

CMM compliance in small organizations

Francisco Alvarez
Departamento de Sistemas Electrónicos
Universidad Autónoma de Aguascalientes
(UAA)
fjalvar@correo.uaa.mx

Alfredo Weitzenfeld
Departamento de Ciencias Computacionales
Instituto Tecnológico Autónomo de México
(ITAM)
alfredo@itam.mx

1. Introduction.

CMM [8][6] is currently considered as the de facto standard for process capability certification in organizations dedicated to software development. Independent of their size, these organizations can be certified at one of CMM's five process maturity levels: Initial, Repeatable, Defined, Managed or Optimizing. At each level a number of Key Process Areas (KPA) need to be satisfied, The KPAs are for the Level 2: 1. Requirements Management, 2. Software Project Planning, 3. Software Project Tracking and Oversight, 4. Software Subcontract Management, 5. Software Quality Assurance, 6. Software Configuration Management; for the Level 3: 7. Organization Process Focus, 8. Organization Process Definition, 9. Training Program, 10. Integrated Software Management, 11. Software Product Engineering, 12. Intergroup Coordination, 13. Peer Reviews; for the Level 4: 14. Quantitative Process Management, 15. Software Quality Management, and for the Level 5: 16. Defect Prevention and 17. Technology Change Management.

A large number of the organizations that develop software around the world are small in their size (between 10 and 100 employees). For example, 87% of the companies devoted to software development in Mexico have between 7 and 60 employees [1]. Many of the smaller companies, oppose the CMM model due to the expensive compliance effort, both in time and money. Some of the shortcomings are [5]:

1. Produces excessive documentation.
2. Defines an extensive number of KPAs.
3. Requires extensive resources.
4. Involves high training costs.
5. Defines practices independent of project type.
6. Lacks guidance in satisfying project and development team needs.

Among the shortcomings encountered in applying CMM to software process improvement in smaller organization, it was identified that many of the KPAs do not apply, such as Software Subcontract Management almost in all small projects don't requires external services [7], while other ones require tailoring [4]. These difficulties are closely related to the underlying philosophy of CMM, in other words, CMM is intended for larger organizations with a larger number of employees, where projects tend to be more complex. On the other hand, a limited CMM compliance model tailored for smaller organizations does not exist at this point.

2. Experiences with CMM in small organization.

Some empiric evidences in the application of CMM to smaller development teams and less complex projects show level 2 compliance (Repeatable) by satisfying KPAs at this level only [2].

The case study analyzed involves a small organization (10 people), where 4 stages (group of activities to implant CMM) were followed during 2 years and 10 months.

The first stage involved tailoring of existing software processes. Since the number of activities phases is reduced and the effort invested is also decreased in terms of project management, SQA (Software Quality Assurance) and data compilation (KPAs 2, 3, 5, and 6).



The second stage was the interpretation and adaptation of KPAs in a series of steps: 1) reduction of roles to only three: Software Engineer, Software Quality Assurance, Project manager, 2) development of an activity diagram adjusted to small organizations, and 3) design of products generated for each KPA 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. Mencionar la KPA correspondiente.

The fourth stage concentrated on KPAs 1, 2, 3, 5 and 6 of level 2, The only KPA that required no intervention was KPA 4 (Software subcontract management), since smaller organizations do not usually subcontract software development [7][2].

3. Conclusions.

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

- The software process achieve level 2.
- The application of CMM 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 the KPAs achieving at level 2. (1,2, 3, 5 and 6).

Current work focuses on achieving CMM level 3 by satisfying the remaining KPAs (7,8 and 10) in the organization, Future research will focus on higher CMM levels 4.

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

References.

- [1] Amiti / Bancomext. 2001. Outline of government support to the software industry. Secretaría de Economía, Mexico. (in spanish).
- [2] Batista J., Dias de Figueiredo A. 2000. SPI in a Very Small Team: a Case with CMM. Software Process Improvement and Practice 2000; 5: 243-250.
- [3] Basili, V.R., Rombach, H.D. 1998. The TAME project: towards improvement-oriented software environments. IEEE Trans. On Software Engineering, 14(6), 758-773. (Caps. 24, 25).
- [4] Brodman JG, Jonson DL. 1992. Software process rigors yield stress, efficiency. Signal Magazine, 55, August.
- [5] Brodman JG, Jonson DL. 1995. Tailoring the CMM for small business, small organizations, and small projects. Proceeding of the 7th Annual Software Technology Conference, April.
- [6] Humphrey W. 1993. Introduction to software process improvement. CMU/SEI-92-TR-7, revised, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- [7] Leung H, Yuen, T., 2001. A Process Framework for Small Projects. Software Process Improvement and Practice, 6:67-83.
- [8] Paulk M, Curtis B, Chrissis M., Weber C. 1993. Capability Maturity Model for Software, Version 1.1. CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburg, Pennsylvania, USA.