Multi-Model Performance Improvement in Action

Jozef De Man
Alcatel-Lucent and Ghent University, Belgium

Abstract

In this paper we explain a simple general framework for dealing with a combination of models, techniques and methods, based on the CMMI®. The framework builds on recent work in this area and includes some important enhancements. We also show with real-life examples in our organization how this framework is used for introducing new methods and techniques building on the accomplishments of earlier and ongoing improvement programs instead of throwing them away. The framework uses the well-known distinction between what-models and how-models (techniques and methods). The CMMI is used as the central what- model. How-models are mapped to the CMMI practices, process areas, categories and maturity levels and these mappings are compared to determine scope and show how these models can coexist and reinforce each other. Practical examples include Scrum, stage-gate product development, the DMAIC life cycle of Lean Six Sigma, supporting techniques of Lean Six Sigma.

Background

In this section we briefly explain how our work is positioned in the landscape of multi-model research.

Three broad categories of multi-model initiatives can be distinguished [7]:

- Harmonization: modification of a model to align one or more characteristics (e.g. terminology, structure, granularity) with another model. The models remain separate but their combined use is facilitated.
- Integration: two or more models are replaced by a single model that combines the benefits of the original models. The original models are typically replaced by the integrated model. The CMMI is an example.
- Mapping: establishing a mapping between elements of one model and elements of another model or a taxonomy. Halvorsen and Conradi identify four types of mapping: characteristics, framework mapping, bi-lateral mapping and needs mapping [5].

In this paper we are not dealing with harmonization and integration, but with mapping of models.

When establishing a multi-model approach it is necessary to think in terms of a hierarchy of models where one model can be an instantiation of another. At the top of the hierarchy we find the what-models. They define the goals that must be achieved and the requirements that must be satisfied by more concrete models. The how-models provide concrete guidance on how the requirements can be satisfied. Several how-models can coexist, each with a specific domain of application. We can refer to these how-models as techniques or methods. Organizational processes can be thought of as lower level models implementing methods and techniques. The defined processes of projects created by tailoring of standard models are again more concrete models. Finally, the enactment of the defined process in a project is the lowest level model. See Figure 1.

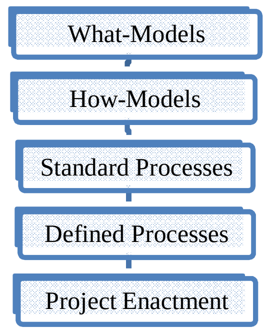

Figure 1 Model Hierarchy

Appraisals using the SCAMPI [10] can be thought of as establishing a mapping between the enactment of projects and the goals and practices of the CMMI. A tailored SCAMPI C can be used to establish the mapping between a process definition and the CMMI.

The mapping between models differs substantially depending on whether they are at (approximately) the same level of abstraction or not. If they are not, the mapping defines which elements of the concrete model are instantiations of elements (requirements) of the more abstract model. If the two models have the same level of abstraction, the mapping will determine elements that can be considered equivalent or may conflict with each other.

Maps can be defined bilaterally between all models of a set. It is also possible to map all models of interest to a common target. That reduces the number of maps that needs to be considered in a multi-model context. The target model can be an existing model or one that is defined specifically for the purpose of comparing the other models.

For example in [13] an ad hoc classification in three areas is proposed: good practice elements, improvement methods elements and institutionalization elements. The good practice elements define what an organization wants to improve. The improvement methods are used to implement the improvements. The institutionalization elements must ensure that the improvements become ingrained in the culture of the organization, are sustained and are not abandoned when a crisis occurs. The generic practices of the CMMI are probably the best examples of explicit guidance for institutionalization.

In this paper we are concerned with the mapping of models to a common target model. We have limited the scope further by considering the interaction between how-models. Because of this particular focus we can conveniently use an existing what-model as the common target model.

Our work has been strongly influenced by Mike Phillips' column of the SEI web site "CMMI with Agile, Lean, Six Sigma, and Everything Else" [8] presenting the idea of using a common what-model to understand how multiple techniques and methods (how-models) can be used together.

The CMMI as a multi-model framework

As a framework for understanding the relationship between how-models we use a mapping of each of these models to the CMMI as the common target what-model [1].

The structure of the CMMI is illustrated in Figure 2. Process areas are clusters of related practices. Specific goals and practices are specific for a process area. Generic goals and practices apply to all process areas and contribute to the institutionalization of the processes associated with a process area.

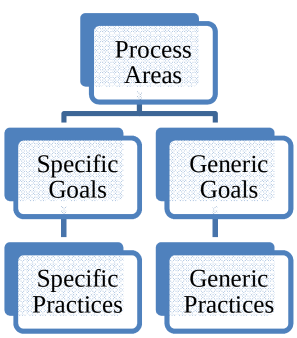

Figure 2. CMMI Structure

The CMMI comes with two representations. The continuous representation is used to determine individual process area capability. In the continuous representation process areas are grouped in four categories: Process Management, Project Management, Engineering (in the CMMI for Development) and Support. The staged representation is used to determine the process maturity of an organizational unit and groups process areas in maturity levels from 2 up to 5.

There are several reasons for selecting the CMMI as the common what-model.

- The CMMI has been carefully designed as a what-model. The definition refers to many good practices, but in the informative components of the model.
- It probably offers the best support for process institutionalization by means of the generic goals and practices. Moreover, the generic practices (with one exception) are each supported by one of more process areas. For example, the generic practice 2.2 Plan the Process is supported by the Project Planning (PP) process area [1].
- The categories of the continuous representation match many of the ad hoc classifications that are proposed in a multi-model context (see an example below).
- The CMMI involves a common foundation of core process areas that can be applied beyond the currently defined constellations of Development, Services and Acquisition.

We will typically start with mapping the elements of a method to the practices of the CMMI. In most if not all cases we can rely on existing maps available in the literature. From that detailed mapping we identify the CMMI process areas addressed by the method. That map can be conveniently presented in a table of the process areas organized in columns by the categories of the continuous representation and in rows by the maturity level of the staged representation. At a glance we can see the scope of the method, the categories that are addressed and the maturity level at which it can most effectively be deployed.

	Process Mgmt	**Project Mgmt**	**Enginee-ring**	**Support**
ML 5	OPM			CAR
ML 4	OPP	QPM		
ML 3	OPF OPD OT	IPM RSKM	*RD TS PI VER VAL*	DAR
ML 2		**PP PMC REQM** SAM		PPQA CM **MA**

Table 1 – CMMI process area map for Scrum

See Table 1 for an example of a CMMI process area mapping for Scrum based on [9]. The process areas with specific practices addressed by the method are in bold: Project Planning, Project Monitoring and Control (PMC), Requirements Management (REQM) and Measurement and Analysis (MA). Process areas with generic practices address by the method are in italics. That are the process areas of which the generic practices are supported by the process areas in bold. In the example, the Engineering process areas Requirements Development (RD), Technical Solution (TS), Product Integration (PI), Verification (VER) and Validation (VAL) are supported. The mapping can be refined by recognizing that MA is only addressed to the extent that it supports the generic practices of PP, PMC and REQM.

If we compare our approach with the 3 category classification of [13], we find the good practice elements in all categories, the improvement method elements in the process management category and the institutionalization elements implicitly addressed through their supporting process areas. In the example, Scrum is addressing the generic practices of the engineering process areas that are supported by the applicable process areas in bold, PP, PMC and MA.

To avoid any misunderstanding, this map is not intended to be exhaustive or complete. It shows the process areas largely addressed by Scrum. By no means should it be inferred that all specific practices of the process areas are fully covered by the method. On the other hand, it can be argued that Scrum addresses process improvement with retrospectives and Quality Assurance through coaching by the Scrum Master. The mapping should be considered to be approximate.

In summary, our multi-model approach is based on a model hierarchy with the CMMI as highest level root model and more concrete methods and techniques at lower levels in the hierarchy. To understand the relationship between the methods and techniques we use a mapping to the CMMI and not bilateral mappings between the methods and techniques.

Obviously, the process area map cannot show the detailed mapping at the practice level. Its main value becomes apparent when multiple models are considered. By summarizing the mapping in the same picture for each model, we can more easily recognize the scope, the relationships between the models and the opportunities for cross-fertilization.

Application of the Framework

In this section we discuss the maps of some important how-models illustrating the benefits explained in the previous section.

Agile methods

Consider an organization with an established performance improvement program based on the CMMI. Some R&D departments decide to deploy agile methods and claim that the CMMI framework must therefore be abandoned. There is an extensive body of literature explaining why this is an incorrect point of view and how CMMI and agile methods can work together (see e.g. [4]). The main argument is of course that CMMI is at a higher level of the model hierarchy and agile methods can be implementations of CMMI practices in a particular context. In the previous section we used Scrum as an example to demonstrate the tabular presentation of the map. It shows at a glance the rather limited coverage of Scrum. It is mainly concerned with the project management category at maturity level 2. As mentioned earlier, the map is approximate but it clearly shows that Scrum cannot replace CMMI as a framework for process improvement. Scrum does not rely on the implementation of process areas at lower levels of maturity and can therefore be introduced in a low maturity organization. In combination with implementation of the support process areas Configuration Management (CM) and Process and Project Quality Assurance (QA) it can cover the maturity level 2 process areas.

Stage-gate Project Development

Can agile methods be combined with stage-gate project management? A stage-gate development model describes the progression from an idea to a product through a number of stages. Between each stage, a gate review is held with all stakeholders to decide whether to proceed to the next stage or not, considering risks and the impact on the portfolio of ongoing projects. The iterative life cycle of agile methods could easily be regarded to be at odds with a waterfall-inspired stage-gate approach. When we established the map of the stage-gate Product Life Cycle used in our company [2] it appeared to address mostly practices of Integrated Project Management (IPM) at maturity level 3 (Integrated Plan, Coordination and Collaboration with Relevant Stakeholders). The IPM practices are supporting the engineering process areas. The map is summarized in Table 2. Comparing with Table 1, there is no immediate evidence of a possible conflict between agile methods and a stage-gate project management approach because they address the specific practices of different process areas (in bold) . This is consistent with results reported in the literature [6].

	Process Mgmt	**Project Mgmt**	**Enginee-ring**	**Support**
ML 5	OPM			CAR
ML 4	OPP	QPM		
ML 3	OPF OPD OT	**IPM** RSKM	*RD TS PI VER VAL*	DAR
ML 2		PP PMC REQM SAM		PPQA CM MA

Table 2. CMMI process area map for stage-gate

Lean Six Sigma

Lean Six Sigma (LSS) refers to the combination of two related but different approaches: Lean and Six Sigma. The main focus of Lean is reducing waste and improving speed while Six Sigma aims at reducing variation based on a statistical analysis of process performance. Lean improvement projects are typically shorter and can be implemented with limited training. Six Sigma projects are typically longer, require more resources and extensive training of project participants. Lean Six Sigma originated in manufacturing but can be broadly used for improving any type of process. Our framework still remains valid though because most of the process areas in Process Management, Project (Work) Management and Support are globally applicable.

Lean Six Sigma involves a comprehensive collection of tools and techniques. It is therefore not practical to establish a complete mapping between LSS and the CMMI [11]. It is valuable though to define the mapping for some of the main components.

DMAIC (Define, Measure, Analyze, Improve, Control) is one of the life cycles associated with Six Sigma. A simplified version can be applied to Lean projects as well. The DMAIC life cycle involves toll gates at the end of each phase that are very similar to the gates of the stage-gate project life cycle. At the toll gates, stakeholders agree that the conditions for moving to the next stage are satisfied and secure resources for that next phase. The DMAIC toll gates therefore map to Integrated Project Management (IPM) as well. Of course, Lean Six Sigma is all about the Process Management process areas but the DMAIC toll gates support the generic practices of these process areas through implementation of IPM specific practices. That is represented by means of Table 3, similar to Table 2 but with the Process Management process areas in italics instead of the Engineering process areas. So, although DMAIC and stage-gate address the same specific practices (in bold), the specific practices support different process areas. The two life cycles are therefore complementary.

The common reference to IPM helps us identify a quick-win in deploying DMAIC. In the company, we have a project support environment deployed since many years to support product development [3]. It is a simple tool to globally track gate review dates and their outcome, actors of the core project team and references to the most important project work products. The environment can be tailored to accommodate the tailoring of the Product Life Cycle in product divisions. That facility can now be exploited to implement tracking of the DMAIC life cycle by simply changing the definition of the milestones, roles and associated work products.

	Process Mgmt	**Project Mgmt**	**Enginee-ring**	**Support**
ML 5	*OPM*			CAR
ML 4	*OPP*	QPM		
ML 3	*OPF* OPD OT	**IPM** RSKM	RD TS PI VER VAL	DAR
ML 2		PP PMC REQM SAM		PPQA CM MA

Table 3. CMMI map for DMAIC Toll Gates

Lean techniques typically map to the Organizational Process Focus (OPF) process area at maturity level 3 and the Support process areas. The common tool of Root Cause Analysis maps to Causal Analysis and Resolution (CAR). Techniques specific to Six Sigma often use statistical techniques and map to the higher maturity process areas Organizational Process Performance (OPP), Quantitative Project Management (QPM) and Organizational Performance Management (OPM) [11]. The map confirms the complementary nature of the two approaches and also suggests that Lean can be deployed in low maturity organizations but Six Sigma becomes more effective in higher maturity organizations. However, that does not mean that many of the tools in the LSS kit cannot be applied in lower maturity organizations [11]. That is consistent with a paragraph in the purpose statement of Causal Analysis and Resolution (CAR): "The specific practices of this process area apply to a process that is selected for quantitative management. Use of the specific practices of this process area can add value in other situations, but the results may not provide the same degree of impact to the organization's quality and process performance objectives."[1].

With the comprehensive tool kit of Six Sigma we are hitting the limitations of a CMMI-based approach. Some tools can be mapped to process areas: Failure Mode and Effect Analysis maps to Causal Analysis and Resolution, Pugh Diagrams map to Decision Analysis and Resolution. Many other tools may not map to a specific process area but could still be mapped to an intersection of a category and a maturity level. An example is the mapping of control charts to Support/ML4.

Agile improvements

We mentioned above how the framework helped us in identifying an opportunity for deployment of an existing tool in the new context of DMAIC toll gates.

Here is another interesting example. We discussed above that agile methods are mainly about project management practices applied to engineering process areas. We also mentioned that a different life cycle was applied to engineering projects (stage-gate) then improvement projects (DMAIC). Why would it not be possible to apply the *same* agile techniques used for engineering projects to improvement projects? As a matter of fact several of the methods are applicable. As a result we now use daily scrum meetings for improvement projects and maintain a backlog of suggestions for improvements inspired by the agile product backlog. See Table 4.

	Process Mgmt	**Project Mgmt**	**Enginee-ring**	**Support**
ML 5	*OPM*			CAR
ML 4	*OPP*	QPM		
ML 3	*OPF* OPD OT	IPM RSKM	RD TS PI VER VAL	DAR
ML 2		**PP PMC REQM** SAM		PPQA CM **MA**

Table 4. Map of Scrum for process management

The idea touches on a very important aspect of process improvement projects by contributing to the institutionalization of performance improvement processes. By applying practices of PP and PMC to improvement projects many of the capability level 2 generic practices of the process management processes can be reinforced.

Remarks

Our use of the CMMI and maps to process areas must be considered with care. When a process area appears in bold for a given technique it does not imply that the implementation of that technique will result in the adequate implementation of all practices of the process area. The mapping implies that a considerable number of the practices are addressed by the technique but not necessarily all. The high-level analysis enables to identify the scope of the techniques and potential areas of conflict with other techniques. A deeper analysis to the practice level will be necessary to understand the interaction at a more detailed level.

Techniques that map to the same process areas (bold) and support the same process areas (italic) are not necessarily in conflict with each other. A deeper analysis must reveal whether they have the same domain of application or not. It may very well be the case that the techniques apply to projects with different characteristics. For example, some types of projects may better use a waterfall life cycle, other types may better use agile life cycles.

Conclusion

We have presented a simple framework for comparing how-models by mapping them to the CMMI as a common what-model. A tabular presentation of the maps showing process areas and the classification in categories and maturity levels enables quick evaluation and comparison of many techniques at once. The framework can build on the results of bi-lateral mappings that have been reported in the literature and enables a more comprehensive analysis by combining these mappings in a standard picture.

By building such a combined map, an organization can identify areas that are not yet covered by the methods and techniques it has deployed and focus new initiatives in these areas.

The CMMI is successfully used to appraise organizations, identify the areas where improvement actions can be implemented with highest leverage based on an understanding of the current capability profile of the organization. However, the focus is on identifying *what* must be improve and not *how*. It is very important to understand that the CMMI is not providing the good practices or methods for performance improvement. Improvement initiatives often fail because this is not sufficiently understood. In this paper we have shown how the mapping of good practices and improvement methods to the CMMI structure can build a bridge between appraisals and action planning for improvement.

References

[1] Chrissis, M.B., Konrad, M., Shrum, S. (2011) *CMMI for Development®: Guidelines for Process Integration and Product Improvement.* Addison-Wesley.

[2] De Man, J and Ebert, C. (2003) *A common product life cycle in global software development.* Proceedings of the 11th Annual International Workshop on Software Technology an Engineering Practice, STEP 2003, Amsterdam, The Netherlands.

[3] De Man, J (2007) *A Web 2.0 Project Support Environment*. International Symposium on Management, Engineering and Informatics, Orlando, FL, July 2007.

[4] Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008) . *CMMI or Agile: Why Not Embrace Both!* (CMU/SEI-2008-TN-003). Retrieved May 06, 2012, from the Software Engineering Institute, Carnegie Mellon University website: http://www.sei.cmu.edu/library/abstracts/reports/08tn003.cfm

[5] Halvorsen, C.P., Conradi, R. (2001) *A taxonomy to compare SPI frameworks.* Lecture Notes in Computer Science 2077, 217-235.

[6] Karlström, D and Runeson, P. (2005) *Combining Agile methods with Stage-gate Project Management*. IEEE Software May/June 2005.
[7] Kelemen, Z.D., Kusters R. and Trienekens, J. (2011).*Identifying criteria for multimodel software improvement solutions – based on a review of current problems and initiatives*. Journal of Software Maintenance and Evolution: Research and Practice.
[8] Phillips, M. (2008) *CMMI with Agile, lean, Six Sigma and Everything Else*. http://www.sei.cmu.edu/library/abstracts/news-at-sei/cmmiinfocus200801.cfm
[9] Potter, N and Sakry, M. (2009) *Implementing Scrum (Agile) and CMMI Together*. The Process Group Post Vol 16, No 2.
[10] SCAMPI Upgrade Team, (2011) . *Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document* (CMU/SEI-2011-HB-001). Retrieved May 06, 2012, from the Software Engineering Institute, Carnegie Mellon University website: http://www.sei.cmu.edu/library/abstracts/reports/11hb001.cfm
[11] Siviy, J., Penn, M., & Harper, E. (2005) . *Relationships Between CMMI and Six Sigma* (CMU/SEI-2005-TN-005). Retrieved May 06, 2012, from the Software Engineering Institute, Carnegie Mellon University website: http://www.sei.cmu.edu/library/abstracts/reports/05tn005.cfm
[12] Siviy, J., Kirwan, P., Marino, L., & Morley, J. (2008). Process Architecture in a Multimodel Environment. Carnegie Mellon University. Retrieved from http://www.sei.cmu.edu/library/assets/multimodelSeries_wp4_processArch_052008_v1.pdf
[13] Siviy, J., Kirwan, P., Marino, L., & Morley, J. (2008). Improvement Technology Classification and Composition in Multimodel Environments. Carnegie Mellon University. Retrieved from http://www.sei.cmu.edu/library/assets/3.pdf