RADIP: A Software Development Process for Mexican PyMEs

Arturo Barajas Saavedra, Francisco Javier Álvarez Rodríguez,
Jaime Muñoz Arteaga, Juan Muñoz López

Universidad Autónoma de Aguascalientes, Aguascalientes, Aguascalientes, México
{abarajas, fjalvar, jmunozar, jmunoz}@correo.uaa.mx

Abstract. The actual software development models that employ methods and processes are designed to focus on the definition of the software development processes in large scale companies. In the case of Mexico (a country in which almost 80% of the software industry is composed of small and medium scale companies) the penetration of such models have being quite minimal. In this research, a software development process has been designed, based on the CMMI maturity level 2 in its staged representation, to offer the PyMEs (from Spanish, Pequeñas y Medianas Empresas) a reference framework for building systems and implementing software engineering techniques. This process was designed by identifying the set of better practices that would permit PyMEs to construct a high quality software product. Also, this process enabled 14 development teams to reach a CMMI maturity level 3.

1. Introduction

Presently the majority of countries employ software systems that are faster, more potent, complex and still cheaper. This forces software development companies to optimize the production of their software (Reyes Delgado, 2005), therefore it is necessary to implement advanced software engineering techniques to generate and define detailed processes that permit the production of more advanced, detailed and innovative designs.

The software development methods and the defined processes get a lot of attention in research centers but are rarely used in the software industry. One of the main reasons is that very little is known about the integration, interpretation, definition and adoption of methods in the software development processes and, the exact role that the methods and processes play in the life cycle of software is still very diffused (Plat & J. Toetenel, 1992).

According to (Sommerville, 2002), the software processes that are perfectly adapted to the necessities of all businesses and their software projects don't exist but due to wide research and experience gained, better practices for systems development have been proposed.

Although all the efforts made in research centers, as (Plat & J. Toetenel, 1992) mentions, on a worldwide basis, there exists a considerable percentage of software

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 311-322

Received 24/03/08
Accepted 26/04/08
Final version 03/05/08

companies that don't use methods for the development of their projects and processes. This problem can be attributed, along with the lack of knowledge of integration and the life cycle of methods (Plat & J. Toetenel, 1992), to the fact that a lot of managers or software developers ignore the potential benefits of using methods and defining software processes for product quality as well as process quality and subsequently project quality (Chatzoglou, 1997).

In order to produce a successful software system, i.e., one that satisfies the needs of the client, without errors, that's easy to modify and use (Pressman, 2001), it's necessary to establish and adopt a disciplined development approach.

Even though the software industry and software developing companies are aware of the necessity of the establishment of disciplined approaches for the development of software systems, the discussions continues about which are the best practices to implement within processes.

2. Problem Outline

In the creation of a software product with more than one minor complexity, the only viable way to ensure its quality is guaranteeing production through a high quality development process (Chroust, 1996).

In accordance with (Fuggetta, 1995), a software process is the group of persons, organizational structures, rules, policies, activities and proceedings, software components, methodologies, and tools used or created specifically to conceptualize, develop, offer a service, innovate and expand a software product.

Matching what was previously said, (Oktaba & Ibargüengoitia González, 1998) propose the static view of a software process (**Fig. 1**). In this view it's established that a software process is a composition of stages, activities and resources.

As was already mentioned, the software development processes are complex and are based on human criteria in order to judge and create (Pressman, 2001). Due to these characteristics it is very difficult to automate processes in software development. There exist other reasons for which the automation of a process in software development is not reached (Sommerville, 2002), one of them being that there exist multiple approaches for the development of a software system.

Focusing on the organization's processes allows the increase of company capacity for competition, bettering the administration of all available resources (Harrington, 1998). Also, by focusing in the processes helps the organization to reduce redundant work and improves job organization, making easier the project administration. La process management should be supported by training, budgets, skillful workers, tools, and managers support (Kulpa & Johnson, 2003).

The quality of the process has an important influence on the superiority of software given that its administration comprehends from the standard definition of processes to the elaboration of reports (Sommerville, 2002). In simpler terms, "it's the grade at which a client or user perceives that the software agrees with his needs" (Chroust, 1996).

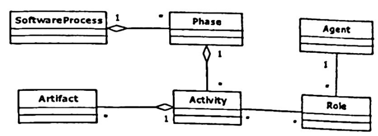

Fig. 1. Software process static view (Oktaba & Ibargüengoitia González, 1998).

As an alternative solution to these deficiencies, there exist different standards and development models as well as quality models that are created by individuals or organizations to harmonize the specifications of products, interfaces, processes, terminology, etc. (Margain Fuentes & Durón Rosales, 2002). Some standards are formally developed by organizations whilst others impose the market (SPICE, 1998). Some of these models and methodologies are namely, amongst others, RUP (Rational _Unified Process) from IBM, CMMI (Capability Maturity Model Integration) from SEI, SPICE (Software Process Improvement and Capability dEtermination) from SQI, MoProSoft (Modelo de Procesos para la Industria de Software) from ProSoft, PSP (Personal Software Process) and TSP (Team Software Process), both from SEI.

Sadly, these models are designed to focus on the definition of software development processes on large scale companies, this being the reason why in Mexico, country in which almost 80% of the software industry is made up of small and medium businesses (PyMEs) (Silva Alarcón, 2004), the penetration and adoption of these models has been quite minimal. Due to this, there is a lack of reference frameworks and models to support the PyMEs' implementation of development process that help them to guarantee the quality of their products.

3. RADIP, a Software Development Process for PyMEs

In this research, a software development process has been designed based on the CMMI maturity level 2 in its staged representation so that the PyMEs have access to a initial reference framework in aid of constructing systems and the implementation of software engineering techniques. This design was based upon the identification of the best set of practices that allow the PyMEs to construct a software product and integration of the best practices identified within the CMMI maturity level 2.

With the development of this process and in general with this research, the following hypothesis has been established and tested: "With the use of a methodology for software development based on the CMMI maturity level 2, that establishes a clear and precise way the set of phases, processes, activities and artifacts, a process for software development can be recognized to generate quality products in a PyME".

In the case of the development and mapping out of the proposed process in this research and the CMMI model, the maturity model of CMMI that has been chosen is the one in its most complete version, CMMI-SE/SW/IPPD/SS (CMMI for Systems

Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing).

The main objective of the CMMI maturity level 2 is that in the projects exist a requirements management process and plan, execute, measure and control the processes.

3.1. The Software Development Process RADIP

The software development process RADIP covers five principal phases and is supported by the use of Personal Software Process (PSP) and Team Software Process (TSP). The general aspect of the process and its main inputs and products are shown in **Fig. 2**.

The RADIP process is aided in its Administration process by the TSP for the general management of the development teams and the establishment of the planning of activities to be carried out. This is due to the fact that TSP offers some important benefits for development: TSP is a predefined process with guides and formats designed by the SEI. TSP allows creating successful projects that reach their goals on time and within an estimated budget. Also the development teams greatly augment their productivity and the quality of software products (IT Institute). Furthermore, PSP supports the process by controlling the individual activity of every engineer in the other processes of RADIP as it permits developers to adopt better practices so that they can establish reliable and precise work plans (IT Institute).

Each process establishes a set of activities to carry out. The products and artifacts that are produced in each process are also defined within each of them. As an additional support, each process has a set of templates to facilitate the adoption of the activities and, doing so, provide a smooth and easy adoption process for the development team.

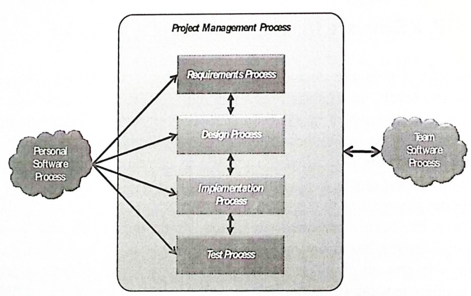

Fig. 2. Software development process main view

3.1.1. Requirements Process

The requirements process (**Fig. 3**) establishes the activities to carry out to be able to conduct the management of the products requirements, project requirements and product components as well as the identification of inconsistencies that may exist within these requirements. (Carnegie Mellon University, SEI, 2002).

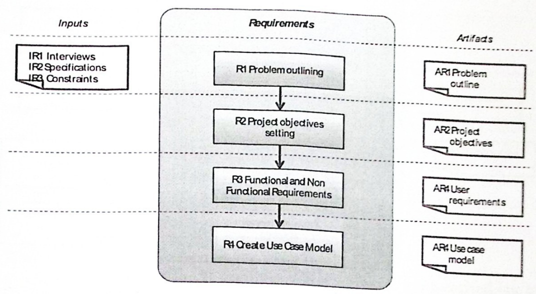

Fig. 3. Requirements process specification

3.1.2. Project management Process

The administration process (**Fig. 4**) establishes the activities to be carried out, so as to be able to comply with the development project management. It starts with the project launch and ends with the elaboration of the project's schedule.

3.1.3. Design Process

The design process (**Fig. 5**) establishes the activities to be realized to be able to complete the conceptual, logical and physical design of the entire project, commencing with an UML model, and in doing so, establish a clear technical solution for the project.

3.1.4. Implementation Process

The implementation process (**Fig. 6**) establishes the activities to carry out the implementation of the technical solution established in the previous process. This is the principal phase supported by PSP, given that in this phase the calculation of development time and defects control is more valuable.

In this process the activity I2 is formed by two sub-activities, I2.1 and I2.2, where the latter generates an artifact AI2 System codes, and activity I2 generates artifact AI3 Defects control.

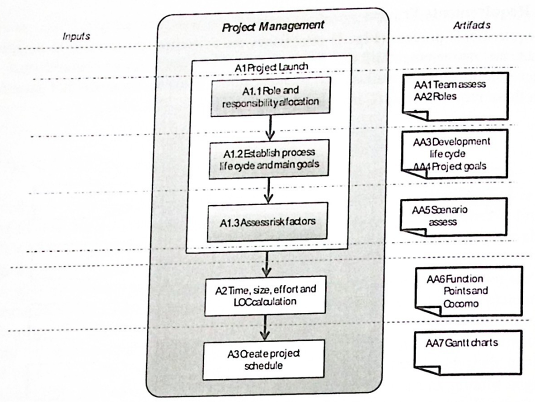

Fig. 4. Project management process specification

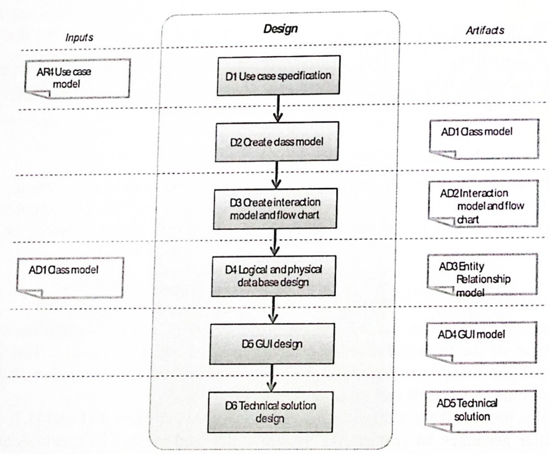

Fig. 5. Design process specification

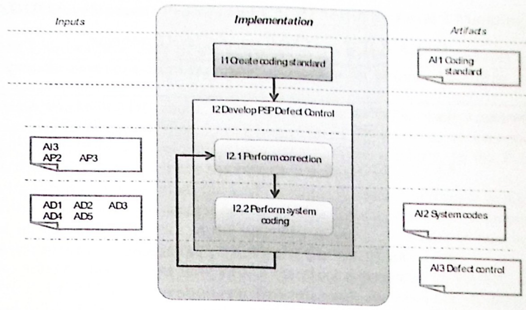

Fig. 6. Implementation process specification

3.1.5. Test Process

Finally, the test process (**Fig. 7**) establishes the activities to be carried out to validate (evaluate) the created product by the implementation process. This process entails activities, which allow the engineers to realize tests in a controlled and clear manner and so guaranties that the product is free of errors.

In this process, it's worth mentioning that the activities P3, P4 and P5 generate only one artifact AP3 Test results. In contrast, the activity P2 has an input (artifact AI2) and produces an artifact AP2 Corrections to be made.

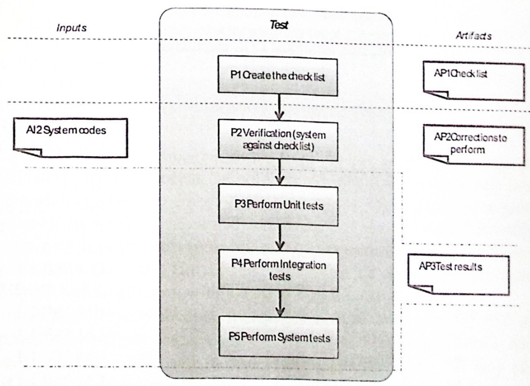

Fig. 7. Test process specification

Table 1. RADIP and CMMI maturity level 2 mapping

Process	ID	Activity	Process area / Specific practice
Requirements	R1	Problem outlining	RM / SP 1.1 PP / SP 1.1 CM / SP 1.1
	R2	Project objectives setting	RM / SP 1.2, SP 1.3 PP / SP 1.1 PMC / SP 1.1, SP 1.6, SP 1.7 CM / SP 1.1
	R3	Functional and Non Functional Requirements	RM / SP 1.1, SP 1.2 PP / SP 1.1 CM / SP 1.1
	R4	Create Use Case Model	RM / SP 1.1, SP 1.2, SP 1.4 CM / SP 1.1 XX / GP 2.2
Project Management	A1	Project Launch	
	A1.1	Role and responsibility allocation	PP / SP 2.5 PMC / SP 1.2 CM / SP 1.1 XX / GP 2.4
	A1.2	Establish process life cycle and main goals	PP / SP 1.3 CM / SP 1.1
	A1.3	Assess risk factors	PP / SP 2.2 PMC / SP 1.3 CM / SP 1.1, SP 1.1 XX / GP 2.8
	A2	Time, size, effort and LOC calculation	PP / SP 1.4 PMC / SP 1.1, SP 1.6, SP 1.7 MA / SP 1.1, SP 1.2, SP 1.3, SP 1.4 CM / SP 1.1, SP 1.3
	A3	Create project schedule	RM / SP 1.3, SP 1.4, SP 1.5 PP / SP 2.1, SP 2.7 PMC / SP 1.1, SP 1.6, SP 1.7 CM / SP 1.1 XX / GP 2.2, GP 2.9
	P2	Verification (system against check list)	CM / SP 1.1
	P3	Perform Unit tests	CM / SP 1.1
	P4	Perform Integration tests	CM / SP 1.1
	P5	Perform System tests	CM / SP 1.1

3.2. RADIP Development Process and CMMI Maturity Level 2 Mapping

The table 1 shows how each of the activities of the RADIP process covers every one of the specific practices from the model CMMI in its Managed maturity level, i.e., Level 2.

The mapping of RADIP and CMMI permits to establish in clear means which specific practices from CMMI model are being covered. By doing so, this mapping guaranty that the proposed development process offers the capacity, to the PyMEs that adopt it, that at least will achieve the Managed maturity level.

It is Important to stress that, the CMMI maturity level 2, although being the first level of the model, it is the hardest one to achieve. This is so because it requires major changes in the work structure of the company in question, which in turn implies, in most cases, a cultural change in said company (Gracia, 2005).

In the table 1, was shown only two of the process proposed in RADIP, this is because the Design, Implementation and Test processes and each one of their activities are mapped to the specific practice 1.1 from the Configuration Management process area in the CMMI model.

In the next table (**Table 2**) is shown the typographical conventions used in the previous table (**Table 1**),

Table 2. Typographical conventions

Acronym	Definition
SP	Specific Practice
RM	Requirements Management
PP	Project Planning
PMC	Project Monitoring and Control
MA	Measurement and Analysis
CM	Configuration Management
XX	Any maturity level 2 process area

4. Results

For the collection of experimental results from the use, adoption and implementation of RADIP development process, 14 software development projects were launched in the course of August to December 2007, the projects where launched in private companies as well as government owned companies and also as internal development for the university. These projects were realized by 9th semester Computer Systems Engineering students from the Universidad Autónoma de Aguascalientes Autonomous University of Aguascalientes, with a total of 60 participants.

The launched projects include web pages design, automation of PSP processes, developments for mobile devices and even billing and inventory control. All the projects were concluded with satisfaction in the established time and they all covered 100% of raised requirements (functional as well as non-functional).

Nevertheless, this is not sufficient data to determine whether or not the RADIP process is a high quality process that will promote the high quality product yield. And so validations of the RADIP process were carried out. The first of these were achieved with a set of interpretations of the CMMI model in its 5 stages of maturity. Through which a set of forms are filled out allowing the determination of the maturity level that the processes of the company or development team has achieved. With this first validation it was verified that the RADIP process permits that the development teams implement and adopt a reference framework for the development and construction of a software product. With this process not only did the development teams achieve the CMMI maturity level 2 for their processes, but they also went on to achieve maturity level 3 with respect to the scaled version of the CMMI. This indicates that RADIP process offers the development teams an initial framework that can naturally evolve to reach higher levers of maturity for development processes.

The second validation was carried out using an automated Web tool (developed by the Universidad Autónoma de Aguascalientes in collaboration with the Universidad Politécnica de Aguascalientes) based on the thesis "Diseño de un Instrumento de Auto-evaluación para Diagnosticar el Estatus de las Organizaciones en México con Respecto al Modelo ProSoft: Proceso de Gestión de Negocios de la Categoría de Alta Dirección". The thesis proposes an auto-evaluation instrument based on MoProSoft and the concept was developed (Reyes Delgado, 2005). Said tool can be found in the following URL of the Universidad Autónoma de Aguascalientes: http://ingsw.ccbas.uaa.mx/AE_MoProsoft/. Using this tool, the *Average Degree of Completion of the Activities per Level* achieved for the Processes Management process was found to be **Medium High**, that is to say, **Satisfactorily Reached**.

Tabla 3. Results interpretation per activity

Section	Activity	Average
Process Planning (PP)	A.1.1	5.51
	A.1.2	5.45
	A.1.3	5.33
	A.1.4	5.33
	A.1.5	5.63
	A.1.6	5.32
	A.1.7	5.90
	A.1.8	5.80
	A.1.9	5.95
	A.1.10	5.10
	A.1.11	5.45
Process Implementation (PI)	A.2.1	5.50
	A.2.2	6.35
	A.2.3	5.70
	A.2.4	6.30
	A.2.5	6.05
	A.2.6	5.20
(PI)	A.2.7	3.80
Process Evaluation and Control (PEC)	A.3.1	5.65
	A.3.2	5.10
	A.3.3	4.75
	A.3.4	5.84
	A.3.5	5.80
	A.3.6	5.85
	A.3.7	4.80
	A.3.8	5.20
	A.3.9	4.80
	A.3.10	5.60
	A.3.11	5.70
	A.3.12	5.00
	A.3.13	5.20
	A.3.14	5.25
	A.3.15	4.60
	A.3.16	4.54

In respect to the *Average Degree of Completion of the Activities per Objective* achieved for the Processes Management process was also found to be **Medium High, i.e., Satisfactorily Reached.** This implies that the RADIP process really supports and is a solid basis for the development and maintenance of a software development process in a PyME. In the next table are shown all the results mentioned above per each activity, and it can be seen that the most of the activities reach a Medium High achievement, which one is comprehended between the values 4 and 6 in the work by Reyes Delgado.

5. Conclusions and future work

Throughout this research it has been proven that with the use of a software development process based on CMMI maturity level 2 in its staged representation, that establishes very clearly and precisely the set of phases, processes, activities and artifacts, a software development process can be made to generate products of quality in a PyME.

Furthermore, the PyME, through the RADIP process, will be able to define a process of greater maturity that, as was validated in this research, allows reaching, at least, a defined process, i.e., a maturity level 3 with respect to CMMI.

It is important to stress that this work is developed within the framework of a research to achieve a Masters Degree in Software Engineering.

As future work there exist the possibility of studying the model for maturity levels Defined, Quantitatively Managed and Optimizing of the CMMI, as well as its implementation in a software development company (PyME).

References

1. Carnegie Mellon University, Software Engineering Institute. (2002). Capability Maturity Model Integration (CMMI), Version 1.1 Staged Representation. Pittsburgh, Pennsylvania, Estados Unidos: Carnegie Mellon University.
2. Chatzoglou, P. D. (1997). Use of Methodologies: an Empirical Analysis of their Impact on the Economics of the Development Process. European Journal of Information Systems , 6, 256-270.
3. Chroust, G. (1996). What is a software process? Journal of Systems Architecture .
4. Fuggetta, A. (1995). Il Processo Software, Aspetti strategici e organizzativi.
5. Gracia, J. (2005). CMM - CMMI Nivel 2. Retrieved Marzo 21, 2008, from Prácticas y métodos para mejorar el desarrollo de Proyectos de Software: http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php
6. Kulpa, M. K., & Johnson, K. A. (2003). Interpreting the CMMI: A Process Improvement Approach. Boca Raton, Florida, Estados Unidos: Auerbach Publications CRC Press LLC.
7. IT Institute. (n.d.). Personal Software Process (Planning and Quality). Retrieved Marzo 23, 2008, from http://it-institute.org/index.php?option=com_content&task=view&id=22&Itemid=39
8. IT Institute. (n.d.). Team Software Process. Retrieved Marzo 23, 2008, from http://it-institute.org/index.php?option=com_content&task=view&id=56&Itemid=39

9. Margain Fuentes, M. d., & Durón Rosales, B. (2002, Octubre). Diseño de una herramienta para la interpretación y evaluación de los procesos de ingeniería del modelo CMMI y su aplicación en el diagnóstico de la industria de software. México.
10. Oktaba, H., & Ibargüengoitia González, G. (1998). Software Process Modeled with Objects: Static View. Computación y Sistemas , 1 (4).
11. Plat, N. v., & J. Toetenel, H. (1992). Application and benefits of formal methods in software development. Software Engineering Journal , 7 (5), 335-346.
12. Pressman, R. S. (2001). Software engineering: a practitioner's approach (Quinta ed.). McGraw-Hill.
13. Reyes Delgado, P. Y. (2005). Diseño de un Instrumento de Autoevaluación para Diagnosticar el Estatus de las Organizaciones en México con Respecto al Modelo ProSoft: Proceso de Gestión de Procesos de la Categoría de Gestión. México.
14. Silva Alarcón, A. (2004). Modelos de calidad: La industria del software en México. México.
15. Sommerville, I. (2002). Software enginnering (Sexta ed.). Pearson Education.
16. SPICE. (1998). The theory and Practice of Software Process Improvement and Capability Determination. (K. El Emam, & J.-N. Drouin, Eds.) IEEE Computer Society Press.