Agile Methods and CMMI-SW: Dancing Elephant on the Internet Zone

Ghazi I. Alkhatib
P. O. Box 36007
Qatar College of Technology
Doha - Qatar
alkhatib@qu.edu.qa.

Abstract

The article explores the link between agile methods and CMMI-SW structure to find out how agile methods fit into CMMI. Agile (light) methods have been successfully used for developing internet-based applications, while CMMI-SW has been used for large outsourced projects. We are hoping that this paper will assist CMMI-SW certified IT Department to utilize CMMI-type (heavy/called elephant sometimes) processes in developing internet-base applications. The comparison of agile methods principles and the structure of CMMI-SW will identify areas where the first one can assist the second one in applying CMMI process in a more dynamic environment with changing requirements. Major areas of links are identified, such as validation and verification, customer involvement, project management of the software development life cycle, and the risk factor. Then two major issues are highlighted: team size/structure and human factor. Throughout these discussions, the paper provides three suggestions to strengthen the link between the AM and CMMI: HBDI creative thinking process, usability laboratories, and a multi-level/multi-dimensional circular team structure. The elephant may, at last, dance on the Internet zone.

Key words: Agile method, CMMI-SW, linking Agile and CMMI, human factor, team size, HBDI, usability laboratories, validation and verification.

1 Introduction to Agile methods and CMMI-SW

Agile methods (AM) efforts were culminated by the agile manifesto, as shown in Figure 1 [1].

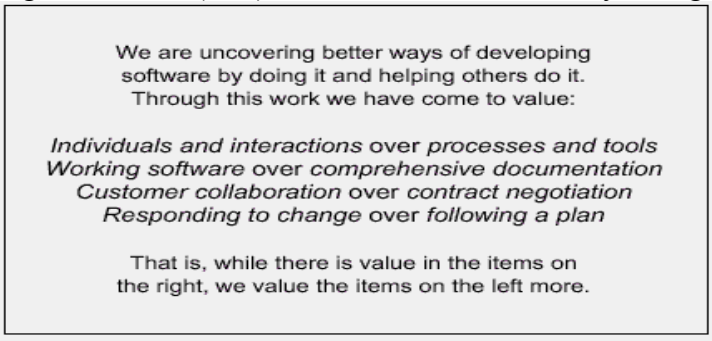


Figure 1. The Agile Manifesto

AM depend on small teams, frequent releases and iterations, less emphasis on documentation and more on communication with users, and less dependent on models and structured development life cycle and more on dynamic responses to changes in software requirements. Capability Maturity Model Integration Software (abbreviated as CMMI-SW and used in this paper alternatively with CMMI) [2]evolved from the multiple models of CMM as a single reference model to make it easier for organization to follow model requirements. Table 1 below shows a summary of the CMMI-SW structure. CMMI has 5 levels of general goals (GG) with a total of 18 generic practices (GP) intersecting with 4 major process areas categories (PAC) having a total of 25 process areas (PA). For each PA, specific goals (SG) are stated with associated specific practices (SP) and are linked to relevant GGs./ CLs. For each specific practices (SP) one or more sub-practice (SuP) are provided, as well. Each PAC is also stated against each GG and its component of GPs with elaboration provided for most of the GPs. This makes GG1 and PAC engineering the most critical areas in CMMI. On the other axis, each PAC has two types of PA: basic and advance, except engineering that has no such distinction. The numbers for SGs and SPs show that: engineering PAC is the only one with SGs and SPs at GG2 and GG3 levels. The Table clearly shows that there in no discussions on PAC in relationship to GG 4 and 5. Also, we note that Engineering PAC has about one third of each the SGs and SPs. The CMMI manual starts with PAs, and then states SGs and related SPs, followed by the list of the GGs and GPs. The related discussions for all the PACs follow the same order. This makes Engineering PAC and GG1 are the most critical areas in CMMI discussions. No wonder, then, that CMMI is referred to as an elephant.

Table 1. CMMI General Goal (GG) /capability levels and Process Area Categories (PAC) and the distribution of the number of specific goals and specific practices among them

Generic Goal (# of Generic Practices)							
GG5: Optimized (2)							
GG4: Quantitatively Managed (3)							
GG3: Defined (2)					7		
GG2: Managed (10)					11		
GG1: Preformed (1)	19	12	31	23	32	19	11
GG0: Incomplete (0)							
Total specific practice (165)	19	12	31	23	50	19	11
Total specific goals (47)	5	3	7	7	15	7	3
		2	3	5	6	3	3
Basic process area Advance process areas Both basic and advance							
Process Areas Category (PAC)	Process Management		Project Management		Engineering	Support	

This research started by assuming that agile methods are best suited to develop internet and e-business systems, while CMMI-SW, on the other hand, is best suited to develop large contractual transaction-based systems [3]. At the same time, this research does not follow the "leave it all" or "take it all" approach. Advocates of both sides of the coin need not worry, because after detailed study of CMMI statement, I found out that many of the CMMI requirements could not be performed without adhering to some form of agility principles. What would happen with the agile movement? Are they going to adapt to some of the CMMI requirements, mainly generating

Top

some documentation artifacts throughout its processes? We will leave that for further considerations by agile supporters. In the meantime, we believe CMMI-SW is the clear winner by trying to introduce some form of agility in performing its processes. This paper will highlight areas where CMMI may benefit from agile methods.

2 Review of previous studies

Two studies were found in the literature. One study compared XP to CMM [4] and another study surveyed participants in a workshop on agile methods on how AM principles are related to CMMI-SW [5]. After converting Paulk study to CMMI structure, we listed the results of both studies in Appendix a, after expanding the GGs and PACs. Analyzing previous research, I found that the complex cross-referenced model structure of CMMI makes it difficult to devise a research strategy for comparing CMMI and AM. Therefore, this research conducts detailed analysis of CMMI-SW document searching for areas of links to AM. In this process, we looked at literature related to AM and CMMI as well, and provided some suggestions as to how the link between AM and CMMI could be strengthen. Our final objective is to capitalize on the benefits AM can bring to CMMI, so the elephant may dance 'sometimes'.

3 Areas where agile methods can be used in CMMI-SW

The following sections present some of the issues and areas where a link could be established between CMMI and AM.

3.1 To plan or not to plan and the risk factor

Boehm distinguished between plan-driven and agile-driven software development. One factor to consider when selecting which one to use is risk. Boehm's [6] article proposes that high risk project use plan-driven approach, such as CMMI-based methods, while use agile-based methods for low risk projects. On the other hand, Orr [7] suggests that agile-based methods are used for high risk projects. It seems that if a project is classified as a high risk one, then we need to construct strong development teams. Consequently, these teams will work better with agile-like methods than work with heavy-handed CMMI-based environment.

As a starting point, CMMI list the following sources of risk in projects: Uncertain requirements, Unprecedented efforts—estimates unavailable, Infeasible design, Unavailable technology, Unrealistic schedule estimates or allocation, Inadequate staffing and skills, Cost or funding issues, Uncertain or inadequate subcontractor capability, and Uncertain or inadequate vendor capability [2]. We may expand the list by three new risk factors:

- Aptitude of all levels of users/management towards involvement in software development activities. They should believe that "it takes two for the ride: IT staff and end-user."
- Computer/IS literacy level of users.
- Level of software engineers' knowledge of business processes and management practices. Surveys in [8] reveal that software engineers lack knowledge in business and management skills.

Top

Risk mitigation techniques may include "…. prototyping, simulation, alternative designs, or evolutionary development." [2]. The first and last techniques are particularly relevant to AM through its support of iterations and releases development. Agile methods with its ability to respond to change and emphasis on customer interaction fit better to deal with such projects. Once risk is alleviated, project teams could use CMMI for detailed planning and documentation.

3.2 The human factor

The human factor is a critical element in software development, especially when employing AMs [6, 9]. Also, CMMI-SW emphasized empowered workforce to achieve agility at CL5. DeMarco-Boehm debate [10] raised several words related to human factor, such as "premium people," "skill building," and "superman." The assumption is a simple one: you need better people to develop systems with agile methods than any other methods. We raise several questions. Can the best horse and worse jockey win a major derby? Can software quality be accomplished simply by instilling organizational rules, procedures and forms, without a committed, dedicated, trained, educated IS professionals and users? Can we give a blind trust in CMMI certification [11]? These questions become even more critical when dealing with contracted or outsourced projects. How can an organization verify and guarantee the quality of the contractor's development employees? For example, one contractor for an ERP implementation brought highly qualified consultant during initial contacts. Once implementation started, however, inferior quality people were used. If we presume that AM and CMMI require different skills and even different capabilities, how an organization can determine which IS employee would fit better with CMMI or AM? This is the reason why CMMI supports agility at the highest capability level. However, with the small percentage of organization achieving capability levels 4 and 5, many of these organization may never employ agile methods Therefore, CMMI may be best suited for large outsourced and contracted projects with low risk. For in house development, however, CMMI could be intermixed with AM depending on the profile of individuals and teams. We suggest using the Herrmann Brain Dominance Instrument (HBDI) [12] as a guideline. For those who are not familiar with such technique, Appendix b contains an exposition of the HBDI technique. AM needs individuals and teams, who are strong in quadrants C and D. On the other hand, individuals and teams who are strong in quadrants A and B perform better under CMMI type of environment. With some form of dynamic team membership program, we could control team profiles to match project's human resource requirements: type CMMI or AM. We hope that CMMI-based organization will allow some form of agility to be performed on carefully selected projects and teams. Further use of HBDI in organization would compare organization's profile to individual profiles as well. An innovative organization will be strong in C and D quadrants, where AM advocates believe that AM could be used [13].

3.3 Lessons learned

CMMI stressed in many places sharing lessons learned. We classify lessons learned as unstructured information requiring knowledge management technology to capture, classify, store, distribute and use of such knowledge. To share such knowledge manually, AM is best suited through its strong support of face-to-face interaction between team members, which must include users. If some form of groupware technology is used, especially for large projects, for example

Top

[14], to store lessons learned from AM and CMMI practices, either CMMI or AM may use such technology, though the latter one to a lesser degree, since AM doesn't recommend the use of automated tools.

3.4 Validation and verification

CMMI lists validation and verification as two separated process areas under "engineering" PAC. It advocates that both be conducted concurrently if possible. This is very difficult to accomplish if milestones are far apart from each other. AM is perfectly fit to accomplish this objective by adhering to small delivery iterations, supporting verification, and requiring users to be an active participant in team meetings, supporting validation. We believe this is an area where AM can significantly improve CMMI processes. To actuate concurrent validation and verification, we recommend the deployment of Usability Laboratory. Usability testing was used mainly for system software before final release. Currently, however, studies on usability for business applications started to emerge [15]. The usability laboratory, or else called technology lab [16], may not be as elegant and complex as those used in IBM and Microsoft, for example. However, they should be equipped adequately for a given budget constraint and application types developed the by organization. Such laboratory should be used for briefing top management as well as actual testing of releases by lower level users. An example of user interface (UI) tools (launchpads) that could be employed to enhance validation and verification of system's UI is the one used in IBM usability laboratory [17].

3.5 Team size and project size

AM advocates the use of small teams regardless of project size. CMMI identifies three types of risks: Cost, schedule, and performance risks (that) should be examined during all phases of the product life cycle to the extent they impact project objectives. CMMI is concerned with process granularity, as well. To apply AM in CMMI organizations, process granularity should be very small while phases should be very short. If we assume that teams will be assigned to processes within phases, then we have to be concerned with team integration, a process area under "project management" PAC. Also, the following SuP for SG1-4.2 "The traceability should cover both the horizontal and vertical relationships, such as across interfaces." [PA146.IG101.SP104.N101]. In addition, GP2.10 lists "review status with higher management." Consequently, and after combining all these requirements, we need a team structure that allows inter-team communication in all directions. These teams may operate sequentially or concurrently. This research provides a suggested team structure based on circular team organization [18] that best fit AM and CMMI requirements. Circular team allows inter-team communication to span multi-level management as well as cross-functional. For e-commerce and multi-national corporations, we need to involve all constituents to form inter-organizational teams and achieve effective communication. It proposes a multi-tier cross-functional inter-organization and dynamic assignment team structure, as shown in figure 2.

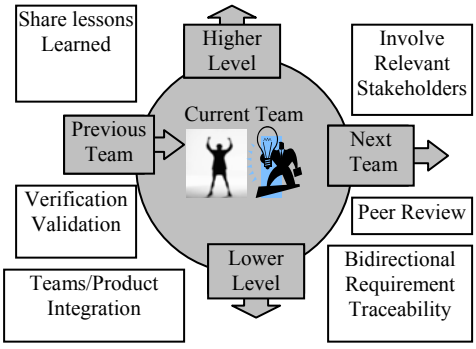


Figure 2. Multi-level dynamic team structure


IS Professionals and Users, respectively

Top

Just like AM recommends, teams will have both IS and users members. Team members at any given stage may consist of at least four types of members: from the previous team, subsequent team, current IS, and current users. However, user and previous/subsequent team members will not serve 100% of the time, and it will follow dynamic coordination policy as suggested in [19]. Previous team member will serve enough time for current members to understand previous team requirements and how it relates to current team objectives. If needed, subsequent team representatives will be called in at the completion of current team requirements till the hand-over to the next team. Teams will be constructed based on the functional structure of the organization, i.e. production, accounting, marketing, human resources, and aligned with the hierarchical structure of the organization. Number of teams will be decreasing as we move to a higher level. The lowest user reference for teams will be data entry clerks. To insure effective communication, team size will be kept as small as possible.

At any point, teams will be communicating in four directions: from the previous team, to next team, to a higher level team, and to a lower level team. For real large projects, concurrent teams could be operating at the same time and all levels but mainly at the middle level. If this becomes necessary, two more communication directions are added: left and right communication – bringing communication directions conceptually to a total of six-, and team members will be increased by one. When two or more concurrent teams complete the requirements for one stage, the two teams will merge into one team, removing redundant experiences and knowledge. Meeting duration and frequency will be more at lower levels than at higher levels. Nature of interface between IS and users will be different as teams move to higher levels, stressing briefings and validation of requirements. This team structure will assist in achieving several goals in CMMI, as shown in Figure 2. It will also assist in deploying AM in large projects.

3.6 Requirement Management

CMMI lists the following specific goal and specific practices under "requirements management" PAC. The last digit indicates the capability level it applies to.

SG 1 Manage Requirements [PA146.IG101]

- SP 1.1-1 Obtain an Understanding of Requirements
- SP 1.2-2 Obtain Commitment to Requirements
- SP 1.3-1 Manage Requirements Changes
- SP 1.4-2 Maintain Bidirectional Traceability of Requirements
- SP 1.5-1 Identify Inconsistencies between Project Work and Requirements

Under SG 1, the following statement is given about the scope of requirement "The requirements may be a subset of the overall product requirements, or they may constitute the entire product requirements. [PA146.IG101.AMP101] This may indicate that AM may be used in projects with narrow scope, while CMMI is used with large scope projects.

Furthermore, a SuP is provided for SP1.1-1 as "Reach an understanding of the requirements with the requirements provider so the project participants can commit to them. [PA146.IG101.SP101.SubP104] Two SuP are provided to 1.3-1 as "Evaluate the impact of requirement changes from the standpoint of relevant stakeholders [PA146.IG101.SP103.SubP103], and "Make the requirements and change data available to the project. [PA146.IG101.SP103.SubP104]

Top

With these statements in mind, AM provides at the front line the mechanism to achieve these SPs and SuPs effectively through active participation of users in project teams and its strategy to deal with changes in requirements. In the background, however, we need to keep CMMI documentation for these SPs created, updated, and maintained. .

3.7 Requirements Development

Under SP1.1-2 Elicit Needs, the following list is provided. We listed only the techniques where AM can be used through its direct interaction with end-uses and its support of iterations and releases.

Examples of techniques to elicit needs include the following: [PA157.IG101.SP102.N103]
• Interim project reviews • Questionnaires, interviews, and operational scenarios obtained from end users • Operational walkthroughs and end-user task analysis • Brainstorming • Use cases • Business case analysis

4. Conclusions

This article discussed how AM can assist in CMMI processes. Major areas where AM can assist CMMI are in validation and verification, stakeholder involvement, and risk management. PACs with a weaker link between CMMI and AM are requirement management and requirement deployment. This help was possible because AM principles of user communication, small teams, and small development iterations. Continuous lines in Appendix demonstrate current research areas. The article suggested the use of HBDI creative thinking to assist in assigning individuals to teams to achieve team effectiveness and communication and their respective mapping to CMMI and AM approaches to software development. Also, we recommend the use of usability laboratory to facilitate validation and verification. However, the plan to deploy AM practices in CMMI organization should be done carefully, so the elephant can dance without falling down. We also hope that our research, especially the use of HBDI, may assist organizations to get closer to GP 5.1 which states "Optimizing processes that are agile and innovative depends on the participation of an empowered workforce aligned with the business values and objectives of the organization. The organization's ability to rapidly respond to changes and opportunities is enhanced by finding ways to accelerate and share learning." Another area of research is linked organization learning to CMMI, such as the link between ISO and OL [20]. In the meantime, further experimental research on the use of AM and post-implementation evaluation of AM adopters is needed to validate AM's claims of success. At the end, we need research on how GP5.1 on agile and innovation processes is linked to the different PACs, since all cross discussions of the PACs with GG/CL 4 and 5 are listed without any elaborations. Finally, we hope that this research will encourage CMMI-certified IT departments and software houses to consider incorporating AM in CMMI processes in order to develop emerging dynamic internet-based information systems [21].

Top

5. References

[1] www.agilemanifesto.com

[2] CMMI-SW (2002), Capability Maturity Model® Integration (CMMI[SM]), Version 1.1 CMMI[SM] for Software Engineering, 2002. Continuous Representation CMU/SEI-2002-TR-028 ESC-TR-2002-028 CMMI August.

[3] Alkhatib, Ghazi, "Agile Methods – CMMI-SW: Linking the Two Extremes," Proceedings of the Second Conference on Administrative Sciences (KFUPM, April, 2004), pp. 461-472.

[4] Paulk, M. (2001). Extreme Programming form CMM Perspective. *Paper for XP Universe*, and later published in *IEEE Software*, November-December, 2001, pp. 19-26.

[5] Turner, R., and Jain, A. (2002). Agile meets CMMI: Culture clash or common cause? Lecture Notes on Computer Science, Electronic Edition, pp. 153-165. available through www.informatik.com.

[6] Boehm, B. (2002). Get Ready for Agile Methods, with Care. *Computer*, 35:1, pp. 64-69.

[7] Orr, K. (2003). CMM Versus Agile Development: Religious Wars and Software Development. 3:7, http:// **www.cutter.com/project/index.html**.

[8] Lethbridge, T. (2000). What Knowledge Is Important to a Software Professional? *Computer*, 33:5, pp. 44-50.

[9] Cockburn, A. and Highsmith, J. (2001b). Agile Software Development: The People Factor. *Computer*, 34:11, pp. 131-133.

[10] DeMarco, T., and Boehm, B. (2002). The Agile Methods Fray. *Computer*, 35:6, pp 90-92.

[11] O'Connell, E. and Saiedian, H. (2000). Can You Trust Software Capability Evaluations. *Computer*, 33:2, pp. 28-35.

[12] http://www.hbdi.com

[13] Cockburn, A. and Highsmith, J. (2001a). Agile Software Development: The Business of Innovation," *Computer*, 34:9, pp. 120-122.

[14] Reifer, D. A Little Bit of Knowledge is a Dangerous Thing. *IEEE Software*, pp. 14-15

[15] Gilbert, A. (2003). Business Apps Get Bad Marks in Usability. CNET News.com, January 14, 5:00.

Top

[16] McCarthy, J. (2003) “The IT Lab Evolution. InfoWorld-CTO Network, February 14. www.infoworld.com.

[17] Wickham, D. P. (2001) "Empowering users with launchpads," http://www.ibm.co

[18] Romme, A. G., and Witteloostuijn, A., (1999) “Circular organization and triple loop learning.” Journal of Organizational Change Management. Vol. 12, No. 5. 1999.

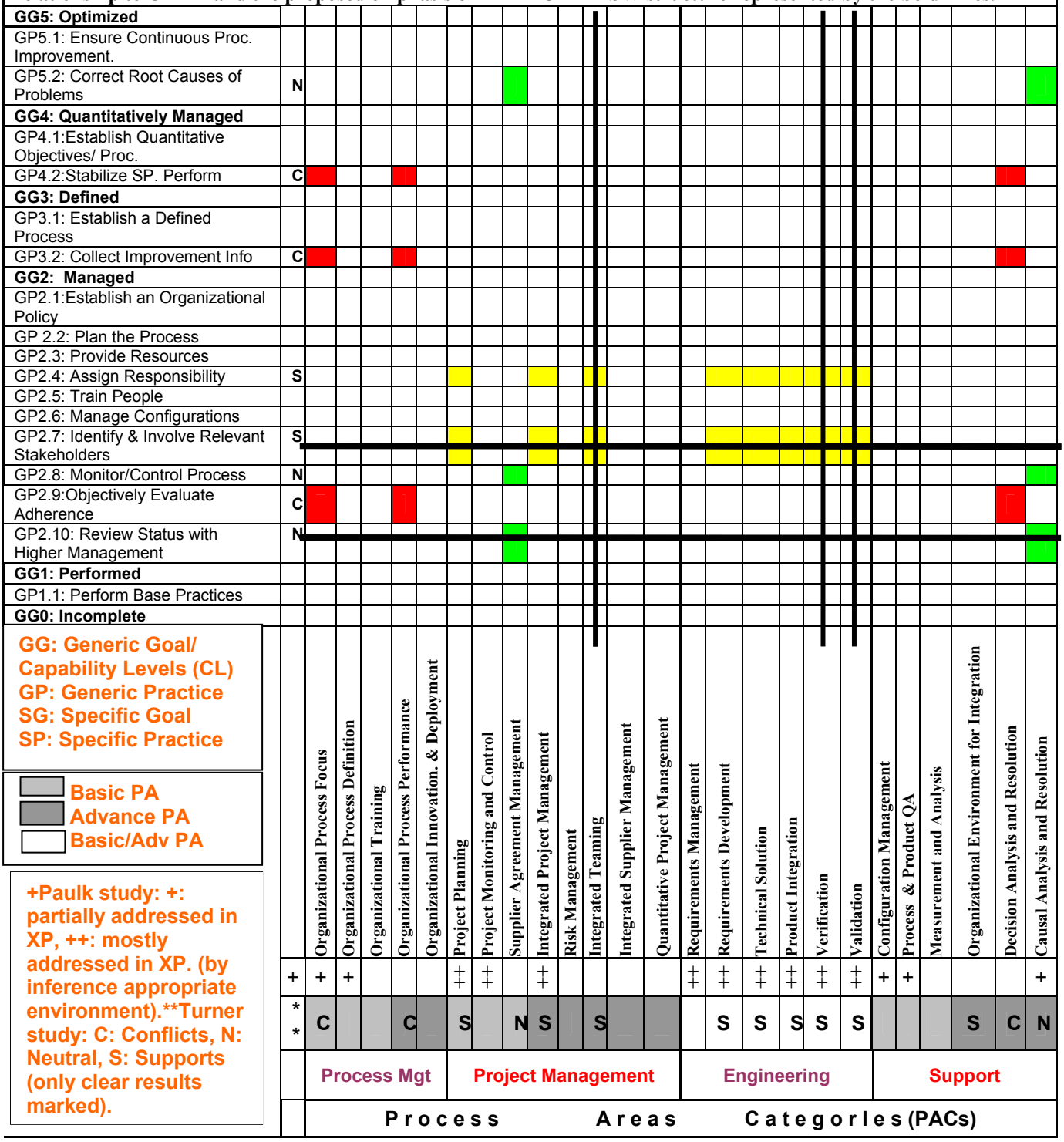
[19] Mookerjee,V, and Chiang, I, (2002). A Dynamic Coordination Policy for Software System Construction, *IEEE Transaction on Software Engineering*, 28:7, pp. 684-694.

[20] Zain-ul-Abdeen T., G. Alkhatib, (2003) ”A Framework for Building a Learning Organization Through ISO 9000 Requirements,” Proceedings of Americas Conference on Information Systems (AMCIS), page 2607-2617, Tampa, FL, July 4-6 2003.

[21] Ambler, S. (2002). Lessons Learned in Agility From Internet-Based Development. IEEE Software, 19:2, pp. 66-73.

Top

Appendix a. CMMI capability levels and process areas with results of previous research on Agile methods relationship to CMMI and the proposed emphasis of AM in CMMI-SW structure represented by the bold lines.



Top

Appendix b. HBDI for Creative thinking and team effectiveness and innovation

The following is a brief discussion on Herrmann Brain Dominance Instrument (HBDI) creative thinking concepts.

The Four Brain Quadrants

The brain is divided on four quadrants: Upper left, called A quadrant; lower left, called B; lower right; called C quadrant; and upper right, called D quadrant. Each of the quadrants has its own characteristics. The following is a summary description of these quadrants:

A quadrant: Logical, factual, rational, critical, analytical, quantitative, authoritarian, and mathematical. The blue color is used to represent this quadrant.

B quadrant: Technical reader, data collector, conservative, controlled, sequential, articulate, dominant, and detailed. The green color is used to represent this quadrant.

C quadrant: Musical, spiritual, symbolic, talkative, emotional, intuitive, (regarding people), and reader (personal). The red color is used to represent this quadrant.

D quadrant: Intuitive (regarding solutions), simultaneous, imaginative, synthesizer, holistic, artistic, and spatial. The yellow color is used to represent this quadrant.

Based on a survey by HBDI, about 60% of the surveyed people are with two quadrants, 30% with three quadrants, 7% with one quadrant, 3% with four quadrants.

Each quadrant is divided into four levels. These are listed from the inside to the outside: low, intermediate, strong, and very strong. And each quadrant is divided into two sub-pies.

Top