SOFTWARE PROCESS CONVERSION RULES IN *IMPPROS*
Quality Models Conversion for a Software Process Implementation Environment

Sandro Ronaldo Bezerra Oliveira
Centro de Ciências Exatas e Tecnologia – Universidade da Amazônia (UNAMA)
Av. Alcindo Cacela, 287, 66060-902 – Belém – PA – Brasil

Alexandre Marcos Lins de Vasconcelos, Tiago Soares Gonçalves
Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brasil

Keywords: Process Conversion, Software Process, Quality Models, Software Development Environment.

Abstract: The software process conversion is a technique based on the mapping of the existing relationship between the content of the quality norms/models. The basic estimated of the conversion is to obtain making an adaptation of the software processes without the necessary effort to specify new models, guaranteeing the unicity and the consistency. For a company to reach a definitive market, its software process will have to be guided by patterns defined for a norm, and if it glimpses the penetration in other markets perhaps it is necessary the guide for other norms so different. This paper presents a process to convert software processes using quality models/norms, and a discussion of some rules used to support the execution of this process in a software development context. This process is part of a software process implementation environment, called *ImPProS*, developed at CIn/UFPE – Center of Informatics/Federal University of Pernambuco.

1 INTRODUCTION

The existing technology for the software processes definition has some limitations (Rocha, 2001). The trend of the organizations in making use quality models and standards of process, it is excellent to identify the interrelation between the assets (useful elements of the quality models and standards to improve the performance of the processes used in the some projects) of these models and standards, in order to provide a mapping and an unicity of its terminologies, and so that the result of this mechanism can be used for the software process definition adherent to a specific model or capable to be adapted the different quality models. The support automated for this practical does not have to be oriented for specific models and must make possible a dynamicity to add new models to the structure of its meta-model (repository of components of the software process that it becomes the definition of this one more flexible, therefore the types to be manipulated are not pre-defined) of processes.

One of the problems the technology of software processes definition that was used as motivation for the present work is in the fact with the great amount of quality models and standards of process existing in the market, the support automated to the software process definition does not foresee a conversion of the assets defined to the processes in relation the different quality models, or either, there is not possibility, for instance, the structure of a process defined from CMMI (Chrissis, 2003) can be converted for the structure proposal for ISO/IEC 15504 (ISO, 1998), thus providing a inflexibility the processes defined (Oliveira, 2006b).

To treat this motivation, the following decision was taken: the analysis of an only software process defined from different quality models, is kept with the specification of conversion mechanism of software processes and the inter-relations existing between the assets of the quality models and standards kept in the software process meta-model.

Thus, the importance for the organizations of software development to adapt or to convert its software process from the many existing quality

Ronaldo Bezerra Oliveira S., Marcos Lins de Vasconcelos A. and Soares Gonçalves T. (2007).
SOFTWARE PROCESS CONVERSION RULES IN IMPPROS - Quality Models Conversion for a Software Process Implementation Environment.
In *Proceedings of the Second International Conference on Software and Data Technologies - SE*, pages 258-263
DOI: 10.5220/0001324802580263

norms/models is proven, that is why to establish in definitive markets it is necessary to guide itself by norms/models more accepted in this context.

Therefore, the intention of this work is to define a methodology of software processes conversion from quality models/norms and to describe the details of the rules that evidence this conversion, integrated to the *ImPProS* (Gradual Software Processes Implementation Environment) (Oliveira, 2005). This environment will make possible the specification of the processes in accordance with a specific project domain and the characteristics of the organization; the instantiation of the process for projects properties; its simulation from the configuration parameters (stated period, pressures, cost, resources, etc.); its execution based on the organizational process; and its evaluation through metrics collected about its execution.

Besides this introductory section, the paper presents other four sections. Section 2 describes the properties that compose the software process implementation environment. In section 3 the detailing the workflow of software process conversion is presented. The section 4 presents the rules specified to automate this one. Finally, section 5 presents the final considerations of this paper.

2 *ImPProS*: A SOFTWARE PROCESS IMPLEMENTATION ENVIRONMENT

The *ImPProS* is a project which is being performed at the Center of Informatic of UFPE – Federal University of Pernambuco with the partnership of UNAMA - University of Amazônia, and financed by CNPq - National Agency for Scientific and Technological Development. The objective of *ImPProS* is the creation of an environment to support the implementation of a software process in an organization in a gradual way. The "gradual" term means that the implementation of the process is improved with the experiences learned in its definition, simulation, execution and evaluation.

The *ImPProS* is composed of a cooperative environment, formed by ten main tools (Oliveira, 2005): the definition, simulation, automatized execution and evaluation of software process from the analysis of specific characteristics and learning acquired; the systematic execution of activities the software process improvement, from the IDEAL model; the analysis and taking of decision concerning the evaluation items which composes the software process; the execution of software process reuse from the definition of project scope and its adaptation to the use context; the collection, analysis and use of knowledge learned during the execution of software process; the conversion of software process components from the structures of quality norms/models and their mapping; and the diagrammatic modelling of the software processes assets.

To take care of the needs of all these tools, it searches to design a repository that could add the concepts considered for the quality norms and models, the software process meta-model (Oliveira, 2006a). Thus, from the need proposal for *ImPProS* about its software process definition from quality norms and models, the concept of Reference Models was analyzed initially, that are structures of the conceptual point of view, that they allow to consider alternatives of implementation in different computational contexts, as well as arguing and the comparing proposals of these contexts on the same point of view (Alves, 2000).

From this one, it is fact to perceive the existence of reference models, in the quality context, more abstract than others about the use of its assets for the software processes definition. To treat this abstraction, this work defined two types of quality standards for this goal (Oliveira, 2006a): the ones that we call Concrete Reference Models, which describes orientations for the processes definition and implantation through practices specified (activities, tasks, work-products, etc.) that they are used directly in the software process, for example the CMMI; and Abstract Reference Models, equal to the abstract reference models however it does not indicate practices for the composition of processes clearly, but intentions (goals to be reached), results waited (artefact produced, a significant change of state and the attendance of the specifications) and additional information (references that can help in the process definition and implementation) that they are described through a relationship with concrete reference models, it can mention the MPS.Br (Softex, 2006), ISO/IEC 15504.

Beyond these two types of quality standards, the meta-model of *ImPProS* used ISO/IEC 12207 norm (ISO, 1997) for establishing a common architecture for the software processes life cycle with a terminology well defined, contends processes, activities and tasks to be applied during the software products supply, development, operation and maintenance (Oliveira, 2006a). It allows that the processes are specified with a unified terminology and this norm serves to promote the relationship between the quality norms and models from the mapping of processes and practices recommended by these norms and models to the processes, activities and tasks constant in ISO/IEC 12207 norm.

Thus, it specified three existing types of relationship between the quality norms and models (Oliveira, 2006a):

- the mapping of processes and activities of the concrete reference models to the processes and activities of ISO/IEC 12207 norm, it aims to standardize/unify the concepts defined for the two types of norms about the software development cycle, in order to take advantage the terms used for ISO/IEC 12207 norm because it treats the common architecture of software process life cycle;
- the mapping of processes of the abstract reference models to the processes of ISO/IEC 12207 norm, to provide the same goal of the relationship previously specified. It is important to stand out that it does not make the mapping of activities in this case, because the abstract reference models has results waited and not tasks to be executed in a software process;
- the composition of results waited of abstract reference models to the activities of concrete reference models, which guides as a reference of the model can be implemented in a software process, or either, as the abstract reference model does not define "how" a process must be implemented, but "what" this process must generate as resulted, this type of relationship makes possible to specify the detailing of these results from the tasks defined for the concrete reference models.

3 A SOFTWARE PROCESS CONVERSION MODEL

The existence of ISO/IEC 12207 norm, and the reference models ISO/IEC 15504 (SPICE), CMMI and MPS.Br, and others, implies in a lack of unicity of standards, thus bringing a problem for the organizations of systems development. For one organization reaches a definitive market, its software process will have to be guided by the standards defined by a norm, according the above mentioned ones (Gonçalves, 2006). If this one glimpses to establish in other markets has the need to take as guide other sufficiently different norms. This way, it becomes important to adapt or to transform/convert processes in accordance with the need of market.

The conversion of software processes is a technique based on the mapping of the existing relationship between the assets of the quality norms and models (Gonçalves, 2006). The basic estimated of the conversion is making an adaptation of these processes no adding so great effort to specify new models, guaranteeing unicity and consistency. As part of this work, a management mechanism was defined and the development of a tool, called *ProConverter*, that it is a prototype for the automation of this conversion. This section supplies a description of the *ProConverter* mechanism. Later, it will be presented the workflow and the rules of mapping between quality norms and models automated by the activities defined to the tool.

ProConverter is a support tool to the *ImPProS* with the goal to provide the conversion the software process in implementation in this environment, or either, from the use of abstract and concrete reference models, this tool analyzes the processes and activities that compose a process implemented in the *ImPProS* and convert them from the mappings made between the models, constant in the process meta-model of this environment. Above, each one of the activities presented in the conversion workflow is detailed:

- **Creating Conversion Project**: the focus of this activity aims to create a project in *ProConverter* for the software process conversion, or either, an area in *ProConverter* that it makes possible to store and keep the inherent information to the execution the life cycle of software process conversion from a specific identification (name, description, date and responsible user for conversion);
- **Configuring Conversion**: in this activity it is selected the organization which intends to choose a software process to be converted. From this one, the level of software process definition (Standard Process, Specialization or Instantiation of Process) of origin is chosen and finally the software process that will be converted;
- **Visualizing Process**: in this activity it is possible to visualize the details of all assets (life cycle models, software processes development life cycle, activities, artifacts, resources, procedures, etc.) of the software process chosen for execution of the conversion;

- **Parametering Conversion**: the focus of this activity is to parameter the software process conversion, or either, to establish the parameters of identification the new resultant process of the conversion. This way, initially it is defined the name, the description and the date of generation of the converted process. Later it is chosen the quality norm and model, which this process will be based during the conversion, or either, the new quality norm and model that the converted process will have its assets adherent to the end of the execution the rules of mapping the conversion, described in section 4;
- **Parametering *ImPProS* Project**: if during the Configuring Conversion activity was selected the level "Standard Process", it becomes necessary to create a project in the *ImPProS* in order to associate the process converted for its manipulation and modification into the environment, because it is the first level of software process definition of *ImPProS* environment. The others process levels (Specialization and Instantiation of Process) have an automatic reference when the choice of process to be converted, or either, the specialized process is associated with a standard process and, therefore the instantiated process is associated with a specialized process, kept in *ImPProS*;
- **Mapping Process Directly**: based in the configuration and parametering of conversion defined in the Configuring Conversion and Parametering Conversion activities, all the possible processes of the software development life cycle and the activities that compose the origin process (process selected in the Configuring Conversion activity) are mapped directly in accordance with the mapping rules described in section 4;
- **Mapping Process Indirectly**: after the direct mapping the part of content of origin process, all the possible processes of software development life cycle and activities of this origin process are indirectly mapped in accordance with the mapping rules described in section 4;
- **Characterizing and Configuring Converted Process**: it allows to the visualization and analysis of the process converted from the processes of software development life cycle and activities mapped directly and indirectly, and the configuration in *ImPProS* of this process to be manipulated and modified in accordance with the needs of its definition.

4 MAPPING RULES BETWEEN QUALITY NORMS/ MODELS

According with the types of mappings considered in section 2, it is possible to provide for *ProConverter* a directed service to the software processes conversion in accordance with the quality norm and model defined for the process of organization, or either, to convert a software process from the terminologies (processes and activities) present in the quality norms and models available and mapped in *ImPProS*. This way, the conversion proceeds analyzing the terminologies inferred to a software process, later verifies the mapping rules (possible relations between the terminologies of the quality norms and models) available in the process meta-model of *ImPProS* and becomes these terminologies for the standards adopted for quality norm and model chosen (Gonçalves, 2006).

This conversion happens in two levels: **Direct Conversion**, which the terminologies analyzed and converted are originated from quality norms and models and its conversions obey the types of mapping (Mappings of Processes and Activities between ISO/IEC 12207 norm and Concrete Reference Models; Mappings of Processes between the Abstract Reference Models and ISO/IEC 12207 norm; Composition of the Results Waited in Activities between the Abstract Reference Models and the Concrete ones) considered in the process meta-model of *ImPProS*; and **Indirect Conversion**, which the origin of terminologies of processes and activities is proceeding from the organization, type of software project, type of organization, generic or the quality norms and models that are not contemplated in the mappings included in the process meta-model of *ImPProS*. This classification serves to identify the software process assets that are mapped in the meta-model of *ImPProS*.

It still is important to emphasize that, because the *ImPProS* makes use of two types of quality standards, defined in section 2, the rules present in this conversion vary depending on the type of quality norm and model defined for the origin software process (process that is being used for the execution of the conversion mechanism) and the definite type to the process to be converted (resultant process of the execution of conversion mechanism), or either, during the software process definition in

ImPProS, the work team chooses a quality norm and model for guiding its definition and another one to convert this process already defined (Gonçalves, 2006). This way, the rules, besides being categorized in accordance with the type of conversion, vary in function of the two possible types of quality norms and models to the software processes: the first definition is referenced in the possibility of the origin process to be based on a Concrete or Abstract Reference Model and the process to be converted to be based on another Concrete Reference Model; and the second one references the possibility of the origin process to be based on an Abstract or Concrete Reference Model and the process to be converted to be based on another Abstract Reference Model.

Above, a detailing of the rules characterized like Direct is done. It allows an understanding of the conversion rules in accordance with the levels of process conversion and the types of definition the quality norms and models:

- If a process of software development life cycle or activity present in the origin software process of organization will be of ISO/IEC 12207, it is checked in the meta-model of *ImPProS* if this one has some mapping with some process or activity that is the Concrete Reference Model defined to the software process to be converted. If it is positive, the result of the conversion references this process or activity mapped;
- If an activity presents in the origin software process of organization will be of the same type of Concrete Reference Model defined to this process of organization, is checked in the meta-model of *ImPProS* if this one has some mapping with some activity of ISO/IEC 12207. If it is positive, it checks in the meta-model of *ImPProS* if this(these) activity(ies) of ISO/IEC 12207 has(have) some mapping with some activity that is the Concrete Reference Model defined to the process to be converted. If it is positive, the result of the conversion references this activity mapped;
- If a process of software development life cycle presents in the origin software process of organization will be of the same type of Concrete or Abstract Reference Model defined to this process of organization, it is checked in the meta-model of *ImPProS* if this one has some mapping with some process of ISO/IEC 12207. If it is positive, it is checks again in the meta-model of *ImPProS* if this(theses) process(es) of ISO/IEC 12207 has(have) some mapping with some process that is the Concrete or Abstract Reference Model defined to the process to be converted. If it is positive, the result of the conversion references this process mapped;
- If a process of software development life cycle presents in the origin software process of organization will be of ISO/IEC 12207, it is checked in the meta-model of *ImPProS* if this one has some mapping with some process that is the Abstract Reference Model defined to the process to be converted. If it is positive, the result of the conversion references this process mapped;
- If an activity presents in the origin software process of organization will be of the same type of Concrete Reference Model defined to this process of organization, it is checked in the meta-model of *ImPProS* if this one has some mapping with some result waited of the Abstract Reference Model defined to the process to be converted. If it is positive, it is checks again in the meta-model of *ImPProS* if this(these) result(s) waited has(have) some mapping with some activity that is some Concrete Reference Model, different of one defined to the origin software process of organization. If it is positive, the result of the conversion references this activity mapped.

And the rules considered Indirect are detailed above:

- If a process of software development life cycle or activity presents in the origin software process of organization will be of ISO/IEC 12207, it is checked in the meta-model of *ImPProS* if this one has some mapping with some process or activity that is the Concrete Reference Model defined to the process to be converted. If it is negative, the result of the conversion references to the process or activity deriving of ISO/IEC 12207;
- If an activity in the origin software process of organization will be of the same type of Concrete Reference Model defined to this process of organization, it is checked in the meta-model of *ImPProS* if this one has some mapping with some activity of ISO/IEC 12207: if it is negative, the result of the conversion references to the activity deriving of Concrete Reference Model defined to the origin process; if it is positive, it is checks again in the meta-model of *ImPProS* if

this(these) activity(ies) of ISO/IEC 12207 has(have) some mapping with some activity that is the Concrete Reference Model defined to the process to be converted, if it is negative, the result of the conversion references to the activity of ISO/IEC 12207;

- If a process of software development life cycle presents in the origin software process of organization will be of the same type of Concrete or Abstract Reference Model defined to this software process of organization, it is checked in the meta-model of *ImPProS* if this one has some mapping with some process of ISO/IEC 12207: if it is negative, the result of the conversion references to the process deriving of Concrete or Abstract Reference Model defined to the origin process; if it is positive, it is checks again in the meta-model of *ImPProS* if this(these) process(es) of ISO/IEC 12207 has(have) some mapping with some process that is the Abstract Reference Model defined to the process to be converted, if it is negative, the result of the conversion references the process of ISO/IEC 12207;
- If a process of software development life cycle or activity presents in the origin software process of organization will be specific of Organization or Generic, the result of the conversion references to this exactly process or activity, because it is not part of the existing mapping rules in the meta-model of *ImPProS*;
- If an activity presents in the origin software process of organization will be specific of the Type of Organization or the Type of Software, the result of the conversion references to this the same activity, because it is not part of the existing mapping rules in the meta-model of *ImPProS*;

5 FINAL CONSIDERATION

This work presented a proposal for conversion in the context of software process. This proposal consists of defining, configuring, parametering, mapping and characterizing processes of software life cycle defined for organizations related to their software processes according to a systematic and controlled methodology.

A tool was developed to support the execution of this methodology and an application of this tool was carried through in the software development context. This tool, called *ProConverter*, was integrated to the *ImPProS* environment.

An experimental study was executed in the context of micro and small companies which develops software from which one it was possible to evaluate the benefits of conversion and to identify improvements to be carried through in this approach. Currently, the tool was applied in the academic context, during the development of research projects by members of the *ImPProS* group at CIn/UFPE.

REFERENCES

Alves, C. F., Guedes, L. V., Pinto, R. C., 2000. Ferramentas CASE, O Modelo de Referência Suas Origens e Tendências, Orientador Alexandre Vasconcelos. Monografia de Disciplina. Recife-PE.

Chrissis, M. B., Konrad, M., Shrum, S., 2003. CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley.

Gonçalves, T. S., 2006. Análise da Conversão de Processos de Software a partir de Modelos/Normas de Qualidade, Orientador Prof. Alexandre Marcos Lins de Vasconcelos. Trabalho de Graduação. Recife-PE.

NBR ISO/IEC 12207:1995, 1997. Tecnologia de informação – processos de ciclo de vida de software, Associação Brasileira de Normas Técnicas.

ISO/IEC TR 15504, 1998. Information technology – software process assessment, International Organization for Standardization.

Oliveira, S. R. B., Vasconcelos, A. M. L., Rouiller, A. C., 2005. Uma Proposta de um Ambiente de Implementação de Processo de Software, Revista InfoComp – Revista de Ciência da Computação da UFLA – vol. 4, n. 1, Lavras-MG.

Oliveira, S. R. B., Vasconcelos, A. M. L., Pereira, J. F., Ramos, I. C., 2006a. A Process Meta-Model in a Gradual Software Process Implementation Environment, In Proceedings on The First International Workshop on Metamodelling – Utilization in Software Engineering – MUSE, Setúbal-Portugal.

Oliveira, S. R. B., Vasconcelos, A. M. L., Pereira, J. F., Ramos, I. C., 2006b. A Structure to Software Process Definition in ImPProS, Proceedings on EUROMICRO Work In Progress Session - Software Engineering and Advanced Applications (SEAA). IEEE Conference, Cavtat/Dubrovnik (Croatia).

Rocha, A. R. C., Maldonado, J. C. and Weber, K. C., 2001. Qualidade de software: teoria e prática, São Paulo: Prentice-Hall.

Softex - Sociedade para Promoção da Excelência do Software Brasileiro, 2006. MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral, versão 1.1.