Taba Workstation: Supporting Software Process Deployment Based on CMMI and MR-MPS.BR

Mariano Montoni, Gleison Santos, Ana Regina Rocha, Sávio Figueiredo, Reinaldo Cabral, Rafael Barcellos, Ahilton Barreto, Andréa Soares, Cristina Cerdeiral, and Peter Lupo

COPPE/UFRJ - Federal University of Rio de Janeiro
POBOX 68511 – ZIP 21945-970 – Rio de Janeiro, Brazil
{mmontoni, gleison, darocha}@cos.ufrj.br

Abstract. Deployment of software processes based on reference models is a knowledge-intensive task, i.e., a great amount of technical knowledge must be applied in order to guarantee conformance and adherence of processes deployed to the reference models adopted. Moreover, software process deployers have to deal with organizational and individual cultural problems on a regular basis, for instance, resistances to organizational changes. Therefore, the success of software process deployment within an organization or organizational unit depends on both technical and social aspects of the software process deployment strategy definition and execution. This paper presents the Taba Workstation, an enterprise-oriented Process-centered Software Engineering Environment (PSEE) constituted of an integrated set of tools to support software process deployment based on the Capability Maturity Model Integration (CMMI) and the Reference Model for Brazilian Software Process Improvement (MR-MPS.BR). Software process appraisals demonstrated that the Taba Workstation constitutes one of the most important organizational assets to facilitate the success of software process deployment initiatives and to overcome the inherent difficulties.

1 Introduction

Deployment of software processes based on reference models is a knowledge-intensive task, i.e., a great amount of technical knowledge must be applied in order to guarantee conformance and adherence of processes deployed to the reference models adopted. Moreover, software process deployers have to deal with organizational and individual cultural problems on a regular basis, for instance, resistances to organizational changes [1, 2]. Therefore, the success of software process deployment within an organization or organizational unit depends on both technical and social aspects of the software process deployment strategy definition and execution.

One important characteristic of a software process deployment initiative is the selection of an appropriate reference model to base the definition of the software processes and evaluation of the organization. International standards like ISO 12207 [3] and ISO 15504 [4] and software process quality models like CMMI (Capability Maturity Model Integration) [5] were developed aiming to define the requirements of an ideal organization, i.e., a reference model to be used in order to assess the maturity of

J. Münch and M. Vierimaa (Eds.): PROFES 2006, LNCS 4034, pp. 249 – 262, 2006.

the organization and their capability to develop software. Based on these standards and models, Brazilian industry and research institutions have worked together during the last two years aiming to define the Reference Model for Brazilian Software Process Improvement (MR-MPS.BR) [6, 8, 9]. This model has been deployed in many companies in Brazil and official appraisals were already conducted.

This paper presents the Taba Workstation, an enterprise-oriented Process-centered Software Engineering Environment (PSEE) constituted of an integrated set of tools to support software process deployment based on the Capability Maturity Model Integration (CMMI) and the Reference Model for Brazilian Software Process Improvement (MR-MPS.BR).

Section 2 presents the Reference Model for Brazilian Software Process Improvement and the appraisal method developed. Section 3 presents the main characteristics of PSEE approaches to support software process definition, deployment and enactment. Section 4 describes the main objectives of the Taba Workstation, and how it supports software process deployers during the deployment of processes according to reference models. Section 5 presents the conclusions and points out future directions for the presented work.

2 The Reference Model for Brazilian Software Process Improvement

The Reference Model for Brazilian Software Process Improvement (MR-MPS.BR) was created with the objective to provide an adequate model to Brazilian public and private organizations with different characteristics and sizes based on the most important reference models for software process definition and improvement (ISO/IEC 12207 [19], ISO/IEC 15504 [20], and CMMI [21]).

The reference standard for the software processes of MR-MPS.BR is the ISO/IEC 12207, i.e., this standard is the framework for the definition of the processes that constitute the MR-MPS.BR. Similarly to the ISO/IEC 12207 standard, the MR-MPS.BR defines fundamental processes, supporting processes and an adaptation process. Each company interested in deploying the MR-MPS.BR should select the pertinent processes from that set according to the adaptation process. The expected results for the deployment of the MR-MPS.BR processes are an adaptation of the expected results of the ISO/IEC 12207 processes and activities.

Seven maturity levels were established in the MR-MPS.BR: Level A (Optimization), Level B (Quantitatively Managed), Level C (Defined), Level D (Largely Defined), Level E (Partially Defined), Level F (Managed), and Level G (Partially Managed). For each of these maturity levels, processes were assigned based on the ISO/IEC 12207 standard and on the process areas of levels 2, 3, 4 and 5 of CMMI staged representation. This division has a different graduation of the CMMI staged representation aiming to enable a more gradual and adequate deployment in small and medium size Brazilian companies. The possibility of rating companies maturity considering more levels, not only diminishes the cost and effort of achieving a certain maturity level, but also allows the visibility of the results of the software process improvement within the company and across the country in a shorter time when compared to other models, such as CMMI. The criteria used to divide the processes across the maturity levels G-C were the importance of the process to the company, the facility to implement it and the dependency of the process to the others.

The MR-MPS.BR Appraisal Method for Process Improvement was defined based on the ISO/IEC 15504 standard. The level of deployment of the expected results related to a specific process is evaluated based on indicators that evidence such deployment. These indicators are defined for each company, related to the expected results of a process, and can be one of the following types: (i) Direct, (ii) Indirect, or (iii) Affirmations. Direct indicators are intermediate work products that result from an activity. Indirect indicators are generally documents that indicate that an activity was executed. Affirmations are results of interviews with the project teams of the evaluated projects. The implementation of an expected result is evaluated according to four levels: (i) TI – Totally Implemented; (ii) LI – Largely Implemented; (iii) PI – Partially Implemented, and (iv) NI – Not Implemented. The appraisal method adheres completely the ISO/IEC 15504 standard appraisal method defined to the staged representation.

A company is considered MR-MPS.BR level A, B, C, D, E, F or G if and only if all of its units, divisions or sectors had been rated as such level. Since one or more appraisals can be executed in a company, it is possible that parts of a company are rated with different levels. No matter the appraisal context, the evidential document of the appraisal must explicitly state the objective of the appraisal (appraisal scope), and the maturity level ratings.

3 PSEE Approaches to Support Software Process Definition, Deployment and Enactment

A great variety of PSEE approaches have been defined, designed and implemented over the past years. Many of these approaches have been developed to cope with the software engineering dynamic environments, such as software process evolution, decentralization of software process modeling and enactment, and support of cooperative activities. In the following, some of the most significant approaches to process model definition, deployment and enactment will be discussed.

EPOS (Expert System for Program and ("og") System Development) is a SEE (Software Engineering Environment) with emphasis on Process Modeling, Software Configuration Management and support to cooperative work [13]. EPOS supports a reflexive, object-oriented software process modeling language called SPELL. EPOS facilitates basic mechanisms for incremental (re)planning and enactment of the process models by process tools like Planner and Process Engine. An evolution of EPOS is EPOSDB built to store versioned software products, as well as their related process models. EPOS also supports cooperative transactions. EPOS is constituted of a meta-process for managing model evolution, and mechanisms for managing process evolution: retrieval of project experience, recording of project experience and manipulation of task-network layout. Although EPOS efficiently supports process modeling, evolution and enactment, there are not knowledge management mechanisms to provide knowledge to process executants during process enactment. Moreover, EPOS provides a meta-model only to software process domain. The connection between this domain and other areas of Software Engineering is not allowed.

Oz is a Process-centered Software Engineering Environment (PSEE) that implements the requirements for a decentralized PSEE based on a design for decentralization of process modeling and enactment [14]. The Oz environment supports process modeling using two popular families of Process Modeling Languages (PMLs), rules and Petri-Nets. Although the Oz environment implements a decentralized Process Centered Environment architecture, there is no integration of the process models to other software engineering tools. Moreover, the formalisms for process modeling sometimes become a burden for process modeler due to lack of intuition. This creates significant barriers to entry and, consequently, limits the possibility for the PMLs to be adopted in practice [2].

SPADE is a research project with the goal of developing an environment for Software Process Analysis, Design and Enactment [15]. SPADE is centered on a process modeling language called SLANG, an extension of a high-level Petri nets. SPADE-1 evolved to support cooperation in software development. Although SPADE-1 has demonstrated to be efficient to deal with synchronous and asynchronous activities among distributed, there is no evidence of applicability of such approach in the industry in large scale.

MILOS (Minimally Invasive Long Term Organizational Support) aims at offering support for agile processes by providing collaboration and coordination technology for distributed software development [16]. It also supports project planning and knowledge management. MILOS is constituted of the following components: a workflow engine, an experience base, and a resource pool. MILOS also supports agile software development with the use of some Extreme Programming (XP) practices. The MILOS PSEE does not provide support to important knowledge management tasks, such as consultation of organizational members' skills and knowledge that fulfill projects specific needs. Moreover, the MILOS PSEE does not model a comprehensive set of the Software Engineering Domain. Therefore, efficient integration of MILOS tools to other tools that support different areas of Software Engineering is not to be guaranteed.

Artemis 7 is a Web-based software developed to support business processes and roles associated, and to enable deployment of multiple solutions on a common platform [17]. Artemis 7 allows configurable access levels based on role and rights granted that allow users to access the various modules and features of the solution based on their individual needs. This approach ensures that each user need only see the functionality and information necessary to perform their responsibilities, thereby making the application easier to use for all stakeholders. Since Artemis 7 were developed to be used in general domains, the definition, design, implementation and integration of tools to support specific needs in the Software Engineering area are not feasible.

4 The Taba Workstation

The Taba Workstation is an enterprise-oriented Process-centered Software Engineering Environment (PSEE) that supports 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. It also integrates knowledge management activities within software processes aiming to preserve organizational knowledge, and to foster the institutionalization of a learning software organization. The workstation also provides the infrastructure to the development and integration of tools to support the execution of software processes. Moreover, this infrastructure maintains a useful repository containing software project information gathered across its life cycle [7, 10, 11, 12].

In order to support the definition, deployment, and improvement of software processes, the Taba Workstation supports the definition of organizational standard processes and tailoring of these processes to specific projects aiming to increase the control and improve the quality of software products. Therefore, the Taba Workstation not only supports software engineers in the execution of software development processes activities, but also provides the means to execute these processes according to organizational software development processes.

During the last years, the Taba Workstation evolved to comply with the different levels of capability maturity models of software organizations. Therefore, the main objectives of Taba Workstation are:

- to support the configuration of process-centered software engineering environments for different organizations (Configured PSEE);
- to support the automatic generation (i.e., instantiation) of software engineering environments for specific projects (Enterprise-Oriented PSEE);
- to support software development using the instantiated environment; and
- to support the management of organizational knowledge related to software processes.

The Taba Workstation has been used by the Brazilian software industry since 2003. The Taba Workstation was identified during three official SCAMPI appraisals for CMMI Level 2 as one of the greatest organizational strengths to facilitate the success of software process deployment initiatives and to overcome the inherent difficulties. Moreover, the Taba Workstation was also identified as an important organizational asset to guarantee the quality of software process and product quality in other three official MR-MPS.BR appraisals.

The Taba Workstation is constituted of integrated tools to support software processes definition, deployment and enactment. These tools are adherent to the practices of the CMMI levels 2 and 3 process areas. The functionalities of other tools to support Knowledge Management activities are integrated into the environment to facilitate the preservation of organizational knowledge and support activities execution.

The next section presents the software process definition approach adopted in the Taba Workstation. The functionalities of specific tools to support software process deployment and enactment are presented in section 4.2. Section 4.3 presents the main characteristics of the Taba Workstation that helps organizations to obtain success in their software process deployment initiatives based on CMMI and MR-MPS.BR.

4.1 Software Process Definition Approach

The Software Processes definition approach adopted in the Taba Workstation establishes phases and intermediary products using the ISO/IEC 12207 [3] as a basis for defining standard software processes from the Taba Workstation. Figure 1 depicts the presented approach.

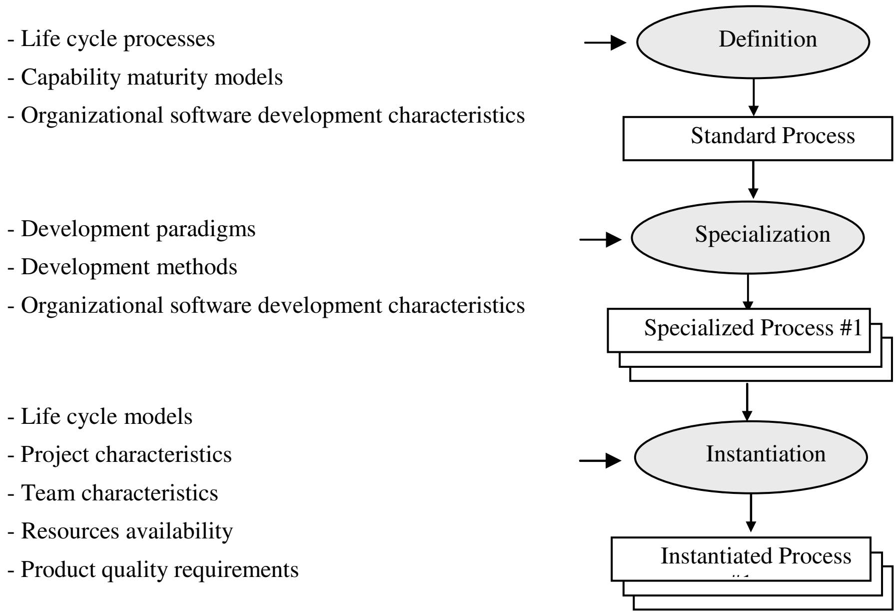

Fig. 1. Software processes definition approach

The standard processes and the specialized processes are considered to be organizational level processes. The instantiated processes are project level processes. This approach guarantees some practices of CMMI level 3 process areas, for instance, the establishment of defined processes for each process area.

4.1.1 Defining Organizational Software Process

During the Standard Process definition phase we not only consider the ISO/IEC 12207, but organizational software development characteristics related to the work environment, knowledge and experiences of the teams involved and the organizational software development experience and culture are also considered.

From the Standard Process, different software processes can be specialized according to different software types produced by the organization, (for instance, specialists systems and information systems) and to development paradigms adopted (for instance, object oriented or structured). During this phase, new activities can be defined and inserted into the specialized processes and the activities execution description can be adapted. Nevertheless, all the basic elements defined in the Standard Process must always be presented in the specialized processes.

The definition of organizational standard process for a specific organization is done during the configuration of a specific environment for the organization. The tool responsible for supporting this configuration is named Config. This tool supports the following activities:

- Configuration contextualization;
- Definition of environment configuration proposal;
- Definition of standard process;
- Definition of specialized processes;
- Definition of domain theory and tasks descriptions;
- Generation of configured environment

The configured environment for the organization contains not only the standard process and the specialized processes, but also specific knowledge related to software development and maintenance. By using this environment, the software engineers are enabled to generate instantiated environments to each of the projects to be developed.

4.1.2 Instantiating Software Process to Specific Project

In order to be used in a specific project, the most adequate specialized process to a specific project must be instantiated to satisfy the characteristics of the project (for instance, size and complexity of the product and relevant quality characteristics), development team characteristics, etc. In this phase, the life cycle model, methods and tools are selected.

The figure 2 presents a screenshot of a tool named AdaptPro that supports the institutionalization of the organizational processes since it facilitates the adoption of these processes in all the projects of the organization. By using the AdaptPro tool, the

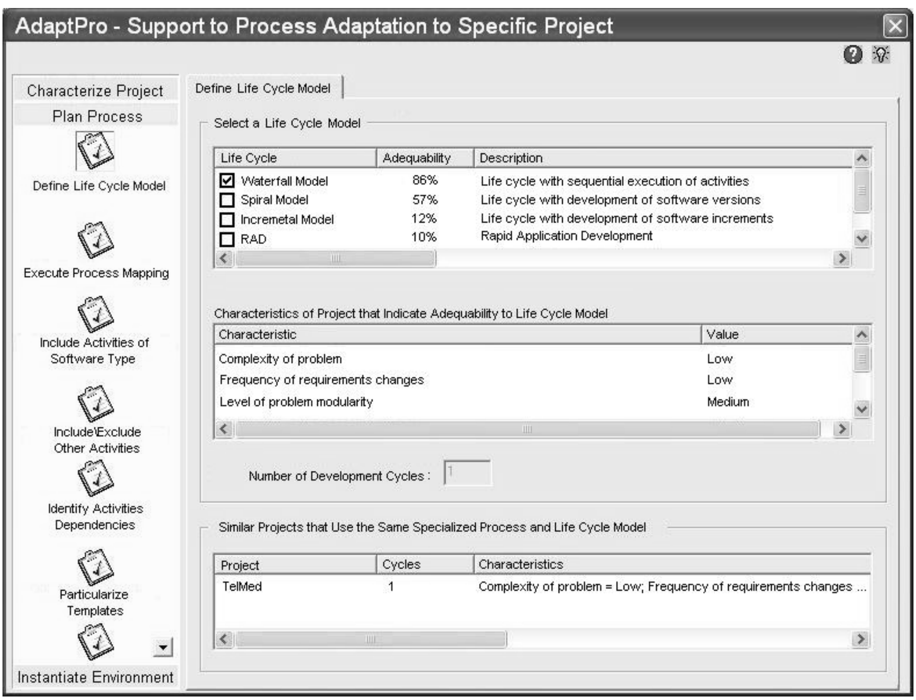

Fig. 2. AdaptPro tool to support instantiation of software process to specific project

software engineer 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 considering the project characteristics; and (iii) instantiate a PSEE to support the execution of the planned process.

On the left side of figure 2, 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 adequability 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 adequability 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.

The next sections present specific Taba Workstation tools to support software process deployment and enactment.

4.2 Supporting Software Process Deployment and Enactment

Once the software process for a specific project has been defined and a Software Engineering Environment has been instantiated, the basic means for software process deployment and enactment are established. Software process enactment involves coordination of relevant team members to enact various tasks, i.e., the enactment of a software process is the procedure of enacting various partially-ordered tasks to achieve the process objectives [18]. Therefore, the PSEE supports software process enactment by guaranteeing that software process information and resources are appropriately organized for their effective use, and as a consequence, the process can easily be put into action. Figure 3 presents the picture of the main interface of a Software Engineering Environment instantiated to a specific project.

On the left of the picture, it is presented the instantiated software process organized in terms of project phases, activities and sub-activities. By selecting a phase or activity, the system presents to the user on the right of the picture important information related to the element selected, for instance, associated tools, artifacts produced and consumed, and information related to the execution of the activities (for instance, time and effort estimates).

From the PSEE, the process's executers can execute tools associated to perform a specific activity. The executer can also download controlled versions of artifacts to be consumed in order to perform the activity. Once the activity has been initiated or concluded, the executer can upload to the system all the artifacts produced during that activity. From the main interface, the user can also directly consult knowledge related to the process activities, for instance, programming patterns and detailed software inspection procedures and techniques.

The process enactment is supported through the control of information related to activities entry and exit criteria, activities responsibilities, processes sequences derived from decision making situations, concurrency of activities, etc. The system also provides the means to identify process critical paths in order to support the monitoring and controlling of processes execution.

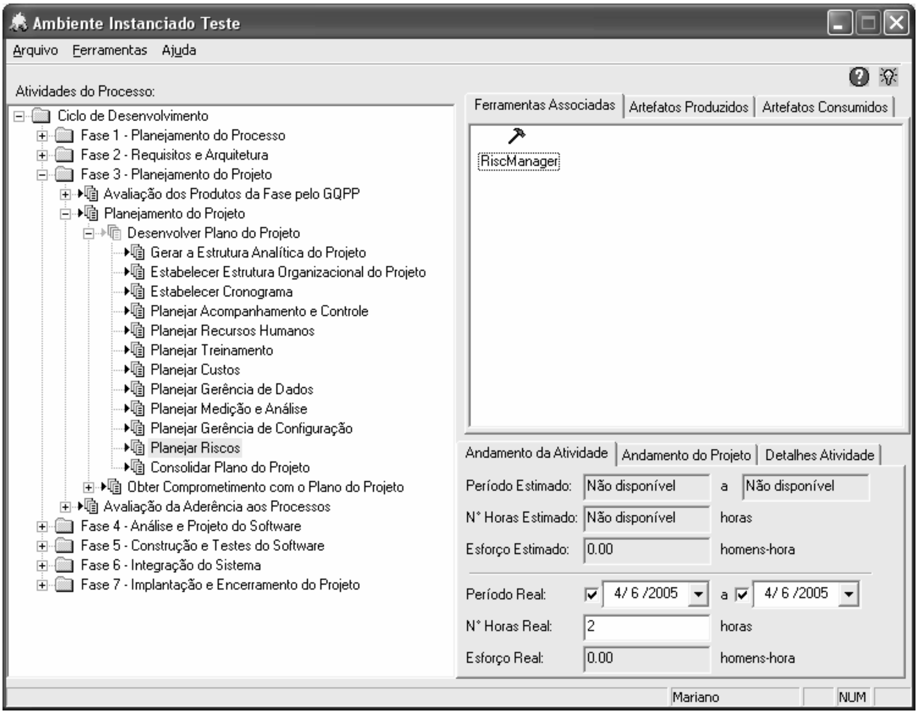

Fig. 3. Software Engineering Environment to support specific project software process deployment and enactment

4.3 Using the Taba Workstation to Guarantee the Success of Software Process Deployment Based on CMMI and MR-MPS.BR

In the context of software process deployment, we executed a survey with the objective to identify success factors and difficulties related to software process deployment experiences. The participants of the survey were members of COPPE/UFRJ, an institution of the Federal University of Rio de Janeiro with vast experience in software process research and deployment [7, 8, 10, 11, 12]. The success factors and difficulties identified through this survey were grouped according to the category of the findings. 12 categories were identified related to success factors and 16 categories related to difficulties in software process deployment experiences based on CMMI and MR-MPS.BR. From a comparative analysis of these findings, we identified important factors that contribute significantly to the success of software process improvement programs in small, medium and large organizations.

The consideration of such factors during software deployment initiatives can significantly increase the success of software process improvement programs, because they can help organizations to tailor their process deployment strategies considering particularities of the software development organizations and available resources. Figure 4 depicts the distribution of those factors according to categories of the findings.

The Taba Workstation provides the means to assure that most of those factors are strongly present during software process deployment initiatives based on CMMI and MR-MPS.BR. Moreover, it facilitates the endurance of software processes deployed over time.

The most relevant success factor in software process deployment is related to the commitment obtained from organization members and high management. Our experience demonstrated that the results are often satisfactory when the organization members are committed with the deployment process and the high management continuously supports the execution of the activities. The lack of commitment of the high management with the deployment process, and the lack of involvement of the organization members were also considered to be great difficulties in deploying software processes. Since, the Taba Workstation efficiently supports the enactment of software process and provides accurate project status reports to high level managers, the probability of lack of high management commitment is significantly reduced, because high management strategic decisions are based on the data extracted from the Taba Workstation. In order to provide such data, the organization members must be committed to the processes definition and the procedures deployed through the Taba Workstation.

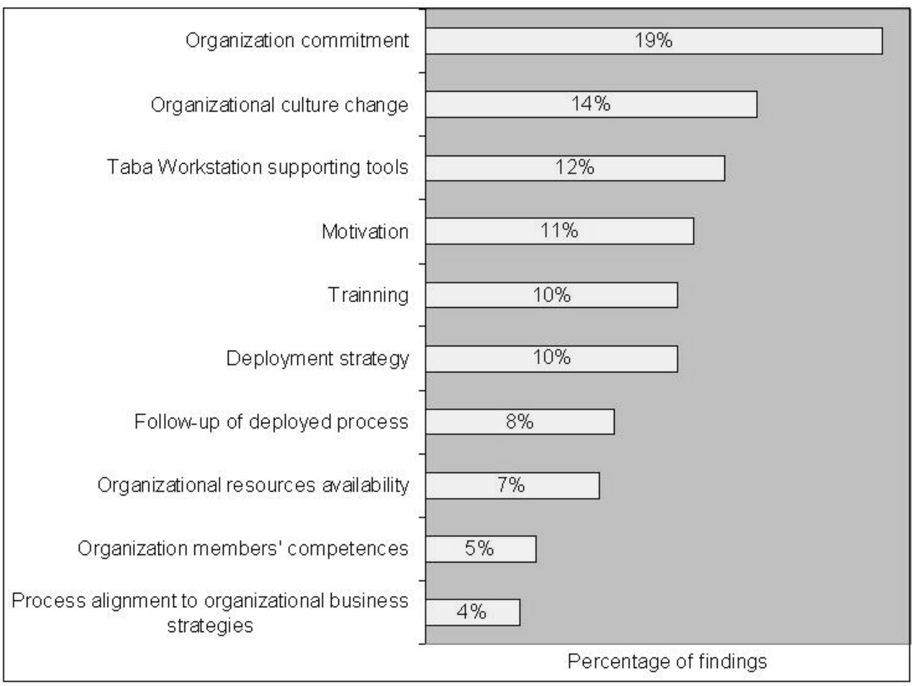

Fig. 4. Important factors that contribute significantly to the success of software process deployment based on the CMMI and the MR-MPS.BR

Another difficulty found in software process deployment is related to organizational culture change. In our deployment experiences, we found great difficulty in customizing the standards process according to organizational needs when a not completely correct culture about Software Engineering procedures (system analysis, testing, documentation, etc) were already disseminated within the organization. Moreover, we noticed great resistance from software project developers during the deployment of process activities that were traditionally executed *ad hoc*.

One of the main characteristics of the Taba Workstation is to guarantee the institutionalization of the organization processes through automated support of important software engineering tasks, such as project process definition, and collection of project measures. Therefore, most of the culture change impact can be minimized by reducing the effort of software process activities deployed. Moreover, since all Taba Workstation tools are process driven and integrated to a knowledge base, most of the difficulties of executing a new process demonstrated to be easily overcome by providing important knowledge related to the current process activity in the exact moment that the executer needs it, such as organizational directives and lessons learned from past project experiences.

The organization members' motivation was also a very important factor in software process deployment. This motivation occurred in various levels. The high management was motivated to deploy software process, because their main objective was to successfully achieve an official certification/appraisal of the software process reference models due to clients' pressure and market competitive needs. The motivation of other organization members was related to the need to learn and improve the execution of their activities. The Taba Workstation could satisfy both needs, because the institutionalization of the PSEE speeded the execution of the processes deployment and enactment, and provided the means for deployment of new and competitive technologies, i.e., organizational competitive advantages and members satisfaction increased due to the fact that organizational members were able to not only learn about new technologies, but also to apply them in real projects in a reduced time.

Since the delivery of software engineering trainings was considered essential to deal with the lack of organization members deficiencies in software engineering and to guarantee and adequate execution of the software process, the Taba Workstation tools integrated to knowledge management applications were also identify as an important factor, because it helped to diminish the training effort necessary to execute the process.

Many difficulties were related to the deployment strategy adopted by the software process deployment team. For instance, deployment strategies that required approval of many organization members in order to deploy a specific practice usually took a great amount of time since many conflicting needs had to be dealt with. In order to deal with this difficulty, our software process deployment strategy supported by the Taba Workstation defines that, no matter the level of the capability of the organization, all processes of the projects will have to be based on an organizational standard process defined by our consultants and a specific organization process group.

That aspect of our strategy requires less time to define organization processes, and speeds the institutionalization of organizational processes. Adjustments of these processes are executed on-the-fly and new processes definitions are derived through the

Taba Workstation Software Process Modeling tool. This characteristic of our deployment strategy also contributes to another important aspect: alignment of software process with the organization business strategies to obtain software processes that satisfy organizational development characteristics. This characteristic also helps to diminish the impact of organizational culture changes.

The greatest difficulty found in our software process deployment experience is related to deficiency of organization members' competences. The most relevant deficiency was the little knowledge in Software Engineering. Once this difficulty was found in an organization, most of the procedures, methods and techniques used to support software development had to be taught, for instance, how to describe a use case, classes' diagrams and requirements specifications, etc. This difficulty is related to the lack of organization members' computer science background knowledge.

In order to overcome that difficulty, before deploying the software processes in the organization, we fill the Taba Workstation Knowledge Base with important theoretical knowledge related to software engineering area and systems analysis methodologies. During the follow-up of the projects, our consultants are instructed to access the knowledge stored in that base and to discuss it with the organization members. This practice allows the organization members to learn about Software Engineering during the execution of their daily activities.

Another important consideration to be stated is that the amount of time dedicated to support software process deployment in the organization especially during pilot projects, and the dedication of the deployment team in the organization are key factors to guarantee the success of the deployment. Technology and knowledge transference demands a lot of involvement of the people related to the deployment process.

The results of the software process improvement program are not satisfactory when the cost of deployment restricts the amount of time dedicated to support software process deployment. This factor is directly related to the availability of financial resources to spend on software process deployment activities. During these activities, the organization must be able to provide sufficient financial resources in order to cope with dynamic deployment necessities.

5 Conclusions

This work presented the Taba Workstation, an enterprise-oriented PSEE developed to support software process deployment based on the CMMI and the MR-MPS.BR. Characteristics of Taba Workstation that increases the efficiency and efficacy of software process deployment initiatives and reduces the inherent difficulties were also presented by comparing these characteristics to important factors that contribute significantly to the success of software process deployment based on the CMMI and the MR-MPS.BR.

Since the Taba Workstation is based on a Software Engineering Ontology, the integration of Taba Workstation tools to other tools that support different areas of Software Engineering is facilitated. Moreover, the Taba Workstation software process definition approach demonstrated to support process modeling and evolution in an efficient and efficacy way. One of the mayor contributions of the Taba Workstation is that its architecture and software development supporting tools were modified over the past years to become more adequate to the necessities of software organizations executing real projects in dynamic and evolving environments. Another important aspect of Taba Workstation in this evolution is that it supports important knowledge management tasks, such as consultation of organizational members' skills and knowledge that fulfill projects specific needs. These characteristics facilitate preservation of organizational memory.

The Taba Workstation is continually evolving. The next steps is to evaluate the adequacy of the tools that support CMMI Level 3 process areas, and to define and integrate other tools to support CMMI Level 4 and 5 process areas and to facilitate the elevation of organization software development maturity to higher levels.

References

1. Arent, J. and Norbjerg, J.: Software process improvement as organizational knowledge creation: a multiple case analysis. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pp 1-11, Jan 4-7 (2000)
2. Fuggetta, A.: Software Process: A Roadmap. In: The Future of Software Engineering, Ed. Anthony Finkelstein, 22nd Int. Conference on Software Engineering, pp. 27-34 (2000)
3. ISO/IEC 12207:2000 - Information technology – software process life cycle (2000)
4. ISO/IEC 15504 –1 Information Technology – Process Assessment, - Part 1: Concepts and Vocabulary (2003)
5. Chrissis, M. B., Konrad, M, Shrum, S.: CMMI: Guidelines for Process Integration and Product Improvement. Addison-Wesley (2003)
6. MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Geral (v. 1.0) (2005)
7. Santos G., Montoni M., Rocha A. R., Figueiredo S., Mafra S., Albuquerque A., Paret B. D., Amaral M.: Using a Software Development Environment with Knowledge Management to Support Deploying Software Processes in Small and Medium Size Companies. In.: Lecture Notes in Artificial Intelligence, ISBN 3-00-016020-5, pp 72-76, presented at the 3rd Conference Professional Knowledge Management Experiences and Visions, Kaiserslautern, Germany, April 10-13 (2005)
8. Rocha, A. R., Montoni, M., Santos, S., Mafra, S., Figueiredo, S., Albuquerque, A., Mian, P.: Reference Model for Software Process Improvement: A Brazilian Experience. In.: Lecture Notes of Computer Science (LNCS), ISBN 3-540-30286-7, pp. 130-141, presented at the European Software Process Improvement and Innovation Conference (EuroSPI 2005), Budapest, Hungary (2005)
9. Weber, K.C., Araujo, E.R., Rocha, A.R., Machado, C., Scalet, D., Salviano, C.: Brazilian Software Process Reference Model and Assessment Method. In.: Computer and Information Sciences – ISCIS 2005, LNCS 3733, pp 403-411 (2005)
10. Montoni M., Santos G., Villela K., Rocha A. R., Travassos G. H., Figueiredo S., Mafra S., Albuquerque A., Mian P.: Enterprise-Oriented Software Development Environments to Support Software Products and Process Quality Improvement. In.: Lecture Notes of Computer Science (LNCS), ISBN 3-540-26200-8, pp. 370-384, presented at the 6th Int. Conference on Product Focused Software Process Improvement, Oulu, Finland, June (2005)
11. 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. In.: Lecture Notes of Computer Science (LNCS), ISBN 3-540-24088-8, pp. 117–128, presented at the 5th Int. Conf of Practical Aspects of Knowledge Management, Vienna, Austria, (2004)

12. Montoni, M., Miranda, R., Rocha, A. R., Travassos, G. H.: Knowledge Acquisition and Communities of Practice: an Approach to Convert Individual Knowledge into Multi-Organizational Knowledge, In.: Lecture Notes in Computer Science (LNCS), ISBN 3-540-22192-1, pp. 110-121, presented at the 6th International Workshop on Learning Software Organizations (LSO'2004), Banff, Canada, June (2004)
13. Minh, N. N., Wang, A.I., Conradi, R.: Total Software Process Model Evolution in EPOS Experience Report. In: Proc. of the 19th Int. Conf. on Software Engineering, pp: 390–399, May 17-23 (1997)
14. Ben-Shaul, I.Z., Skopp, P.D., Heineman, G.T., Tong, A.Z., Popovich, S.S., Valetto, G.: Integrating groupware and process technologies in the Oz environment. In: Proc. of the 9th Int. Software Process Workshop, pp.: 114–116, 5-7 Oct. (1994)
15. S. Bandinelli, Di Nitto, E., Fuggetta, A.: Supporting Cooperation in the SPADE-1 Environment. IEEE Trans. on Software Engineering, Vol. 22, No. 12, pp. 841-865 (1996)
16. Bowen, S., Maurer, F.: Process support and knowledge management for virtual teams doing agile software development. In.: Proc. of the 26th Annual Int. Computer Software and Applications Conference (COMPSAC) pp:1118–1120, 26-29 Aug. (2002)
17. Artemis 7, http://www.aisc.com/Product/1 (2006)
18. Yan, J., Yang, Y., R., G. K.: Decentralized Coordination for Software Process Enactment, F. Oquendo (Ed.): EWSPT 2003, LNCS 2786, pp. 164–172 (2003)
19. ISO/IEC 12207:2000 - Information technology –software process life cycle, (2000)
20. ISO/IEC 15504 –1 Information Technology – Process Assessment, - Part 1: Concepts and Vocabulary, (2003)
21. Chrissis, M. B., Konrad, M, Shrum, S.: CMMI: Guidelines for Process Integration and Product Improvement, Addison-Wesley, (2003)