On Improving CMMI in an Immature World of Software Development

JING-CHIOU LIOU
Department of Computer Science
Kean University
Union, NJ 07083, U.S.A.

During the last decade, software engineering researchers have tried to identify the success and obstacle factors to the software development. Conclusions from many of the researches related to the issue show that software process is one of the most important subjects contributing into the success or failure to the software development. Among the Capability Maturity Model Integration (CMMI), the ISO9000 series, Total Quality Management, and other best process practices, the CMMI is the one specially focusing on the software development. However, as evolving from the Capability Maturity Model for Software (SW-CMM), the CMMI has inherited some process issues that can severely degrade the effectiveness of process maturity of an organization. In this paper, we demonstrate some of the issues existing in current version of the CMMI. The upward only maturity level certification method is a major issue. Another major issue for the CMMI is its lack of process practice(s) for organizations and their suppliers that have higher maturity level than they do. Some possible remedies for those issues are also proposed to improve the CMMI.

Keywords: CMMI, software process improvement, SDLC, rapid development, quality

1. INTRODUCTION

Software development organizations are facing more challenges and difficulties today as the complexities of software applications significantly increasing, the technologies of programming languages rapidly changing, and the design patterns and tools constantly modifying. In addition to required training that helps the staffs to improve their skills, software engineering plays a major role on modern software development. Software engineering helps a software development organization to have better management on its projects, provides a means for quality assurance, reduces the cost for production maintenance, as well as establish statistical process control for process improvement.

1.1 Background on Software Process Management and Improvement

There are some existing standards or guidelines to help software development organizations to establish the processes that suit their needs. The Capability Maturity Model Integration (CMMI) [1], the ISO9000 series [2], and Total Quality Management [3] are well known parts of the process management and improvement. Among them, the CMMI is the one specially focusing on the software development.

The Capability Maturity Model Integration (CMMI) evolved from the Capability Maturity Model (CMM) [4]. The concept of the CMM started at late 1980s and appeared to the public when Watts Humphrey [5] published his book “Managing the Software
Process” that was based on the earlier work of Phil Crosby [6].

However, before the book was published, funded by US Air Force, the Software Engineering Institute (SEI) at Carnegie Mellon University has actively worked on the model in 1986. The SEI published the SW-CMM V0.2 in 1990 for software development, and released the V1.0 and its assessment method in late 1991.

Since then, the CMM model has gained its popularity through contractors of US Department of Defense (US DoD) and other software development organizations around the world. The trend of adopting SW-CMM into organizational software processes is driven by primary two reasons: (1) The lack of good process practices/guidelines for software development organization, and (2) the mandatory requirement of CMM level-2 certification and above to acquire US DoD contracts. [7]

Before the SW-CMM, there are some best practices that were widely used in the software industry. Total Quality Management (TQM) [3] and ISO9000 [2] are two examples. TQM appeared in the earlier 1980s and soon became popular as it specified clear objectives of quality and process to achieve them. The ISO9000 released in 1987 was designed for organizations whose activities include the creation of new products. The revision of ISO9000 in 2000 (called ISO9000-2000 to distinguish it from the ISO9000 released in 1994 and earlier versions) adopts similar concepts of TQM.

Nevertheless, TQM and the ISO9000 were unable to meet the needs of software industry. Even the process laid out by TQM has clearly defined 4 phases of the development cycle, it did not address the issues on software process activities in details. Similarly, the quality management model of the ISO9000 was geared for manufacturing and the emphasis tended to be on the conformance with procedures for quality assurance rather than the overall process management and improvement.

Although ISO9000-3 released in 1991 focuses on the development, supply, and maintenance of software [8], it still could not provide the answer that software industry is looking for. Both models failed to accommodate the uncertainty in the software development process that forced the software industry to seek for alternatives. With the issues described above for TQM and the ISO9000, software organizations turned their eyes onto CMMI for a better path to improve on their software development processes.

The motivation of this research is to review the lessons we have learned from software development organizations that have granted certain CMMI maturity level, and to seek for improvements. These software organizations are those that the author either worked at as a software process staff or worked with as a consultant for CMMI software process improvement. The size of these organizations range from around a thousand staff to several thousands staff and are either private or government sector.

The actual results, including human factors and issues, presented in the paper were collected from software metrics reports of those organizations. Some of the human factors that negatively impact on software process had been reported in literature (see section 2 for details.) Some issues not shown on this paper were removed after the revision of CMMI V1.2. Hence, the presented issues and remedies focus on current version of CMMI guidelines.

1.2 The Capability Maturity Model Integration

The CMMI model was formally introduced in 2002 when the V1.1 version was pub-
lished. The revised CMMI V1.2 released in 2006 co-exists with the SW-CMM until the phased out of the CMM model in August 2007. It evolved from the SW-CMM, but it adds the level 0 to the five-level of capability maturity in SW-CMM.

The six capability maturity levels, designated by the numbers 0 through 5, are as follows [1]:

0 → Incomplete: No process in place. Project success depends on personal skills and experiences.
1 → Initial: Processes unpredictable, poorly controlled and reactive.
2 → Managed: Processes characterized for projects and is often reactive. This level covers 7 process areas for repeatable success of project implementation: Requirement management, project planning, project monitoring and control, supplier agreement management, measurement and analysis, process and product quality assurance, and configuration management.
3 → Defined: Processes characterized for the organization and is proactive (Projects tailor their process from the organization’s standard) This level covers 11 process areas for organization processes standardization.
4 → Quantitatively Managed: Processes measured and controlled. This level covers 2 process areas: Organizational process performance and quantitative project management.
5 → Optimizing: Focus on process improvement. This level covers 2 process areas: Organizational innovation and deployment and causal analysis and resolution.

There are 22 process areas in CMMI V1.2. CMMI supports two improvement paths: continuous and staged. The continuous path enables organizations to incrementally improve processes corresponding to an individual process area (or process areas) selected by the organization. The staged path enables organizations to improve a set of related processes by incrementally addressing successive sets of process areas.

These two improvement paths are associated with the two types of levels that correspond to the two representations. For the continuous representation, we use the term “capability level.” For the staged representation, we use the term “maturity level.”

The SEI also defined the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) for CMMI level-2 to level-5 certification. The CMMI level 0 is actually a pseudo level referring to any organizations that have no sound standard development processes. The CMMI level 1 is another pseudo level. It is difficult to distinguish level 0 and 1 as no formal method defined to assess them.

The SCAMPI defines three appraisal approaches: A, B, and C. SCAMPI A is utilized for capability maturity level certification that requires an appraisal team and a lead appraiser. SCAMPI C is for organization internal assessment and does not require a lead appraiser. SCAMPI B is most likely being used as a mock appraisal prior to SCAMPI A.

The CMMI model provides a very good guideline for organizations that are seeking process management and improvement. The activity-based pattern at the CMMI level-2 and level-3 presents a clear path and easy implementation for organizations to establish the standard processes starting from scratch and to keep them intact. At the CMMI level-4 and level-5, the centralized process group within an organization concentrates on process improvement effort and assures the process reflecting the organization’s business change.
This paper is organized as follows. In section 2, the effects of human factors that negatively impact on software processes of CMMI assessed organizations are identified. These factors provide us a high level understanding where the CMMI guidelines can be improved. Process issues within CMMI guidelines are discussed in details in section 3. Section 4 presents a set of remedies that could improve the CMMI base on those process issues listed in section 3. Finally, a conclusion of this study is shown in section 5.

2. HUMAN FACTORS IMPACTING ON PROCESS MATURITY

In general, a software development organization that wants to be competitive, improve quality, and increase productivity, the following actions are required [9]:

• Focus on the processes that develop the software to improve quality and productivity.
• Ensure that the processes are properly supported.
• Manage poorly behaving processes by fixing the processes, not blaming the people.
• Process improvement comes from reducing variation in all processes.
• Management action uses data from the process to guide decisions.

Moreover, process management and improvement are necessary to ensure the process serving its purpose. When a process is clearly defined, well measured, and under control, the process can reach a certain level of maturity. In that maturity level, a software development organization can achieve its goals by employing the software process as the policy of development.

However, it was reported in [10] that quite a few human factors could negatively impact on the process maturity of a software organization that is certified for certain CMMI maturity level, either intentionally or unintentionally. Those factors vary from change in leadership to technology upgrade. All of the following cases, grouped in different categories based on their root causes, were actually experienced in a 5-year period in two telecommunication software applications development organizations, one called SDS, with about 1,800 staffs, another called NMS, with about 800 staffs. These cases were also observed in an US Army Software Engineering Center that was seeking for CMMI level-2 appraisal in 2007. These human factors can be categorized into three case studies.

2.1 Change of Maturity Level

The request for an accomplishment of achieving specific process maturity level (e.g., change from CMMI level 1 to 2) usually comes from the upper management of an organization. However, following factors that had influence on the process maturity were identified during and after the assessment:

• Negligence from low-level management: Even the CMMI effort was requested from the director of the organization, with no funding to the CMMI effort, some low-level managers responded, “I’d rather have it (process) wrong than have it (schedule) late” For other managers, the CMMI effort is as important as the product delivery, but they just did not have the resources to do all of the works. Hence, they relied on the CMMI consultant to recommend the minimum requirements to achieve the maturity level in
minimizing the learning curve and additional efforts. In both cases, the special force (the CMMI core team) was expected to satisfy all of the key practices of KPAs with recommended set of artifacts. Any conflicts between daily works and the process works must be absorbed by, neither the consultants nor the core team, but the project team.

- **Resistance from staffs**: With usually a short timeframe to accomplish the goal, some recommendations (fixed formulas and checklists) were produced. When the organization followed the checklist strategy, piles of documents and mountains of paper were generated to “prove” that the software processes are at some prescribed level. The staffs would resist doing more “process work” if the core team suggested to them. Even they followed the recommendation they would prefer not to execute the process completely.

- **Miscommunication between the directorate process team and project team**: The Software Engineering Process Group (SEPG), consisting of assigned full time process staffs and voluntary project representatives, is responsible for the organization’s software process activities. While the project representative is not a very popular assignment, not many projects in the directorate sent their representatives to the SEPG. The result is that many negative feedbacks would be sent back to SEPG once the process was effective and projects were asked to follow. Some of the typical comments were:

  - Some (or our) project has already a very good process for this.
  - The process will not work for our project.
  - Why my project was not invited in developing the process.

### 2.2 Change of Development Structure

Re-organization is a very common phenomenon in the software industry. A case is studied here to examine the influence of human factor on the software process during the organization re-structure.

A software engineering group, called W team, was assigned to develop software applications for three customers. The group manager was very process-oriented and the group has established some well defined software processes. These defined processes were completely executed during the development cycle:

- **Detailed project planning and management**.
- **Complete configuration management**.
- **Focusing on quality**: Statistical methodologies and heuristics were used for performance analysis.
- **Emphasizing on process management and improvement**.

When the re-structure happened, another software engineering group, called B team, was assigned to development a similar telecommunication application for a new and bigger market customer. With the two teams developing similar function of application in parallel, the levels of process maturity of these two teams are totally different.

While the manager of the B team does not focus on the process management, in order to meet the schedule, the B team was prototyping the development. The best description of the development concept is: “Let us just get the job done by due day, and if there are any production problems, we can fix them later.”
The B team’s development process consists of:

- **Strong customer focus team**: The B team maintains a group of people, called Customer Focus Team (CFT), to deal with customer.
- **Minor project planning and management**.
- **Minimum project tracking and oversight**.
- **Weak configuration management**: Work product was delivered to production from individual’s personal file(s).
- **Incomplete quality gates certification**: Quality Gates (QGs) sign-off was performed only for business requirement and for installation that require signatures from both customer and the development organization.

Performance metrics did show that the effort the W team put into process management and improvement did pay. Comparing to what the W team achieved, the cost for B team is at least 1.4 times higher while the issue density is 94% worse.

Factors that influence the software process in this case were identified:

- **Acknowledgement for the meaning of the “customer satisfaction.”**
- **Previous experience with the software processes.**
- **Focusing on quality or something else.**

### 2.3 Changes in Leadership

Change in leadership could be either a part of the re-structure or simply a change in someone’s personal career plan. With different personal management styles, sometimes change in leadership means totally different software process in the organization.

In the case studied here, a software development group was assigned to develop a software application. The first manager of the group had over 15 years of management experience and was very process-oriented in his management style.

When the first manager retired, a new promoted manager took the leadership of the group. With additional function assigned, new staffs were brought into the group. Those new staffs were mostly younger and no previous experience of the software processes. The new manager of the group did not involve before with the CMMI certification. After the change in leadership, software project behaviors also changed.

In this case, most of the factors that impacts on software process are the same as those identified in the case of “Change of Development Structure.” However, there is one big difference:

- **Management style affects the efficiency of process execution**: The first manager focused on customer relationship and let his staffs take care their own technical responsibilities (very coarse grain macro management). On the contrary, the new manager was doing his best to tack care almost everything that is from planning down to technical details (a good example of micro management). As the results of micro management, technical decision made by staffs based on the process often superseded by manager’s judgments. Superseding itself is not an issue, but too many times the superseding was not made according to the process, but heroic actions. In other words, the manager would use what
he believes the best way to do the job learned from personal experiences, regardless what the process would recommend. This is exactly what the CMMI describes as level 1 process.

3. PROCESS ISSUES WITHIN CMMI

Among the 5 maturity levels of the CMMI, level 1 is actually not a level for certification by the SEI. The CMMI model does not prevent organizations from skipping levels when apply for certification. However, most of the organizations start at level 2. And if the organization is willing to improve on its maturity level, it would most likely go through next higher levels in sequence.

Through years of application and numerous projects/organizations appraisal assessments in both CMM and CMMI, the model has achieved its success and established the reputation as a reliable assessment method for process management and improvement. However, while the SEI enjoys the reputation and expands its reach to the rest of the world, the current version of CMMI model has some process issues to its own. These issues have resulted in negative impacts that are observed in the capability maturity of many software development organizations.

3.1 Need to Better Defining the Certification Renew Process

One of the primary issues of the CMM and CMMI V1.1 models is that there is no periodical re-assessment and SEI does not require the organization to renew its maturity level certification. Once an organization passed the appraisal and received the certification for certain maturity level, the CMMI leaves the re-assessment to the organization internal Software Quality Assurance (SQA) function and believe it should be able to help the organization to maintain the same maturity level through years of changes. Only the change in the title of the organization will invalid the CMM/CMMI level-n certificate.

Comparing with CMM/CMMI, ISO9000 imposes an annual audit after issuing the certification to an organization and requires it to be renewed at regular intervals recommended by the certification body, usually around three years. In ISO9000, if an organization failed the external annual audit twice in a row, the certificate issued to the organization may be cancelled. Even it is never reported that any ISO9000 certified organizations were revoked from their certification, it is not uncommon that some organizations failed in their external audit and were provided a second chance to correct any identified issues or problems.

Without the periodical re-assessment, the CMM/CMMI level-n certificate may not reflect the actual process maturity in an organization even just a year after the appraisal. Fortunately, SEI recently recognized the issue and established in V1.2 revision to have those CMMI appraisals expire after three years. For organizations that were certified under V1.1, they will need a SCAMP A appraisal for V1.2. However, it is not clear at this moment what type of SCAMPI (A, B, or C) will be used for the re-assessment since none organization that certified under SCAMPI A V1.2 has gone through that process yet. It is assumed that another SCAMP A is required for re-appraisal.

To go through SCAMP A every three years is a very costly and time consuming pro-
cess for any business and it is not suitable for the purpose of “renewing certification,” as comparing with ISO9000. It also raises another question that we will discuss next: what if the organization fails the re-appraisal for its current maturity level?

3.2 Unidirectional Certification Path

Unlike the ISO9000 or Six-Sigma [11] assessment that has only pass or fail, the CMMI defined 4 maturity levels of certification. And based on the assumption that organizations will continue on improving their process maturity, the CMMI defined an upward only path for maturity level certification appraisal method. In other words, the only path in the CMMI certification for an organization is to receive the level-2 first (very few jump start at level-3), and then promotes it to a higher level until it reaches level-5, if it is willing to.

This upward only path creates a process issue. That is, once an organization is issued certain CMMI level-$n$ appraisal, the only possible change to that certification is getting a CMMI level-$n+1$ and above certification. Even under the newly established re-appraisal process, the organization may not be downgraded to $n-1$ level if it fails the re-assessment. One exception is when the organization changes its name which would revoke the certificate as it is issued to the organization per the name listed on the certificate.

In a real world scenario, an organization could be satisfied at the CMMI level-2, or level-3, and would be not seeking for further appraisal. With the upward path only appraisal method, the organization does not need to worry if it would lose the maturity level certification as long as it keeps its name. But in the fast-changing world of business, an organization may take the priority on schedule or cost above the process works to avoid delay to the delivery or increasing the development cost. It just needs to maintain the process documents for re-appraisal later.

On the other hand, if the organization is restructured and renamed to accommodate the dynamics of the business, it loses the certification awarded by the SEI, even it may still maintain the same level of the process maturity. This issue has kept many organizations stay at the CMM/CMMI level-2 or level-3. As many organizations may have been awarded the CMMI level 3 and above, but they tend to be working toward the CMMI level-2 again after the restructure effort.

3.3 Insufficient Process Practices in Sub-contract Management

Current trend of outsourcing and off-shore creates a new business dynamic in software development sector. As mentioned earlier that some IT sub-contracts require the contractor to be at certain CMMI level, this has forced many IT consultant firms to pursue the CMMI certification. As a matter of fact, quite a few firms (even outside of USA) have been issued the CMMI level-4 or even level-5 maturity level and they use it to assure the customers their capability in software development.

In this business dynamic, there is less problem in the case that the subcontractors have lower maturity levels than that of the acquirer. Defined in CMMI-ACQ V1.2 model and a guidebook for acquirers [12], the subcontractor simply follows the processes defined by the acquirer. However, problems occur in the case that the sub-contractor has a higher CMMI maturity level than the organization that outsources or off-shore the works.
When that happens, it is always a struggle for the two organizations on who would follow whose process and what process works needed to be done to keep the project scope, schedule, and cost intact.

Although the CMMI-ACQ V1.2 model provided some guidelines to acquisition organizations for initiating and managing the acquisition of products and services, it was never defined any practices for the above mentioned case. The CMMI-ACQ model is primary in the view of the acquirer and leaves the organizations with flexibility of negotiation. But the flexibility can be a process issue in a sense that the higher maturity level of the sub-contractors may have to be compromised to be able to work with the lower maturity level of the acquirer.

This insufficient process practices in sub-contract management has posted a significant negative impact on organizations that holds the CMMI certificate, especially at the level-4 and level-5. When those level-4 and level-5 organizations receive contract from organization that has lower maturity level, they have to adjust their own process works to comply with the acquirer’s organization’s processes.

Along with the process issues described in the previous sections, this process issue degrades the actual process maturity of the sub-contractor organization when its staffs are used to the process works of the lower maturity level and become unfamiliar with its own process of higher maturity level.

### 3.4 Rigid Process Adherence to Homogeneous Standard Processes

As the CMM model was originated from an US Air force funded project, the concept behinds the model embedding with some rigorous process assessment for the military to use as an objective evaluation of software subcontractors. Even it expanded its scope beyond the military use, the CMMI inherited such attributes and maintained the requirements in its SCAMPI. In the SCAMPI, a mandated organization standard process must be institutionalized as a pre-condition of level-2 certification.

However, the commercial software development environment is significantly different from what that is in military. James Bach [13] commented the CMM model: “The model is based mainly on experience of large government contractors, and Watts Humphrey’s own experience in the mainframe world.” Even today, we may still say that the CMMI model is more suitable for long term software development such as those in government contracts. In such cases, the software has much more stable objectives and requirements.

Even the improved CMMI model cannot address all of the challenges in the commercial software development environment. Keith Lutz reported the Intel IT Flex Service’s adoption of CMMI demonstrates that it is not a one-size-fits-all model, nor is there one “right” approach to implement it, even within a single organization [14].

For commercial software application development, one of the challenges is the constantly changing business environment that will impact on the scope and requirement of the software even during the development cycle [15]. In electronic business era, software development, either for Web 2.0 or Service Oriented Architecture, must accommodate the pace of business.

The volatility of scope/requirements in business software development is one major foe to the process works when the Time-To-Market schedule is placed as higher priority than any other factors to be considered for the success of the business.
The CMMI model assumes that stable (fully balanced scope, cost, and schedule) medium to large projects are the common case in an organization. Although the CMMI allows any small development to have tailoring and waiver from some of the process works, the SCAMPI does not allow multiple organization standard processes. The homogeneous view of organization processes prevents many software projects from utilizing different development paradigm best fit for them. In many cases, the time consuming process work products (usually documents that are overlapped and sometimes beyond the project deliverables) are prepared just for SQA review, but not for the purpose of software development process improvement.

3.5 No Bottom-Up Integration of Actual Personal/Team Maturity

Obviously, CMMI focuses on organization but not on people, especially not on leadership as pointed out in [10]. It was also mentioned in [13]: The CMM reveres process, but ignores people. The process issue is due to a concept that the organization standard processes must be institutionalized and is mandated for every staff to follow. In the CMMI concept, the organization standard processes should not be too sophisticated so an average staff can understand and complete the process work product.

Trying to address this issue, SEI developed the Personal Software Process, PSP, and the Team Software Process, TSP [16], as a self-improvement process for software engineers and teams. The PSP/TSP provides a process that aims for the quality of the software, for software engineers and teams. It recommends some defined practices intended to assist software engineers and teams to improve their software quality.

While pushing to integrate PSP/TSP into the CMMI [17], there could be conflicts between mandated organization standard processes and a preferable personal and team processes. Because the SCAMPI looks for the institutionalization of organization standard processes for assessment, personal and team processes have no value to the organization maturity level determination if it is not aligned with the standard processes. With the top-down view of PSP/TSP in CMMI, even the most intelligent software engineer in the organization cannot contribute his/her own personal process to the organizational standard process as average software engineers may not be able to follow it.

Another issue for CMMI is on organization leadership. As a mid-level management put it “I don’t do CMMI,” organization processes is for staff, but not people manager, to follow. Lack of financial support for process related activities was a major complaint in earlier CMM appraisal.

Although CMMI added management awareness into SCAMPI, budget and schedule are more likely to get into management’s priority list in business. The only management awareness might be when the organization can achieve its desire maturity level. Beyond that interest, processes are for staff to follow as long as no SQA report indicating the organization under his/her power fails to observe them.

4. REMEDIES

Remedies for those process issues are proposed in this section for possible revision in the next version of CMMI model. The mapping of remedies with process issues is shown in Table 1.
Table 1. Mapping of process issues and remedies.

<table>
<thead>
<tr>
<th>Issue Description</th>
<th>4.1 Yearly Re-assessment Using Appropriate SCAMPI Requirements</th>
<th>4.2 More Detailed Processes Definition on Sub-contract Management</th>
<th>4.3 Multiple Options for Processes in Organization Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Need to Better Defining the Certification Renew Process</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2 Unidirectional Certification Path</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3.3 Insufficient Process Practices in Sub-contract Management</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4 Rigid Process Adherence to Homogeneous Standard Processes</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3.5 No Bottom-Up Integration of Actual Personal/Team Maturity</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

4.1 Yearly Re-Assessment Using Appropriate SCAMPI Requirements

For issues 1 and 2 in previous section, a new SCAMPI method that falls between current SCAMI B and C should be developed for organization re-assessment annually. The yearly assessment can be carried out by appraisers both within and outside of the organization. Lead appraiser may not be necessary for the re-assessment if the organization received its maturity level or passed its re-assessment last year.

The yearly re-assessment is different from the organization’s own quarterly SQA audit. The internal SQA audits are for organization processes adherence and corrective action plans of projects in the organization. The yearly re-assessment is, however, to verify the effectiveness of process improvement as indicated by the maturity level of the organization. The finding of the yearly appraisal should be reported to SEI for record and be used for certification renew recommendation per the current three-year period.

If the organization fails the yearly appraisal, the organization should be given another year to correct the problems. In this case, a lead appraiser should be appointed for the next yearly appraisal. If the organization fails twice in a row, its maturity level should be revoked and downgraded to next low level until reaching level-1.

The certificate of paper form issued to an organization should not be the authentic evidence to certify the organization’s CMMI maturity level. The SEI should instead maintain an online official list of organizations that are assessed for every CMMI level (currently, SEI only maintains on its web site an incomplete list.)

4.2 More Detailed Processes Definition on Sub-contract Management

For issue 3, the CMMI-ACQ needs to define more practices details for an acquirer whose supplier or sub-contractor has higher maturity level. Suggestion for this remedy is that if the acquirer sub-contracts portion of the work to another organization with higher
maturity level, that portion of the work should require all process works associated with such level. In other words, its processes beyond the acquirer’s standard processes should be part of contract requirements.

The remedy is based on the concept that, if the maturity model stands for its claim, the higher maturity level organization should be capable of doing better than the one that is at the lower maturity level. While an organization should not copy the processes from others, it may be wise to leave the choice of processes to the one who is actually doing it and has better capability.

This will greatly help the supplier or subcontractor to maintain its maturity capability while working on a contract that requires a lower maturity level. Meanwhile, the acquirer may also be benefited from the subcontractor’s process capability and further improve its own processes while maintain such relationship.

While both the supplier and subcontractor can improve on their process by learning from another, they will be also able to perform their work based on their own process. Because the increasing trend of outsourcing and offshoring, software industry will soon see the benefits if SEI implements this remedy in next revision of CMMI guidelines.

4.3 Multiple Options for Processes in Organization Process Areas

The maturity level certification could be issued after an appraisal to organizations from one that is as small as less than 100 people, to one that has several thousand. Hence, a homogeneous process mandated for every project in a large organization tends to create more problems than benefits.

Within a large organization, people in some small projects may find a different process that works better for them than the mandated standard processes. For agile development, the best process maybe different from the functional development. Moreover, the creativity of human being may find a process work best for a few people, while another ingenious way is invented by someone else within the same organization.

Current complicated mechanism of tailoring and waiving in CMMI guidelines makes it too difficult for software project teams to grant approval from senior management for adopting any process that differs from the mandated one. Senior management may not know what process is best for the specific project team, but they must know the cost of getting that CMMI maturity level certification. Hence, the senior management would just determine that, per the CMMI guidelines, if the organization standard processes are there to be followed, those must be followed. Granting any tailoring and waiving just create troubles to the organization, the senior management would believe.

Having consistent organization processes is good for minimum performance. However, innovation and creativity is how software development different from manufacturing line. The CMMI model should recognize the fact that meeting the requirements of process areas does not guarantee the software quality. How to integrate personal process from bottom up into organization processes and allowing two or more processes defined from one process area in an organization are two key challenges for the CMMI revision.

A remedy may be to allow branches to the organization standard process, instead of tailoring and waiving from it. As tailoring and waiving is an exception from the standard process, branches are part of the standard process. So, a project team can choose a branch that best fits for them and still comply with the process audit.
5. CONCLUSION

The CMMI is a very good model for software development process management and improvement. When it comes to its implementation, however, many embedded process issues in the model result in degradation in effectiveness and efficient. The activity-based pattern at the level-2 and level-3 provides clear path for organizations to implement their own processes. But the same pattern also prevents good (sometime best for specific team) process practices from being utilizing by project teams in the organization.

The process issues described in the paper have been observed and identified as negative impacts in many software development organizations. With the latest CMMI V1.2 issued in August 2006 and the CMM phased out in 2007, it will take some time for the SEI to address those issues.

Despite that factor, the CMMI V1.2 can be improved by removing those process issues so the model can maintain its creditability. Otherwise, when the effectiveness of those organizations that received CMMI maturity level certification faded along with time, the more organizations received CMMI certification, the more doubts will be presented in the software industry. Moreover, the CMMI should implement a process improvement approach on itself to correct process issues.

ACKNOWLEDGMENTS

The author would like to express his gratitude to Dr. Jack Ryder of Kean University for his valuable comments and helps in refining a few concepts in process improvement.

REFERENCES

Jing-Chiou Liou (劉景秋) received his B.S. degree in Electronic Engineering from National Taiwan University of Science and Technology in 1983 and both M.S. and Ph.D. degrees in Electrical Engineering from New Jersey Institute of Technology in 1993 and 1995, respectively. He is currently an Assistant Professor in the Department of Computer Science and Information Technology at Kean University, New Jersey, U.S.A. Prior to joining the Kean University, Dr. Liou had worked for software industry, including AT&T, IBM, and Lucent Technologies, for 15 years. His experiences in software industry cover project management, software process improvement, and quality assurance. Dr. Liou is a member of IEEE Computer Society, and is among the Technical Committee on Software Engineering of IASTED and a member of the International Committee on AICT’s International Conference Series. Dr. Liou’s current areas of research focus primary on software engineering and software security.