

IS EMPLOYEE ATTITUDES AND PERCEPTIONS AT VARYING LEVELS OF SOFTWARE PROCESS MATURITY

Janet K. Ply

Pendère, Inc., 1805 S. 9th Street, Waco, TX 76706 U.S.A. {janet.ply@pendere.com}

Jo Ellen Moore and Clay K. Williams

Southern Illinois University Edwardsville, Campus Box 1106,
Edwardsville, IL 62026-1106 U.S.A. {joemoor@siue.edu} {cwillaa@siue.edu}

Jason Bennett Thatcher

Clemson University, 101 Sarrine Hall, Clemson, SC 29634 U.S.A. {jthatch@clemson.edu}

Appendix A

Overview of CMM

In the mid-1980s, the Software Engineering Institute (SEI) and Massachusetts Institute of Technology Research and Engineering (MITRE) Corporation began development of a process maturity framework to help organizations improve their software development and maintenance processes. Over the next several years, the framework evolved from a description of software process maturity (Humphrey and Sweet 1987) to the first version of the Software Capability Maturity Model (Paulk, Curtis, and Chrissis 1991) and version 1.1 (Paulk, Curtis et al. 1993). In 2002, the Capability Maturity Model – Integrated (CMMI) v1.1 was introduced to consolidate the Software Capability Maturity Model (SW-CMM), the Systems Engineering Capability Model, and the Integrated Product Development Capability Maturity Model (SEI CMMI Product Team 2002). Version 1.2 of the CMMI was released in August, 2006. Data for the present research was collected during 2003 and 2004, when few companies had made decisions to move from the SW-CMM to the CMMI. Thus, the participating companies included in this research were those that had achieved levels associated with the SW-CMM, also referred to simply as the CMM.

The CMM is composed of five levels of organizational maturity. The five levels are presented in Table A1. Each maturity level in the CMM is a distinct stage that describes a level of process capability. For example, Level 2 is focused on implementing project management practices and controls to achieve repeatability.

With the exception of Level 1, each maturity level carries a set of key process areas (KPA) (Paulk, Curtis et al. 1993). KPAs for the maturity levels are shown inside the boxes in Figure A1. Each KPA encompasses a set of key practices that are grouped into five common features. The common features represented in each KPA are Activities Performed; Commitment to Perform; Ability to Perform; Measurement and Analysis; and Verifying Implementation. The key practices for each common feature will vary by KPA.

Table A1. Five Levels of the CMM	
Maturity Level	Description
(1) Initial	Software process is characterized as <i>ad hoc</i> , occasionally chaotic. Success is dependent upon heroic efforts.
(2) Repeatable	Basic project management processes are established to track costs, schedule adherence, and functionality. Process discipline is in place to repeat earlier successes on similar projects.
(3) Defined	Software process for both management and engineering activities is documented, standardized, and integrated into a standard software process across the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
(4) Managed	Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
(5) Optimizing	Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

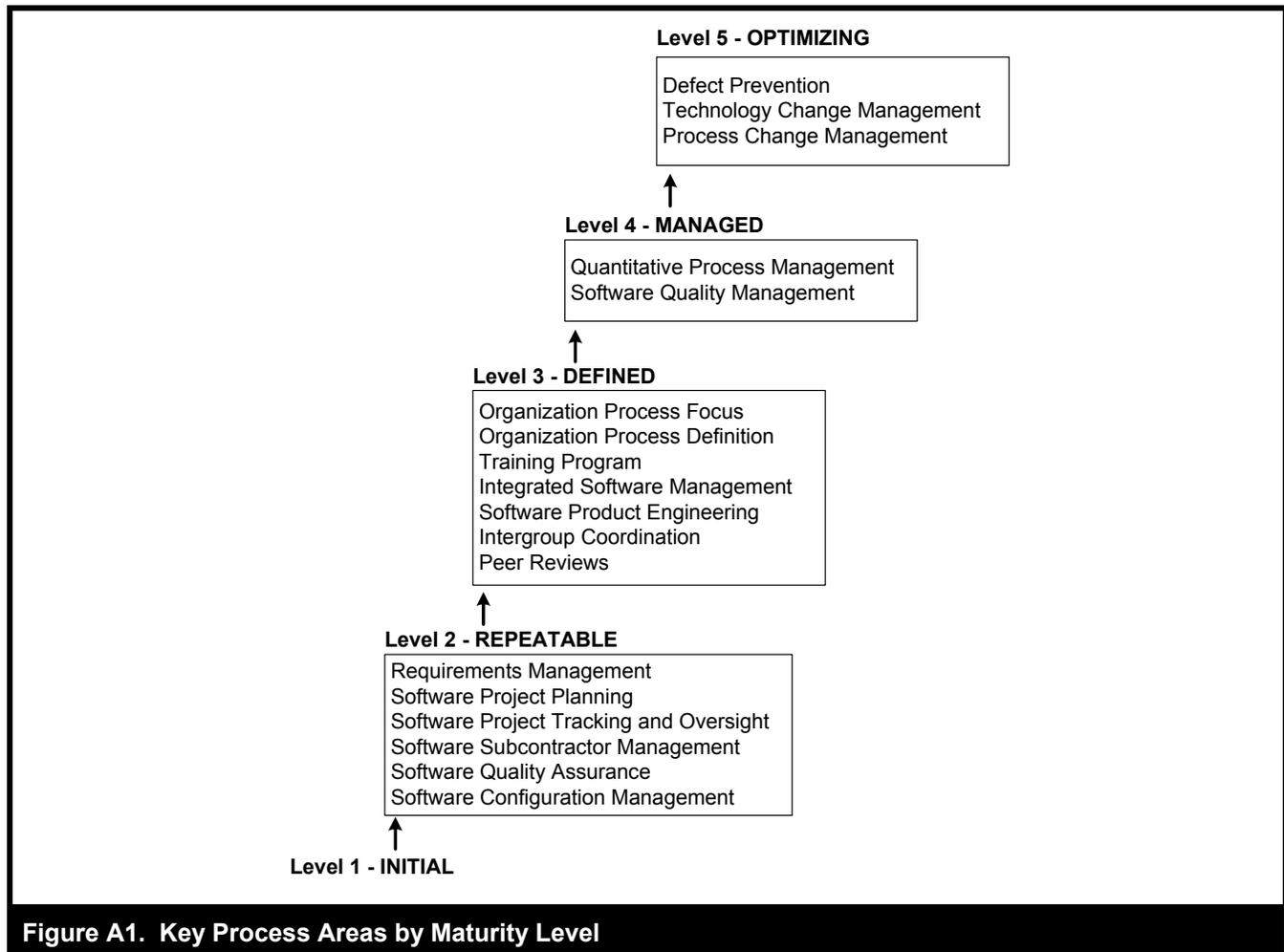


Figure A1. Key Process Areas by Maturity Level

To illustrate how CMM influences the work environment's structure, consider organizational maturity Level 3 (Defined). As shown in Figure A1, one KPA at this level is Organization Process Focus. An example of a key practice within each of the Common Features for Organization Process Focus follows (Paulk, Curtis et al. 1993):

Activity Performed 1: The software process is assessed periodically, and action plans are developed to address the assessment findings.

Commitment to Perform 1: The organization follows a written organizational policy for coordinating software process development and improvement activities across the organization.

Ability to Perform 1: A group that is responsible for the organization's software process activities exists.

Measurement and Analysis 1: Measurements are made and used to determine the status of the organization's process development and improvement activities.

Verifying Implementation 1: The activities for software process development are reviewed with senior management on a periodic basis.

Appendix B

Measures

Cynicism (Proprietary items provided by Consulting Psychologists Press, www.cpp-db.com)

Professional Self-Efficacy (Proprietary items provided by Consulting Psychologists Press, www.cpp-db.com)

Job Satisfaction (Job Diagnostic Survey, Hackman and Oldham 1974, 1975, 1980)

[Scale range: 1 = Extremely dissatisfied; 7 = Extremely Satisfied]

- JS1: The amount of personal growth and development I get in doing my job.
- JS2: The feeling of worthwhile accomplishment I get from doing my job.
- JS3: The amount of independent thought and action I can exercise in my job.
- JS4: The amount of challenge in my job.
- JS5: The amount of pay and fringe benefits I receive.
- JS6: The degree to which I am fairly paid for what I contribute to this organization.
- JS7: The amount of job security I have.
- JS8: How secure things look for me in the future of this organization.
- JS9: The people I talk to and work with on my job.
- JS10: The chance to get to know other people while I am on the job.
- JS11: The chance to help other people while at work.
- JS12: The degree of respect and fair treatment I receive from my boss.
- JS13: The amount of support and guidance I receive from my supervisor.
- JS14: The overall quality of the supervision I receive in my work.

Role Ambiguity (Rizzo et al. 1970)

[Scale range: 7 = very strongly disagree; 1 = very strongly agree]

- RA1: I feel certain about how much authority I have.
- RA2: I have clear, planned goals and objectives for my job.
- RA3: I know that I have divided my time properly.
- RA4: I know what my responsibilities are.
- RA5: I know exactly what is expected of me.
- RA6: Explanation is clear of what has to be done.

Role Conflict (Rizzo et al. 1970)

[Scale range: 1 = very strongly disagree; 7 = very strongly agree]

- RC1: I have to do things that should be done differently.
- RC2: I receive an assignment without the manpower to complete it.
- RC3: I have to buck a rule or policy in order to carry out an assignment.
- RC4: I work with two or more groups who operate quite differently.
- RC5: I receive incompatible requests from two or more people.
- RC6: I do things that are apt to be accepted by one person and not accepted by others.
- RC7: I receive an assignment without adequate resources and materials to execute it.
- RC8: I work on unnecessary things.

Work Overload (Perceived Work Overload, Kirmeyer and Dougherty 1988)

[Scale range: 1 = very strongly disagree; 7 = very strongly agree]

- WO1: I feel that the number of requests, problems or complaints that I deal with is more than expected.
- WO2: I feel that the amount of work I do interferes with how well it is done.

[Scale Range: 7 = Daily, 6 = Almost every day, 5 = Weekly, 4 = 2–3 Times monthly, 3 = Few times per year, and 2 = < 1 Time/year]

- WO3: I feel busy or rushed.
- WO4: I feel pressured.

Appendix C

CMM Assessment Methods

There are two methods for determining a software organization's process maturity rating using the CMM: the CMMSM-Based Appraisal for Internal Process Improvement (CBA IPI) method and the Software Capability Evaluation (SCE). The CBA IPI method was used to assess all of the companies in our sample. The following descriptions are taken from the SEI Technical Report CMU/SEI-96-TR-007 (Dunaway and Masters 1996).

The CBA IPI method is a diagnostic tool that enables an organization to gain insight into its software development capability by identifying strengths and weaknesses of its current processes, to relate these strengths and weaknesses to the CMM, to prioritize software improvement plans, and to focus on software improvements that are most beneficial, given its current level of maturity and the business goals.

The method is an assessment of an organization's software process capability by a trained team of professionals to generate findings and ratings relative to the CMM key process areas within the assessment scope. The findings are generated from data collected from questionnaires, document review, presentations, and in-depth interviews with middle managers, project leaders, and software practitioners.

The CBA IPI method has two primary goals:

- to support, enable, and encourage an organization's commitment to software process improvement
- to provide an accurate picture of the strengths and weaknesses of the organization's current software process, using the CMM as a reference model, and to identify key process areas for improvement

A Lead CMM Assessor assembles and trains a team to use the CBA IPI method to conduct a structured series of activities with key people in the organization to understand their problems, concerns, and ideas for improvement. The method is based on the following key assessment principles:

- Use the Capability Maturity Model for Software V1.1 as a process reference model
- Use a formalized assessment process that complies with the CMM Appraisal Framework
- Involve senior management as the assessment sponsor
- Base the assessment on the sponsor's business goals and needs
- Observe strict confidentiality by guaranteeing that no information will be attributed to an individual or project
- Approach the assessment as a collaboration between the assessment team and the organizational participants

For an assessment to be considered a CBA IPI, the assessment must meet minimum requirements concerning the assessment team, assessment plan, data collection, data validation, rating, and reporting of assessment results. Permissible tailoring options are provided with the requirements.

The other method of assessment, the Software Capability Evaluation (or SCE), is used for software acquisition as a discriminator to select suppliers, for contract monitoring, and for incentives. SCEs can also be used for evaluation of internal processes and the results should be consistent with a CBA IPI if the areas of investigation are the same.

References

- Dunaway, D., and Masters, S. 1996. *CMMSM Based Appraisal for Internal Process Improvement (CBA IPI): Method Description*, Technical Report CMU/SEI-96-TR-007, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Hackman, J. R., and Oldham, G. R. 1974. "The Job Diagnostic Survey: An Instrument for the Diagnosis of Jobs and the Evaluation of Job Redesign Projects," Yale University, Department of Administrative Sciences, Technical Report 4.
- Hackman, J. R., and Oldham, G. R. 1975. "Development of the Job Diagnostic Survey," *Journal of Applied Psychology* (60), pp 159-170.
- Hackman, J. R., and Oldham, G. R. 1980. *Work Redesign*, Reading, MA: Addison-Wesley.
- Humphrey, W. S., and Sweet, W. L. 1987. *A Method for Assessing the Software Engineering Capability of Contractors*, Technical Report CMU/SEI-87-TR-23, ADA 187320, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Kirmeyer, S. L., and Dougherty, T. W. 1988. "Work Load, Tension, and Coping: Moderating Effects of Supervisor Support," *Personnel Psychology* (41:1), pp. 125-139.
- Paulk, M. C., Curtis, B., and Chrissis, M. B. 1991. *Capability Maturity Model for Software*, Technical Report CMU/SEI-91-TR-24, ADA 240603, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. 1993. *Capability Maturity Model for Software, Version 1.1*, Technical Report CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Rizzo, J. R., House, R. J., and Lirtzman, S. 1970. "Role Conflict and Role Ambiguity in Complex Organizations," *Administrative Science Quarterly* (15), pp. 150-163.
- SEI CMMI Product Team. 2002. *Capability Maturity Model[®] Integration (CMMISM), Version 1.1*, Technical Report CMU/SEI-2002-TR-028, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.