Tutelkán Implementation Process: Adapting a Reusable Reference Software Process in the Chilean Software Industry

Julio Ariel Hurtado Alegría[1,2], M. Cecilia Bastarrica[1]

[1] MaTE Group. Departamento de Ciencias de la Computación, Universidad de Chile,
Avenida Blanco Encalada, Santiago, Chile.
{jhurtado, cecilia}@dcc.uchile.cl
[2] IDIS Group. Departamento de Sistemas, Universidad del Cauca,
Edificio de Ingenierías – Sector Tulcán, Popayán, Colombia
ahurtado@unicauca.edu.co

Abstract. Reaching quality and productivity in a competitive and dynamic software industry requires defining suitable software processes. Usually, software organizations define their processes using the specification of recognized software process models and standards. Because these process models and standards include only specifications to be implemented, each organization implements them according to its own interpretation and specific needs. Due to both, high costs and the time it consumes, to define a process for a small and medium enterprise (SME) results almost unfeasible. However, the cost and time can be reduced if a community is created for sharing resources in a software process improvement (SPI) initiative. The Tutelkan project is a communitarian strategy of SPI in the SMEs Chilean software industry where resources are shared for the design and adaptation of reusable process assets. This paper presents and evaluates the Tutelkan Implementation Process (TIP) approach, i.e. the process defined for adapting and adopting the reference process. For the evaluation, a measurement framework has been created and applied to a set of SPI projects applying the TIP. The framework has made it possible to evaluate the reusability and adaptability of the reference process, as well as the efficiency and effectively of the entire approach.

Keywords: software process model, software process definition, process assets reuse.

1 Introduction

Organizations looking for a defined software process frequently use reference (or known) processes as a basis. A reference process can be reused and adapted every time it is used in a SPI initiative. These complex activities of tailoring and instantiation require specialized knowledge for filtering, reusing and adapting process elements. Software organizations can apply directly reference processes (without any tailoring at all) or tailored in an inadequate way as a consequence of the inherent complexity of the adaptation process and expertise required for it. This situation increases the costs and risks of development projects. On the other hand, if the organizational software process

is not defined reusing and adapting a reference process, the organization could be wasting the opportunity of reducing implementation effort and cost.

Tutelkan is a government funded project that intends to create a sustainable mechanism for allowing Chilean SMEs to define and document their development software processes towards a further ISO 9001 or CMMI assessment based on the reuse of software process assets. The Tutelkan project's main goal is to create a public reference process (TRP), an implementation process (TIP), an active community and a supporting platform.

This paper presents the TIP and evaluates both, implementation process and reference process. The TIP is a value-oriented, risk-managed and reuse-based implementation process. The evaluation is realized in terms of efficiency, effectiveness, reusability and adaptability capabilities.

The evaluation is supported in a measurement framework presented in this paper, too. The measurement framework has been applied to a set of the Chilean SMEs and its results are also presented. We found that the TIP yields appropriate processes for the SMEs in general, and the adaptation and adoption process is less traumatic because the steps are better understood. However, some trends and questions still arise and they have been a motivation for refining the approach for software process definition based on the process family concept.

The paper is structured as follows. Section 2 describes some related work. The Tutelkan Implementation Process is presented in Section 3. The application of the validation framework is presented in Section 4. Finally, Section 5 presents some conclusions and discusses future work.

2 Related Work

Sutton & Osterweil [2] work on the observation that the Booch Object Orientation Design (BOOD), a framework for describing different specific processes instead of just one unique specific process. Following a process family approach, BOOD was used by the Programmable Design Process (PDP) project [3] for generating a variety of processes. The PDP project used a combination of mechanisms for selecting, specifying and tailoring. Rather than to show the process adaptability, this work has raised key questions for this research line such as, how process family members can be differentiated, or if one member can be tailored for becoming another one.

A configuration management approach is proposed by Belkhair & Estublier [1] for software process reuse and configuration. The idea is based on using a combination of an object-oriented view and the concept of process interface. For explaining this approach, the authors use the Process Interconnection Language (PIL), which was built for process reuse and composition. The language provides version management characteristics for controlling the specific unit versions within a particular software process. The complete software process model is considered as a configuration, which can be automatically generated and may support multiple configurations simultaneously. The most relevant contributions of Belkahir's work are suggesting that the modeling language must be separated from the composition language, and that the configuration management can support process reuse and configuration at large scale. For theses reasons, the model

defines a configuration model and includes the process unit concept (interface and implementation), the family concept (interface and variant group) and the configuration concept. The model makes a separation in three levels: metaclass, family and instance. This work addresses the technical problem about the control and management process models as a family; however, this work does not offer a specific methodological approach for defining specific processes. This work neither makes emphasis on the way of adapting process models to specific contexts, nor in measuring the process adaptability.

If we take a practical look to some free and commercial process models, as well as strategies for their adaptation, they can be considered as useful concrete sources of information for this work. This evidence is described in [4]. One strategy that is followed by several processes consists of defining a configurable process framework that covers all potentially necessary activities. From these activities, a subset is selected for building a project specific process; this is the case of the Rational Unified Process [8]. Another strategy consists of defining a set of process templates for each kind of project and to select the most suitable template for the project at hand; this is the case of Crystal Methodologies [9]. This framework strategy presents understandability problems and an overloaded model. The template strategy presents the inherent difficulty of defining the adequate set of templates for the satisfaction of different types of projects. In this latter strategy, reuse is partially considered, but adaptation is not.

Simidchieva et al. [5] present an approach for defining a process and a set of variants as a process family. This work proposes a formal approach for defining process families by characterizing them. They use the language Little-Jil for implementing an example. The purpose of this work is showing that the language and the approach are appropriate for defining process families. The key elements for process family management lie on the language used that allows a separation of concerns (specification coordination, agents and artifacts), visual representation, and an experimental platform for modifying the process allowing variants. Generating variants is achieved by the technique of process components reuse and combination, based on the user specifications.

Rombach proposes the Software Process Line (SPrL) concept as a systematic mechanism for managing a process and its variants [6]. This work is a motivation for work on process families because it strengthens the relation between software product lines (SPL) and software process lines stated by Sutter & Osterweil. However, the main contribution of his work is about establishing possible further work for SPrL. The work highlights that PMLs must include variability management mechanisms, and more effective methods for creating empirically supported processes as evidence-based software engineering methods or value-oriented software engineering methods; it also includes theoretical and engineering foundations for an integrated vision of SPrL and software product lines. Rombach's work has generated great interest, but it neither includes a particular approach for modeling SPrLs, nor includes a concrete meta-process.

In [7], the tailoring of a software process based on the SPrL concept is presented. The method uses a SPrL in a specific domain (automobile industry), it follows a top-down adaptation approach, and it supports a bottom-up refinement of the generic process based on the trace of the instantiated processes. The work shows through an informal validation study that tailoring, within the SPrL approach, is efficient and it yields adherence to a generic process at adaptation time.

The Tutelkán project presents key evidence of the effectiveness of defining organizational processes by tailoring a references process. The TIP is a process implementation project based on SPI, process asset evaluation, reuse and adaptation. So, Tutelkán offers practical evidence on process model reuse, defined as TRP, and the efficiency and effectiveness of the production plan, defined as TIP. This experience has been carried out on a set of small companies through the application of a measurement framework. The reuse is within a SPI community, instead of a specific process tailored to each project type.

3 Tutelkan Implementation Process

The main goal of Tutelkán is facilitating CMMI implementation. It is a Chilean initiative for reaching a competitive software industry, particularly the SME industry. The main project strategy is defining organizational software processes from a reference process. A reference process is reused and adapted through an implementation process. The implementation project is realized using a systematic process called Tutelkán Implementation Process (TIP). Fig. 1 presents its main elements.

The Tutelkán Implementation Process is a reuse based Software Process Improvement model. Its main characteristics are:

1. It is value oriented. TIP must be instantiated as an institutionalized project with an agenda based on business needs. So, activities must be assigned a priority according to business needs. These needs are covered through incremental iterations organized according to the assigned priorities.
2. It is a reuse based SPI strategy. The process assets are reused, tailored and applied from TRP to the organizations and adopted via pilot projects. However, the model promotes the respect of the original software process (implicit or explicit) as a result of its culture and evolution.
3. It is organizational learning oriented more than human resource control. Although the process is based on disciplines, these are a reference framework and learning is specified as a key discipline.

TIP has been used for implementing CMMI reusing the Tutelkán Reference Process, a SME-oriented reference software process. This experience is about a set of Chilean SMEs.

3.1 TIP Life cycle

TIP is a value oriented, iterative e incremental implementation process. A project TIP is organized from a management and a technical perspectives. A management perspective describes the process as a set of general phases: launching, diagnostic, formulation, implementation and closure.

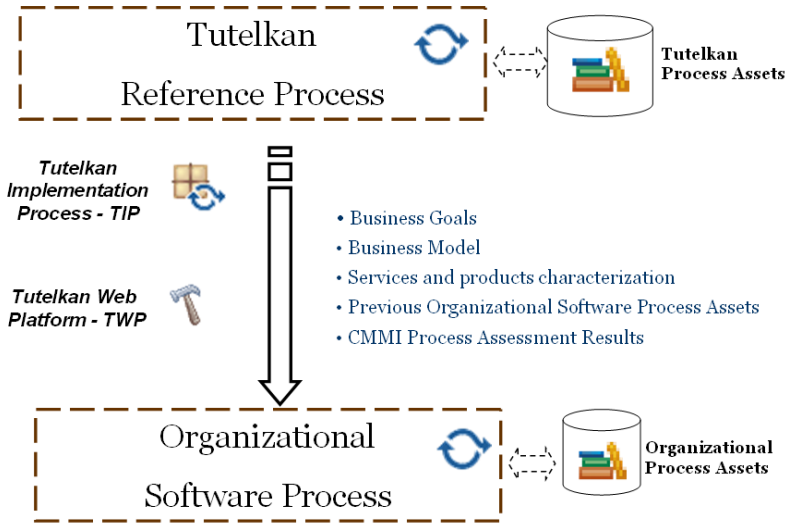

Fig. 1. One or many reference processes are used for defining the organizational software process. Process definition is based on a reuse and adaptation strategy within a value-oriented and managed implementation project.

From a technical perspective the TIP defines a set of disciplines. A discipline is a method unit around a knowledge body (e.g. process evaluation and analysis). TIP defines the disciplines: process evaluation and analysis, process design and adaptation, process adoption, project management, process learning, process training and process configuration management. TIP process structure is showed in Fig. 2.

A TIP iteration is the integration of both perspectives. An iteration is a small implementation project for obtaining early results. A TIP project is divided in a set of managed iterations. A iteration is defined as a set of disciplined activities.

3.2 TIP Phases

A TIP project is performed in five consecutive phases:

Launching: a realizable SPI project is proposed. General goals are defined according to organizational goals. Further, the human infrastructure is defined and the resources required are estimates and fixed.

Diagnostic: the *status quo* of the process is established via a light assessment. Using the SPI goals and the assessment result, reuse opportunities are identified from TRP and implementation needs are defined according to compliment delta (process parts for designing or adapting). The implementation needs are ranked according to business goals.

Formulation: an initial iteration is planned and performed. This first iteration has three goals: to acquire organizational knowledge, to have an early feedback of the project for decreasing the risk level, and to serve as a basis for estimating the following

iterations. A first iteration includes the adoption of the new process in pilot development projects.

Implementation: A set of iterations is planed and performed for reaching a specific implementation objective. This objective is usually associated to a set of implementations needs identified in the diagnostic phase. So, iterations facilitate a natural and incremental implementation of the software process.

Closure: the project is analyzed using the collected information along the project. Specifically learned lessons and implementation results are used for evaluating the SPI project and the funds for a following project. This is a key phase for guarantying a continuous SPI program in the organization.

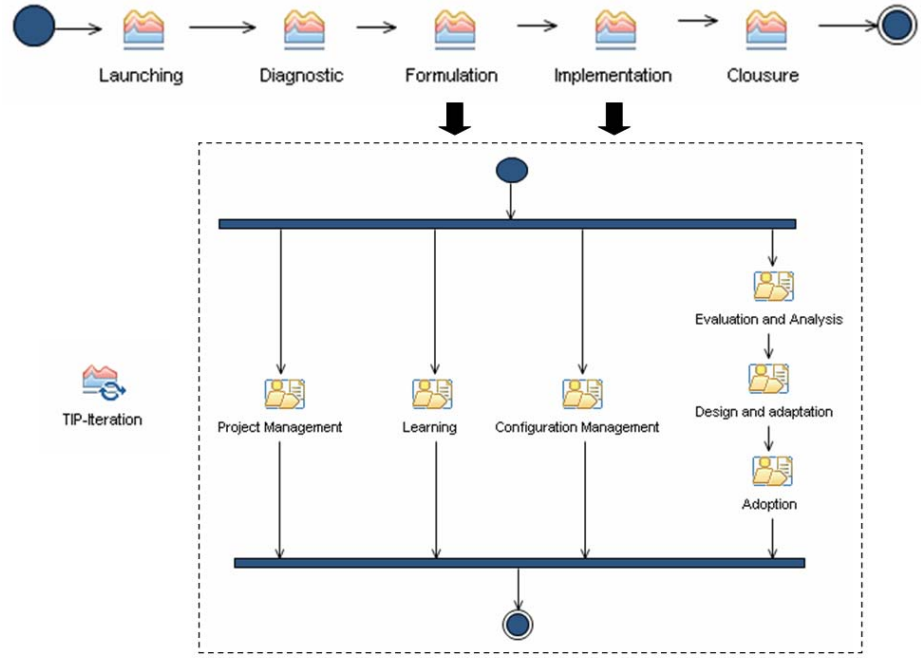

Fig. 2. Life Cycle Structure of a TIP Project – Formulation and implementation phases have a similar process substructure.

3.3 TIP Base Iteration

A base iteration in the TIP is a repetitive walking path in the implementation project. This basic iteration from a technical perspective includes: evaluation and analysis, design and adaptation, and adoption. Also, from a management perspective it includes: project management, learning, software process configuration management and software process training.

Evaluation and analysis: the *status quo* is defined for each iteration taking care of the implementation objectives. For example, if the Project Planning process area of CMMI will be implemented in this iteration, then a deep evaluation must be realized for

establishing the implementation needs. This normally matches with a process area specific assessment. More detail reuse opportunities are also identified.

Design and adaptation: this process area is defined according to implementation needs. The opportunities for reuse are analyzed and incorporated to the process via adaptations. Design and adaptation are realized manually in a meeting where the process technology team (PTT), the process area leader and some process area users reach a consensus about the way the organization works, and they design a set of process assets CMMI compliant. If a reuse opportunity is identified, it is also used; in this case the reuse opportunity is associated to the adaptation.

Adoption: the adoption corresponds to a planned application of the coherent set of process implementations for improving the new process assets designed and adapted in this iteration. The adherence to the organization must be evaluated before the new process assets are incorporated to the organizational software process.

Management: the TIP is value-oriented; and as such it must be planned and tracked. A general plan is realized for easing management, and it is based on phases. Each iteration is planed at the beginning according to previous iterations and the implementation needs considered for the iteration. Tracking is performed in management meetings where the plan information is compared with actual data.

Learning: learning is a discipline scattered along the whole project via learning lessons. A learning lesson is proposed, analyzed, resolved and tested in the implementation project. In the closure phase the lessons are obtained, organized and incorporated to organizational knowledge.

Software Process Configuration Management: the TIP is an iterative and incremental process and the software process is in constant evolution, so configuration management is required. Further, the adoption is continuously applied to different projects with different versions of the same software process.

Software Process Training: a deliverable software process must be created for facilitating the comprehension of itself. The deliverable process must include training material, provide examples and define process training strategy.

3.4 TIP Teams

For supporting the human effort, the TIP defines two teams. One team is oriented to management concerns: the Process Management Team - PMT. This team must be integrated by organizational management personnel with decision capability and by the project leader. The other team is the Process Technological Team – PTT conformed by the process leader, consultant personnel and the process area user leaders.

The PMT must meet monthly and the PTT weekly. The user process area and user leaders participation is determined by the implementations needs covered in the meeting.

4 Validation

The TIP was applied in four software development enterprises. For each software process measured the weighted number of reused elements. For each particular process generated by instantiating the reference software process, the specific effort required was

measured as well as the required adjustments; the more adjustments required the less adequate the derivation process is. So, reuse, adaptation effort and adjustment effort are the basic measurements used for evaluating adaptability of the reference process model and efficiency and effectiveness of the adaptation process.

4.1 Context of application

The TRP is a reference process based on the RUP [12] and it is oriented to the development of software projects where some parts are developed by software factories. Even though the TRP was not built with a process line concept in mind, it has a kernel of CMMI process areas suitable to reuse and adaptation.

For this experience three enterprises were chosen. Nectia and Angecom develop information systems, so these enterprises have a business context similar to that of the original TRP. On the other hand, Rastreosat is devoted to tracking services, and they have developed their own tools for supporting its services. So the nature of the enterprises is diverse.

The TIP model was not applied exactly in the same way in each pilot project because each enterprise had its particular implementation needs and the TIP was still in evolution. Nectia and Angecom used iterations with the adoption discipline realized in a new phase: Adoption Phase. On the other hand Rastreosat implementrf a hybrid approach, where some iterations included adoption; and a more refined and disciplined TIP.

4.2 Measurement Framework

A measurement framework is required for objectively measuring the TIP's adaptability, efficiency and effectiveness. This paper proposes a measurement framework. The metrics have been separated in two groups: direct metrics and derived metrics. The direct metrics can be calculated from direct measurements on the process model, and derived metrics are calculated from direct metrics. The derived metrics correspond to the evaluation characteristics.

Direct Metric	***Definition***	***Measurements***	***Applied to***
Reuse Factor - RF	Weighed reused model elements respect to total process model elements.	Quantity of reused elements with respect total elements – QRE	Reused elements at different granularity levels
		Granularity level where reuse is applied- RAL	
		Reuse frequency: number of times that a reused element is referenced - RF	
Adaptation Effort Factor - AEF	Adaptation effort with respect to total effort	Adaptation effort- AF	Total adaptation
		Total effort - TE	
Adjustment Effort Factor - AjEF	Adjustment effort with respect to total effort	Adjustment effort - AjE	Total adjusts
		Total effort - TE	

Table 1. Adaptation and Reuse Direct Metrics on Process Models.

Derived Metric	***Definition***	***Derived as***
Adaptability - A	Adaptability of reusable process model	$A = RF * AEF$
Adaptation Efficiency – AE	Proximity to ideal adaptation effort	$AEc = AEF+AjEF$
Adaptation Effectiveness – AEv	Proximity to ideal adaptation	AEv= 1-AjEF/AEF

Table 2. Adaptation and Reuse Derived Metrics on Process Models.

4.3 Framework Application

The measurement framework was applied to TRP implementation via the TIP process and it was tailored according to the following considerations:

- Two questions about the TRP and the TIP were initially defined. How much an organization can reuse from TRP? How much did the implementation effort decreased? Provided that these are general questions, the measurement framework was tailored so that an objective response could be obtained.
- Three granularity levels were defined and weighted:
    1. Basic process element: artifact, role, task
    2. CMMI recommended practice (workflow that implements the practice) or TRP practice
    3. CMMI process area (workflow that implements the process area)
- Reuse frequency was fixed to 1
- The measurement framework was only applied to model elements. Template adaptation and adjusts was not considered.

Once the framework was tailored, direct measurements were obtained from the three finished TIP projects. The process model measurement was done manually, counting process model elements. The effort was obtained from registered time and the management data related with the resources.

4.4 Results

The framework was applied to three SPI projects over four CMMI process areas: requirements management, project planning, configuration management and product and process quality assurance. The results showed Fig 3 evidence the reuse and adaptability capabilities of the TRP and the effectiveness and efficiency of the TIP based on TRP.

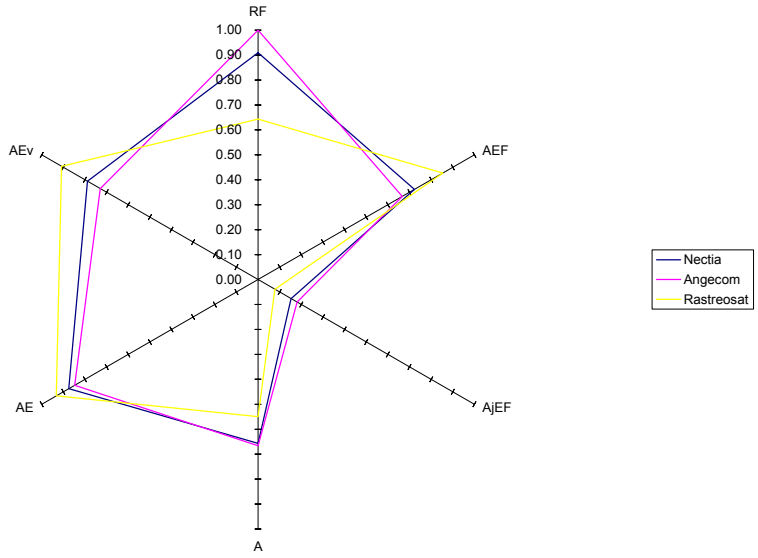

Fig 3. Application of measurement framework to three SPI projects in Tutelkán

The enterprise where the TRP was more reused was Angecom, but this enterprise had the lowest effectiveness and efficiency. On the other hand, the TRP was less reused in Rastreosat, but this enterprise has the highest effectiveness and efficiency. Nectia and Agecom showed similar behavior in general, and the more similar result was in the adaptability of the TRP using the TIP. The TRP using the TIP is more stable respect to adaptability near to 0.62.

4.5 Analysis

The results must be analyzed according to the specific context of each enterprise. The analysis has been separated in two parts because of the similarity of the context.

Agecom and Nectica: reuse and adaptation was high but effectiveness and efficiency was low. In this case the organizational contexts were similar to that of the reference context. However, adaptation errors increased the need for adjustments. So, the TIP had poor results.

Rastreosat: reuse and adaptation was low, but effectiveness and efficiency was high. In this case the organizational context was different to that of the reference context. However, adaptation errors were identified early and removed. Although the TRP was not suitable for the organization and the adaptation effort was high, the resources used and the objectives reached were better than in the first case.

In general these cases show two relevant aspects:

- The reference process context and organizational context must be similar for achieving high reuse and adaptability measurements. So, a process reference scope is necessary for defining the reference process and for evaluating the benefits of implementing the reference process instead of implementing a particular process from scratch or based on another reference process.
- The tailoring of the reference process is as relevant as the reference process itself. Even if the reference process results suitable, an inadequate tailoring could negatively affect the SPI project. Additionally the tailoring could result useful even when the reference process is not suitable for the organizational context.

5 Conclusions and Further Work

We found that it is feasible to reuse and adapt reference processes or process frameworks deriving particular process models for different organizations. We have presented the TIP as an approach for adapting the Tutelkán reference process. The results showed that the TIP increases the effectiveness and efficiency of the adaptation process independently of the reuse and adaptation capability of reference process. But as reuse is a key factor for decreasing the general implementation effort, an enterprise whose organizational context is similar to that of the reference process context and using the TIP process will reach most benefits.

However, a systematic approach is required for defining processes that could be reused across organizations, and for tailoring time because:

- A reference scope must be established for facilitating the decision of what kind of organizational context results suitable to adopt it.
- A reference process must be provided for reusing. This implies, to satisfy the scope and to provide mechanisms such as commonalities and variabilities for facilitating and systematizing the adaptation.
- A tailoring plan must be defined for facilitating the tailoring. This plan must be defined as part of a SPI project inside an adopter organization.

The TRP was not developed with this approach, so the reuse and adaptation processes were done by searching-copying-pasting-understanding-adapting. However, this manual adaptation was managed as an implementation process. Reuse opportunities were identified during evaluation and they were incorporated to the organizations when it was applicable.

In order to achieve adaptable process models, it is necessary to count on an integrated approach that defines methodological aspects in order to support software process engineering taking advantage of the technological and managerial models developed within software product engineering. Our further work is oriented to show the benefits of a meta-process for systematically defining, tailoring, and applying adaptable software process models [11]. This systematic meta-process is based on the idea that a standard software process and its derived processes can be built, applied and managed as a software process line (SPrL) using MDE concept in the process context [12]. We will consider a SPrL as a set of software process models that share a common, managed set

of goals and that satisfies the needs of a particular organization, and that are built using a set of available software process assets. The MDA [10] technical solution is being used as a strategy for separating concerns at different abstraction levels and SPEM2.O [13] and required extensions as the process definition language.

References

1. Belkhatir, N., Estublier, J. Supporting reuse and configuration for large scale software process models. In Software Process Workshop, 1996. Process Support of Software Product Lines., Proceedings of the 10th International, pages 35-39 (1996)
2. Sutton S. Jr. and Osterweil L. Jr. PDP: Programming a Programmable Design Process. In Proceedings of the 8th International Workshop on Software Specification and Design, pages 186_190, March 1996.
3. Sutton S.Jr. and Osterweil, L. Product families and process families. In ISPW '96: Proceedings of the 10th International Software Process Workshop, page 109, Washington, DC, USA, 1996. IEEE Computer Society.
4. Bustard, D., Keenan, F. Strategies for systems analysis: Groundwork for process tailoring. In ECBS '05: Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05), pages 357-362, Washington, DC, USA, 2005. IEEE Computer Society.
5. Simidchieva B., Clarke, L., and Osterweil L.J.. Representing process variation with a process family. In Qing Wang, Dietmar Pfahl, and David M. Raffo, editors, ICSP, volume 4470 of Lecture Notes in Computer Science, pages 109-120. Springer, 2007.
6. Rombach, H. D.. Integrated software process and product lines. In International Software Process Workshop, SPW 2005, Beijing, China, pages 83-90, 2005.
7. Jaufman, O and Münch, J. Acquisition of a project-specific process. In Product Focused Software Process Improvement, PROFES, pages 328-342, Oulu, Finland, 2005. Lecture Notes in Computer Science, Springer.
8. Kruchten, P. The Rational Unified Process, An Introduction. Addison Wesley. 1998.
9. Cockburn, A. Agile Software Development. Adisson-Wesley. 2002.
10. OMG. MDA Guide Version 1.0.1. OMG Specification omg/2003-06-01, Object Management Group, 2003.
11. Hurtado, J., Bastarrica, Ma. C. PhD Thesis Proposal: A meta-process for defining and applying adaptable and reusable software processes. Computer Science Department, Universidad de Chile. 2008.
12. Bézivin, J., Breton, E. Applying the basic principles of model engineering to the field of process engineering. Upgrade, V(5):27□33, October 2004.
13. OMG. Software Process Engineering Metamodel SPEM 2.0 OMG Beta Specification. Technical Report ptc/07-11-01, OMG, 2007.