Interpretation of CMMi for the small software industry

Francisco Alvarez R, Jaime Muñoz A, Pedro Cardona¹ and Alfredo Weitzenfeld R.²

¹Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Av. Universidad No.940, Aguascalientes, México.

²Instituto Tecnológico Autónomo de México, Depto. de Ciencias Computacionales, México.

fjalvar@correo.uaa.mx, jmunozar@correo.uaa.mx, jpcardon@correo.uaa.mx
alfredo@itam.mx

Abstract. This paper pretends to show a framework for the interpretation of CMMi (Capability Maturity Model Integrated) with integration of roles, activities and graphic patrons, this model is for the context of small organizations. The motivation of the development of this work is for the lack of software practices documentation that it is a help for the software practices that the companies make in Latin America, also it promotes the use of standard and documented processes that raise the quality of software products developed by the small scale software companies. This interpretation shows itself the application of a practical case.

1. Introduction

A software development process comprises a group of people, organization structures, rules, politics, activities, software components, methodologies and tools specifically used or created to conceptualize develop, serve, innovate or extend a software product [10]. Additionally, a software process is a group of activities and associated results leading to the creation of a software product [18]. Software processes have been designed to support a number of development tasks, from the creation of new products to the extension and modification of existing ones. In general software processes are complex, based in particular organization needs and according to the people involved in them; thus the difficulty in standardizing processes and the limitations in many of the tools designed to automate the development process. Historically, a large number of the employed software processes are based on the very classic waterfall and evolutionary development models [17], or the more recent spiral and win-win models [5][6].

One of the main challenge with respect to software processes is how to improve their capabilities, something best achieved through standardization and precise definition. A related problem involves the definition of factors to consider in the construction of a process.
standard for smaller projects, since these kind of projects have different characteristics from larger ones, including more limited resources and development time [14].

Among the different processes defined to increment the personal productivity, the Personal Software Process (PSP) is a framework helping software engineers measure their personal work in terms of the following aspects: Planning, Design, Coding, Compilation, Tests and Postmortem, in addition to guides that indicate the activities to follow [12]. Since this is a personal process it can be applied to smaller organizations. Team Software Process (TSP) [13], is the solution to the integration of the PSP, however is at development level, since results of the application of this technique have not been reported. TSP is created by the reason that most of the software projects are developed by teams and the necessity of the knowledge to work in team. The TSP is a technology of software development that puts a framework based on work teams with the objectives: Develop products in several cycles, provide metric for teams, evaluate lists and teams, and offer guides for the solution of problems in teams.

One of the objectives in process development is to improve product quality. For IEEE, quality is defined as the degree to which a system, component or process complies with its specified requirements, in addition to the client or user needs and expectations [2]. In the ISO 9000:2000 norm definition, quality poor, good or excellent, is the degree to which a group of inherent characteristics comply with requirements [19].


A frequent conception about adopting Capability Maturity Model Integration (CMMi) is that it works only for large organizations (its cost and complexity appear to make it impractical for smaller organizations to implement) [3].

Some organizations in USA recently had implemented the CMMi, for example Analytical Service Inc. have three process areas (PA’s) of CMMi as part of a pilot study with the Software Engineering Institute (SEI). The organization said CMMi might even be more beneficial to smaller businesses because it allows them grow more consistently and to make changes when they are less costly, that is, “before growth demands them”.

ASI and Cirrus Technology Inc. are two Huntsville, AL, companies that participated in a recent study to develop a business case and technical guidance for small – to medium – sized enterprises- defined by the study as companies with 25 to 250 employees- that wish to adopt CMMi. Initial results from the pilot look promising: both organization described significant benefits from using CMMi, especially in the areas of project management and changes management, and both area in the process of documenting and disseminating their finding so that others can learn from their experiences.

The pilot, launched in July 2003, is part of joint project between the SEI and the U.S. Army Aviation and Missile Research and Development Center (AMRDEC) Software Engineering Directorate in Huntsville. Suz Garcia, a member of the piloting team from the SEI, said that the pilots help support a business case for deploying CMMi in smaller companies.
For maturity level IV and V of staged model CMMi, is represented like a whole mechanism that allows the organization operation that develops software and assures the process definition in spite of technology changes, through the continuous improvement philosophy.

This CMMi interpretation use a role redefinition regarding the deficit of people involved in software projects in small-scale industries of software. Three main roles are proposed that of some way include the complete roles defined in the reference model:

1. Project Leader, who is project person in charge include the next roles: Manager, Senior manager, Project manager, Project software manager, First-line software manager, Software task leader and Training group.

2. Software engineer and systems, programmer; the groups included are: software engineer, related software groups, software engineer Process, Systems engineer, Systems Tests and software configuration management.


When the human resources are normally limited in small organizations, the staff can be involved simultaneously in several projects of partial time, and can take the same project across multiple roles.

In the figure 1 shows an example of the interpretation across a template of activities assisted by specific formats.
Figure 1. Process areas activities, Level 2.

3. Application of interpretation of CMMi for small organization.

The case study analyzed involves a small organization (10 people), where 4 stages (group of activities to implant CMMi) were followed during 12 months. This organization produced commercial software in the center of Mexico (Aguascalientes, Zacatecas, San Luis Potosi and Guanajuato). The initial evaluation of maturity process in this organization are poor, they have some software practices: Requirements and project administration.

The process to application the CMMi in this organization involucres three stages:

- The first stage involved tailoring of existing software processes. The number of activities in phases is reduced and the effort invested is also decreased in terms of project management.
- The second stage was the interpretation and adaptation of Specifics Practices, the steps are: 1) reduction of roles: Software Engineer, Software Quality Assurance, Project manager, 2) development of an activity diagram adjusted to small organizations, and 3) design of artifacts or products for each SP analyzed.
- The third stage involved the design of a training program and reorganization of roles. Since human resources are usually limited in small organizations, different people can be involved simultaneously in several projects in a part time basis, by playing multiple roles in the same project. There should be well-defined rules in order to avoid any conflicts due to multiple roles. There are restrictions on SQA roles and members of test teams, for example, the SQA team should be different from that involved in software development.

Practical Case Goal


Project General Description

The project consists in a software application development in a systematic fashion (include analysis, design and implementation) for learning objects repository also it will be able to create complete courses from several objects of learning. This work was in February’06 at December’06.

System General Goal

To construct a support tool for the search and management of learning objects; in addition to allowing the construction of bigger entities that contain several learning objects...
(courses). The user support consists in a platform that allows him to achieve his goals in an easy, friendly and intuitive way.


Evidences exist on the assertive application of CMMi in projects and small teams, although a complete solution is not still presented to a necessary adjustment for the Model, the factors that consider the reduction of the process are:

- Reduce of the phases considering the necessities of a short project.
- The consideration of priority of activities in small teams are development and construction of the software.
- Reduce the control formats and documentation to facilitate the administration of the project.

The process development organization has been initially applied in a software development company obtaining to the date the following results:

- The application of CMMi to small organizations is possible and contributes to the improvement of their software process.
- The software process improvement was applied.
- The efficiency of the software process was measured for Specific Practices achieving.

As can be seen from this work, it is necessary not only to have a CMMi model tailored to small organizations but also a corresponding evaluation criteria.

Final state of the Software Process Applied

The software process launch has concluded satisfactorily, in spite of had a very big deviation in the execution time of every stage of the process of software, the realized work recovered in his totality and the commitments were fulfilled, there will be delivered a high quality product and high quality process with entire functionality in the agreed time, evaluated successfully by the SQA group.

5. References.


