassos, G. H., Gallota, C., Menezes, C.: Domain-oriented software development environment, Journal of Systems and Software, Vol. 172, No.2 (2004) 145 - 161
8. Eman, K. E., Drouin, J., Melo, W.: SPICE – The Theory and Practice of Software Process Improvement and Capability Determination, IEEE Computer Society Press (1998)
9. ISO/IEC 12207:2000. Information technology – software process life cycle (2000)
10. Truex, D. P., Baskerville, R., Klein, H.: Growing Systems in Emergent Organizations, Communications of the ACM, Vol. 42, No. 8, Aug. (1999) 117-123
11. Maidantchik, C., Santos, G., Montoni, M. A.: Organizational Knowledge: an XML-based Approach to Support Knowledge Management in Distributed and Heterogeneous Environments, In: Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering SEKE'04, Banff, Canada, Jun (2004) 427-430
12. Houdek, F., Bunse, C.: Transferring Experience: A Practical Approach and its Application on Software Inspections, In: Proc. of SEKE Workshop on Learning Software Organizations, Kaiserslautern, Germany, Jun, (1999) 59-68

13. Holz H., Könnecker A., Maurer F.: Task-Specific Knowledge Management in a Process-Centered SEE, K.-D Althoff, R.L. Feldmann, and W. Müller (Eds): LSO, LNCS 2176, (2001) 163-177
14. Maurer, F., Holz, H.: Process-centered Knowledge Organization for Software Engineering, In: Papers of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning, Orlando, Florida, Jul: AAAI Press, (1999)
15. Decker, B., Althoff, K.-D, Nick, M., Tautz, C.: Integrating Business Process Descriptions and Lessons Learned with an Experience Factory, In: Professionelles Wissensmanagement - Erfahrungen und Visionen (Beiträge der 1. Konferenz für Professionelles Wissensmanagement), eds. Hans-Peter Schnurr, Steffen Staab, Rudi Studer, Gerd Stumme, York Sure. Baden-Baden, Germany. Shaker Verlag, Aachen, Mar (2001)
16. Birk, A., Tautz, C.: Knowledge Management of Software Engineering Lessons Learned, IESE-Report 002.98/E, Jan (1998)
17. Lindvall, M., Frey, M., Costa, P., Tesoriero, R.: Lessons Learned about Structuring and Describing Experience for Three Experience Bases, K.-D Althoff, R.L. Feldmann, and W. Müller (Eds): LSO, LNCS 2176, (2001) 106-118
18. Basili, V., Lindvall, M., Costa, P.: Implementing the Experience Factory concepts as a set of Experiences Bases, In: Proceedings of the Int. Conf. on Software Engineering and Knowledge Engineering, Buenos Aires, Argentina, Jun (2001) 102-109
19. Ruhe, G.: Experience Factory-based Professional Education and Training, In: Proc. of the 12th Conference on Software Engineering Education and Training, March, New Orleans, Louisiana, USA, (1999)
20. Landes, D., Schneider, K., Houdek, F.: Organizational Learning and Experience Documentation in Industrial Software Projects, Int. J. on Human-Computer Studies, Vol. 51, (1999) 646-661
21. Weber, R., Aha, D. W., Becerra-Fernandez, I.: Intelligent Lessons Learned Systems, International Journal of Expert Systems Research and Applications 20, No. 1 Jan (2001).
22. Farias, L., Travassos, G. H., Rocha, A. R. C.: Knowledge Management of Software Risks In: Journal of Universal Computer Science, Vol. 9, No 7 (2003), 670- 681
23. CMU/SEI, Capability Maturity Model Integration (CMMI) Version 1.1 - Staged Representation, Carnegie Mellon University, Software Engineering Institute, Pittsburgh (2002)
24. Markkula, M.: Knowledge Management in Software Engineering Projects, Software Engineering and Knowledge Engineering – SEKE, Kaiserlautern, Germany, Jun (1999)