Deploying Software Reuse Management at COPPE/UFRJ Software Engineering Laboratory

Reinaldo C. Silva Filho[1], Anne Elise Katsurayama[1], Gleison Santos[1], Leonardo Murta[2], Ana Regina Rocha[1]

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

[2]Computing Institute – Fluminense Federal University
Rua Passo da Pátria 156, Niterói, RJ - Brasil, ZIP 24210-240
leomurta@ic.uff.br

Abstract. *Software projects reuse aim at providing more quality to the final product and increase productivity. To achieve this goal it is necessary to define a systematic reuse strategy as part of organization daily activities. This paper presents the deployment of software reuse management at COPPE/UFRJ Software Engineering Laboratory, the first organizational unit evaluated at MR-MPS v.1.2 Level E whose Reuse Management Process is ISO/IEC 15504 compliant. We also present the difficulties and the lessons learned during this process institutionalization.*

1. Introduction

Software reuse is based on the assumption that many systems developed are not totally new, they only represents changes in systems developed earlier. Thus, many organizations develop systems based on certain business lines, also known as application domain. According to Frakes (2005), reuse purpose involves improving product quality and productivity during its development. The product quality is assured by the use of assets previously tested, approved and reused in other projects.

The MPS.BR Program [Softex 2006a] was created in 2003 by the Association for Promoting the Brazilian Software Excellence (SOFTEX) aiming to increase software development capabilities of Brazilian small and medium-sized enterprises (SME) and also to enhance their competitive advantages. MPS.BR is based on the international standards ISO/IEC 12207 [ISO 2007] and ISO/IEC 15504 [ISO 2004] and defines a process named Reuse Management at its maturity level E.

At the end of 2007, the Software Quality Area of the COPPE/UFRJ Software Engineering Laboratory (LENS) started the efforts for processes deployment at MR-MPS v.1.2 Level E, and consequently the implementation of reuse aspects related to the Reuse Management Process (GRU) for this level. The Software Quality Area of LENS, which is the organizational unit selected for the assessment, was successfully evaluated in May 2008, and the lessons learned during Reuse Management deployment are presented in this paper.

This paper presents the Reuse Management process deployment at LENS and the lessons learned during the assessment process. The next section describes the MPS Model and its components. Furthermore, from Section 3 to section 6, the paper will deal with the MR-MPS Reuse Management Process (GRU), the LENS Reuse Management Process in agreement with MR-MPS, the lessons learned during the GRU process institutionalization and finally, the overall conclusions.

2. MPS Model

The main goal of MPS.BR Program is to improve the quality of Brazilian software processes and products through the development and dissemination of a Brazilian software process model, named MPS Model, based on software engineering best practices and aligned to Brazilian software industry context. The MPS Model is constituted of three main components: the MPS Reference Model; the MPS Assessment Method; and the MPS Business Model.

The MPS Reference Model (MR-MPS) is documented in the form of three guides: the MPS General Guide [Softex 2006a], the MPS Acquisition [Softex 2006b] and the MPS Implementation Guide [Softex 2006d]. The MPS General Guide provides a general definition of the MPS Model and common definitions to all other guides. The MR-MPS defines seven levels of maturity and establishes expected results and attributes of processes that an organizational unit must attend when undertaking in improvement aiming to reach one of the maturity levels [Softex 2006a]. The MR-MPS maturity levels are: 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 [ISO 2007] 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 SMEs. The possibility of rating companies' maturity considering more levels, not only decreases 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 short time frame. The MR-MPS also defines process attributes (PA) based on the ISO/IEC 15504-2 process attributes to define capability levels. The MPS Acquisition Guide [Softex 2006b] describes software and service related acquisition process aiming to support organizations that desire to acquire software products or software service-related based on MR-MPS. The MPS Implementation Guide [Softex 2006d] provides information regarding the MR-MPS maturity levels deployment in software organizations, explaining the processes comprised by the MR-MPS and the expected results of these processes. The MPS Assessment Guide [Softex 2006c] describes, among other things, the appraisal method and process that were defined based on the ISO/IEC 15504 standard.

The MPS Business Model (MN-MPS) defines business rules for: (i) training practitioners through MPS official courses, individual examinations and recycling workshops; (ii) implementing the MPS Model by organizations that provide MPS deployment services, (iii) executing process assessments by organizations that provide MPS assessment services; and (iv) organizing groups of enterprises to execute MPS deployment and assessment.

3. Software Reuse Management in MPS Reference Model (MR-MPS)

The MPS Reference Model [Softex, 2007a] defines two reuse related processes: Reuse Management (GRU) at Level E (Partially defined), whose purpose involves managing the reusable assets life cycle; and Reuse Development (DRU) at Level C (Defined), that aims to identify systematic reuse opportunities in the organization and establish a reuse assets development program from application domain engineering.

This paper focus on the expected results for the GRU process, which are: GRU 1: A strategy for assets management is documented, including the definition of reusable asset and the criteria to acceptance, certification, classification, discontinuity and evaluation of reusable assets; GRU 2: A mechanism to reusable assets storage and retrieval is established; GRU 3: (Levels E and D) Data related to reusable assets use are recorded. (Levels C, B and A) Data related to domain assets use are recorded; GRU 4: The reusable assets are periodically maintained, according to defined criteria and changes are monitored throughout their life cycle; and GRU 5: Users of reusable assets are notified about problems, changes, new versions and discontinuity.

4. The Reuse Management Strategy at Software Engineering Laboratory

For meeting the required results of GRU, there was a need to define a strategy consisting of a process of reuse management and some support tools. The first step in building the strategy was the definition of the reusable assets. In the context of the LENS, a reusable asset is an artifact that supports processes execution and which comprises at least one of their work products. So, all software artifacts (process assets, source code or executable) could be considered as assets with potential for reuse.

The management reuse process of LENS contains three macro activities: Providing Reusable Assets, Maintain Reusable Assets and Notify Stakeholders. Each macro activity is composed by sub activities. The Providing Reusable Assets assesses the assets to check whether they meet the requirements to become a reusable asset. It also stores the reusable assets at organizational assets library and classifies them according to their type and potential context of reuse. This macro activity also identifies relevant stakeholders in order to notify about the new item (Figure 1).

The macro activity “Maintain Reusable Assets” consists in identifying improvement opportunities for assets evolution and finding not useful assets that should be discontinued. The Figure 2 shows the possible states of a reusable asset during its life cycle. After each change of the assets, they are evaluated to check if the characteristics that make them reusable assets were preserved. The reusable assets idle for a period exceeding 90 days, are selected for evaluation of discontinuity. The discontinued reusable assets may be changed to increase its potential for reuse and it can become available for organizational unit again.

When the status of the asset is changed, it is necessary to communicate the relevant stakeholders what had happened to avoid inappropriate reuse of the asset or to make a process activity execution easier. This communications are treated on the scope of the “Notify Stakeholders” macro activity.

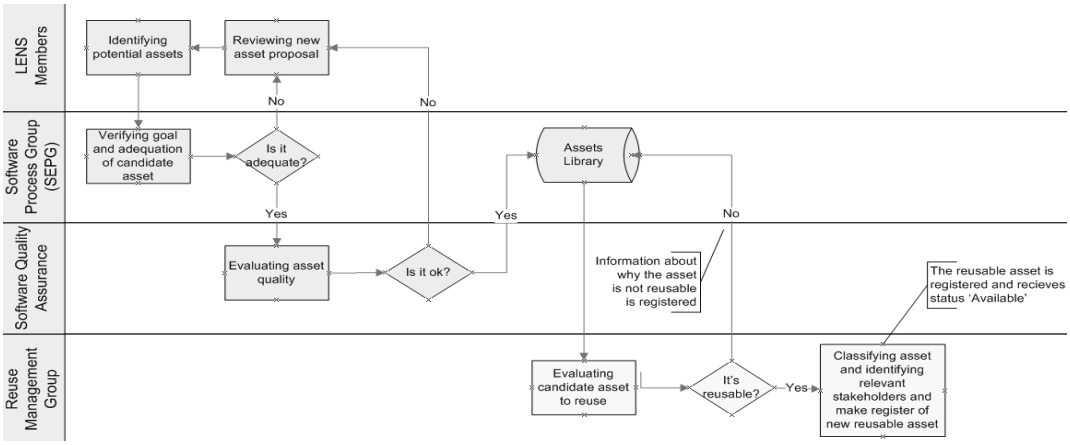

Figure 1. Providing Reusable Assets macro activity

To minimize the effort for reusing, a solution based on combination of tools available at TABA Workstation[1] was outlined. The organizational assets library was used to storage and to classify the assets. The GConf[2] was used to manage changes in reusable assets and assign a new version number on each evolution. The instantiated environments from TABA Workstation were used to provide the reusable assets at the point where the development process prescribes their use.

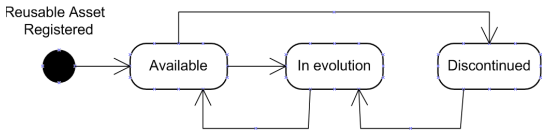

Figure 2. Reusable asset status diagram

The two main difficulties for implementation of the GRU process at LENS were: (i) the definition of a non-intrusive strategy, and (ii) the identification of useful metrics to monitor and control the process. The search for reusable assets was considered the most critical activity for definition of a non-intrusive strategy and for minimizing the organizational cost and effort. The monitoring and controlling of the process was carried out satisfactorily by the use of some indicators. For balancing the usefulness of indicators with the associated cost of measurement, the Management Reuse Group selected only two indicators: rate of reuse of assets and rate of evolution of reusable assets. Some basic measures that compose these indicators can be seen in Figure 3.

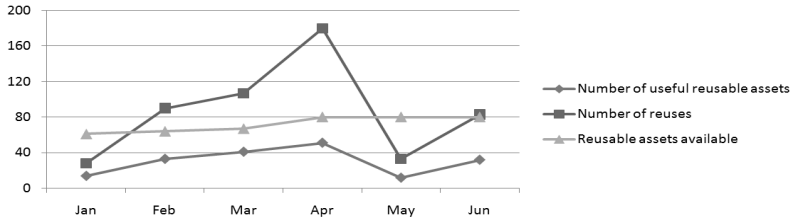

Figure 3. Basic measures of reuse management process

[1] The TABA Workstation is a Process-centered Software Engineering Environment developed by COPPE/UFRJ [Montoni *et al*. 2006].
[2] GConf is a configuration management tool embedded in TABA Workstation.

5. Lessons Learned

The lessons learned during the Reuse Management process implementation were: (i) The identification of critical software processes activities helps to define effective institutionalization strategies; (ii) The definition of a reuse management focus such as minimization of projects cost and effort, can guide to the prioritization of software process improvements; (iii) Systematic reuse fosters software process improvement through reusable assets improvement suggestions; (iv) The more mature the process, the clearer the perception on how it can be automated. In fact, the automation of a preliminary process version may lead to important resources wasting; (v) It is possible to use different technologies to achieve MR-MPS Reuse Management process but to ensure process full adherence in an effective way, some activities like "Assets Evaluation" and "Assets Classification" and "Identifying Relevant Stakeholders" may have to be done manually.

6. Conclusions

The Reuse Management process presented in this paper was appraised with other processes at Level E MR-MPS in May 2008 as part of the Software Quality Area of the COPPE/UFRJ Software Engineering Laboratory (LENS), and was considered fully adhering to the model. In the final assessment, two improvement opportunities were suggested by the appraisal team, in which, both of them are related to automation of reuse management activities held manually, that may lead to errors: (i) automating the reusable assets counting and (ii) developing an automatic mechanism of reusable assets status notification.

A new software process improvement cycle at LENS is currently being defined aiming to achieve MR-MPS Level C. The lessons learned from the deployment of this process will be used to improve GRU process deployment at LENS and will be considered during the Reuse Development Process (DRU) definition.

References

Frakes, W. B., Kang, K., 2005, "Software Reuse Research: Status and Future", *IEEE Transactions on Software Engineering*, v. 31, n. 7, pp. 529-536.

ISO/IEC 12207, 2000, "Information Technology – Software Life Cycle Processes", ISO/IEC.

ISO/IEC 15504, 2004, "Information Technology – Software Process Assessment", Parts 1-5, ISO/IEC.

Montoni, M. *et al*., 2006, Taba Workstation: Supporting Software Process Deployment Based on CMMI and MR-MPS.BR. 7th PROFES, Springer Verlag, pp. 249-262

Softex, 2007a, "MPS.BR: Melhoria de Processo do Software Brasileiro", Guia Geral v. 1.2, Disponível em: http://www.softex.br/mpsbr.

Softex, 2007b, "MPS.BR: Melhoria de Processo do Software Brasileiro", Guia de Aquisição v. 1.1, Disponível em: http://www.softex.br/mpsbr.

Softex, 2007c, "MPS.BR: Melhoria de Processo do Software Brasileiro", Guia de Avaliação v. 1.0, Disponível em: http://www.softex.br/mpsbr.

Softex, 2007d, "MPS.BR: Melhoria de Processo do Software Brasileiro", Guia de Implementação Partes 1-5 v. 1.0, Disponível em: http://www.softex.br/mpsbr.