

A Continuous Improvement Model in *ImPProS*

Sandro Ronaldo Bezerra Oliveira
*Centro de Ciências Exatas e Tecnologia –
Universidade da Amazônia (UNAMA)*
Av. Alcindo Cacela, 287 – 66060-902 –
Belém-PA-Brazil –
Phone: (+55 91) 4009-3000
srbo@cin.ufpe.br

Alexandre Marcos Lins de Vasconcelos
*Centro de Informática – Universidade
Federal de Pernambuco (UFPE)*
Caixa Postal 7851 – 50732-970 – Recife-
PE-Brazil – Phone: (+55 81) 2126-8430
amlv@cin.ufpe.br

In the intention to improve the software development and to get products with the desirable levels of quality, inside the schedule and budget considered, the last decade attended a change of approach about the software process [1]. Then, it has a new approach that the main focus is in guarantee of quality the own productive process, that's why it has shown the determinative factor for reach of final product quality.

From this change of focus, it was intensified research about development process and some norms and standards had been defined in order to assist the software process definition and improvement. With the intensification of these studies, it evidenced that was necessary to improve each step of the development life cycle to reach higher levels of quality. However, so that it became possible, quantitative data, that they could describe the reality of the process, needed to be gotten and duly analyzed.

To help in satisfying such need the Software Engineering Institute (SEI) developed and refined the IDEAL model [2]. The IDEAL model - Initiating, Diagnosing, Establishing, Acting and Learning, was originally conceived as a life cycle model for software process improvement based on the Capability Maturity Model (CMM) for software, and therefore this one used terms about processes improvement. The model supplies an approach of engineering disciplined for improvement, focus in the management of improvement program and establishes the foundation for an improvement strategy for a long time.

Thus, the goal of this work is analyzing each one of the phases that composes the structure of the IDEAL model, as well as its activities and principles, and adapting for the characteristics that the software process implementation environment must have about the continuous software process improvement. This environment helps an organization in the gradual

software process implementation, it is useful to supply automated support by means of an environment (*ImPProS* – Gradual Software Process Implementation Environment) capable to support the phases that the specialized literature considers as necessary: software process definition, simulation, execution and evaluation [4]. The *ImPProS* is a project of initiative the Center of Informatic at UFPE – Federal University of Pernambuco with the partnership at UNAMA - University of Amazônia, financed by CNPq - National Advice of Scientific and Technological Development, which aims the creation of an environment of support to the implementation a software process in an organization of gradual way. The “gradual” term denotes the fact that the process implementation is performed with the experiences learned in its previous phases. Thus, it is composed of a cooperative environment, formed for nine main tools [4]:

- **ProDefiner:** it provides the definition of software process from the analysis of specific characteristics and learning acquired with other definitions;
- **ProSimulator:** it makes possible the simulation of software process instantiated from a execution plan of the process and thus to foresee problems;
- **ProEnacter:** it allows the automatized execution and accompaniment of software process by project team;
- **ProEvaluator:** it provides the evaluation of software process execution from the analysis of qualitative and quantitative criteria;
- **ProImprove:** it makes possible the systematic execution of activities the software process improvement, from the IDEAL model;
- **ProAnalyser:** it allows the analysis and taking of decision concerning the evaluation items which composes the software process;

- **ProReuse:** it provides the execution of software process reuse from the definition of project scope and its adaptation to the use context;
- **ProKnowledge:** it makes possible the collection, analysis and use of knowledge learned during the execution of software process;
- **ProConverter:** it provides the conversion of software process components from the structures of quality norms/models and their mapping.

The focus this work is detailing the *ProImprove* workflow, based in studies and analyses of applications the IDEAL model phases and activities where it was adapted the model to the *ImPProS* through a set of activities, that can be specified through the workflow organized by responsible users. The following paragraphs explain how these activities function [3].

The **Registering Reason** activity is equivalent to Stimulus for Change phase in IDEAL model. It is in this phase that the process designer specifies the organizational reasons that had motivated the beginning of software process improvement. In the **Defining Improvement Context** activity, the designer describes how the improvement incases inside of the organization context, or either, how it affects the existing works in the organization and how the benefits that it brings.

In the **Building Improvement Support** activity some member of the high administration of organization assumes the sponsor role, legalizing its interest for the improvement and guaranteeing the necessary support so that it is executed. After to receive the necessary support, the process designer executes the **Placing Improvement Infrastructure** activity, preparing all the necessary resources so that the improvement execution is initiated and delegating responsibilities to users involved. After that, the designer defines the current and desired states for each software process practice that suffers improvement, through the **Characterizing Practices States** activity.

After defining the current and desired states for each practice clearly, the designer executes the **Defining Procedures** activity, in which specifies how the necessary procedures so that the desired states are reached. The next stage to be executed by designer is the **Definition of Priorities**. In this activity, the designer must select the subgroup of process practices that have the greater priority in reaching its desired states.

The next activity makes possible the designer creates a strategical plan of improvement execution, from the **Developing Strategies** activity. After that, the process manager can execute the **Planning Action**

activity, which makes possible the definition of an action plan for improvement deployment, it has a schedule of tasks, milestones and decision points. The action plan still includes a resources management, responsibilities attribution, definition of metric, mechanisms of tracking, planning of strategies and risks managements.

The manager can initiate the **Creating Solution** activity. The solution is composed for the tools and processes that are used; necessary knowledge and abilities; external support that can be required and any other additional information. This solution must be tested and refined. When solution will finally be validated, it is deployed from the execution of **Deploying Solution** activity.

After the deployment has carried through, the process designer must **Analyze and Validate the Improvement**, registering the learned lessons during its execution. These lessons generate a knowledge base about the method of improvement deployment, allowing that good practices are recognized and errors are prevented in the future. If the designer judges necessary, it can, still, **Consider Future Actions** that bring more efficiency and trustworthiness to the execution of future improvements.

References

- [1] A. G. Junior, "Avaliação de Processos de Software baseada em Medições", *Dissertação de Mestrado*, Orientadora Ana Regina Cavalcanti da Rocha. COPPE-UF RJ, 2001.
- [2] B. Mcfeeley, "IDEALSM: A User's Guide for Software Process Improvement", *Software Engineering Institute Handbook*. Carnegie Mellon University. CMU/SEI-96-HB-001, 1996.
- [3] P. O. A. M. Junior, "Utilização de Métricas em uma Ferramenta de Apoio à Melhoria Contínua de Processos de Software", *Trabalho de Graduação apresentado ao CIn/UFPE*. Orientador Prof. Alexandre Vasconcelos, Recife-PE, 2006.
- [4] S. R. B. Oliveira, A. M. L. Vasconcelos, A. C. Rouiller, "Uma Proposta de um Ambiente de Implementação de Processo de Software", *Artigo publicado na Revista InfoComp – Revista de Ciência da Computação da UFLA* – vol. 4, n. 1, Lavras-MG, 2005.