QFD Application in Software Process Management and Improvement Based on CMM

Xiaoqing (Frank) Liu, Yan Sun, and Gautam Kane
University of Missouri – Rolla
Computer Science Department
fliu@umr.edu

Yuji Kyoya and Kunio Noguchi
Software Engineering Center
Toshiba Corporation
Japan

Abstract

Capability Maturity Model (CMM) from Software Engineering Institute has been used successfully by many organizations for software process improvement. However, there exists a disconnection between business goals and maturity levels. A new framework using Quality Function Deployment (QFD) is developed to deal with this problem. This framework serves three purposes: (1) it provides a connection between business requirements and CMM; (2) it proposed a methodology for the priority assessment of requirements from multiple perspectives; and (3) it helps identify a set of software process improvement actions based on business requirements and CMM.

1. Introduction

In this era of rapid technological innovation and changes, the key to the survival of a software company relies on the continuous improvement of its process. When talking about Software Process Improvement (SPI), many of the software development organizations think about existing models and standards, such as ISO 9000 series of standards, CMM [2,3], etc. Among all these models, it is unfair to make a judgment on which one is better. However, considering the more detailed guidance and greater breadth provided by CMM, it may be a better choice for some software development organizations [4].

Although Capability Maturity Model Integration (CMMI-SE/SW) has been developed to solve the problem of using multiple CMM models for different areas of application, CMM has been working so well in many organizations that they do not have plans to replace CMM with CMMI immdediately. However, similar to the other SPI models/standards, it does not address the connection between its maturity levels and the business goals of an organization. In other words, an organization cannot see an explicit reason to reach a higher maturity level in terms of satisfying business goals. In addition, like all other standards and models on SPI, CMM addresses "what to do" while leaving "how to do" to organizations. Therefore, some methodology is needed to transform CMM activities into actions which are detailed enough to follow by software engineers.

This paper develops a framework which helps map business and other requirements to CMM, and helps develop action plans to satisfy those requirements and CMM goals using QFD which is a methodology for building the voice of the customer, both spoken and unspoken, into a product. Unlike traditional quality systems which aim at minimizing negative quality in a product, QFD adds values to the product by maximizing the positive quality [5]. Nowadays, QFD has been applied to virtually every industry and business, including software development [14]. However, one of the important components of QFD, which focuses on improving the process quality by assuring that organizational processes and actions are in compliance with established standards, has been neglected by most QFD followers in the business [6].

Several attempts have been made to integrate QFD into SPI. Ita Richardson proposed a four-stage model for software process improvement in small companies [8,11]. The measurements in the model are based on self assessment of software process. This model is unsuitable for large companies because self-assessment is difficult across groups.

3-WoSQ '05, May 17, 2005, St Louis, Missouri, USA.

Zultner's Business Process Reengineering model with QFD [7] uses either the major competitor's performance or creative thoughts of employees, but not existing standards, as the source of process improvement. This may cause inefficiency or difficulty in replicating this approach in different situations or environments.

SAP also uses QFD in SPI [9]. Although this approach considers the participation of multiple stakeholders, it treats requirements from all stakeholders as equally important and it does not consider the relationships between multiple perspectives. In addition, the process improvements actions are directly related to process requirements. No standards, such as ISO or CMM, are considered throughout the workflow. Therefore, although the action plan is prioritized, it may be unreliable.

2. Our research approach

Our approach develops a methodology which derives action plans based on software process requirements and in accordance with CMM because CMM is still being used by many organizations. In our framework, QFD is used to help an organization achieve three objectives. Firstly, business and other requirements within an organization are mapped to CMM goals and activities. A connection is established so that the organization can see clearly how CMM helps with its business. Secondly, software process requirements from multiples perspectives are prioritized so that requirements with more and stronger impacts on other requirements can receive higher priority values. Thirdly, QFD helps transform requirements of the organization into process actions through Key Process Areas (KPAs). This guarantees that the actions are in accordance with CMM and, at the same time, satisfy the process requirements from the organization. This directly results in the improvement of the organization process.

The framework is designed in such a way that the requirements from a particular perspective are prioritized within perspectives. At the same time, each perspective carries its own priority. The requirements from multiple perspectives are correlated with each other. As a result, the priority value of each requirement is adjusted after their impacts of requirements from other perspectives are assessed.

These set of requirements with adjusted priorities are related to key goals in CMM KPAs. The goals are prioritized based on process requirements. Hence the goals which achieve higher overall satisfaction of process requirements get higher importance. In order to achieve these goals, CMM has Key Practices (KPs) categorized into five common features. These KPs in each category are prioritized separately based on the priorities of goals. KPs which aim to achieve higher overall satisfaction of goals get higher importance. A separate set of action plans is derived from KPs in each of the common features. The actions which help to support more important key practices get higher priorities. As a result, the action plan follows the standards in CMM and corresponds to process requirements, and actions with higher importance help to achieve higher stakeholder satisfaction.

The remainder of the paper is structured as follows. Section 3 describes our software process improvement framework. The detailed matrices development used in the framework is explained in section 4. Section 5 introduces variations of the framework. An SPI example using our framework is summarized in section 6, and the concluding section provides a discussion on the significance.

3. Framework for software process improvement using QFD

This SPI framework uses CMM model as its foundation and QFD as its tool. The advantage of this framework is that the requirements from various perspectives are incorporated into the final set of actions which are detailed enough to be performed.

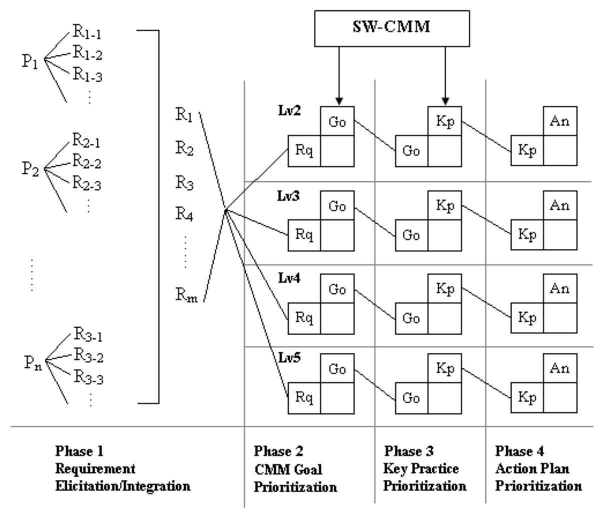

Figure 1. Software Process Improvement through CMM Using QFD

In the first phase of the framework, the requirements toward the organizational process are gathered from various branches/departments. In Figure 1, various perspectives are represented as P_1 through P_n. Each perspective contains multiple requirements. These perspectives are then prioritized based on their relative importance within the organization and integrated into one single set of requirements. The prioritization ensures that requirements from different perspectives are comparable with each other and the integration reflects their correlations.

The second through the fourth phases of this framework are applied to level 2 to 5 of CMM. In the second phase, goals of all KPAs in a CMM level are selected and prioritized based on the requirements from the previous phase. A house of quality is used to establish relationships between requirements from the organization and KPA goals from CMM. This house of quality demonstrates that complying with CMM standard also helps satisfy the business and other requirements in the organization.

The third phase of our framework involves the prioritization of KPs within all KPAs of a specific level. The prioritization is based on the deliverables from the previous phase. In order to incorporate requirements into final action plan, these KPs have to be prioritized based on KPA goals which are now reflecting requirement priorities.

In the fourth phase of our framework, a set of actions are derived from the prioritized KPs. These activities reflect the requirements integrated in the first phase. Meanwhile, they also state what needs to be achieved in order to reach a particular maturity level. These action plans become the guideline of the process improvement. Thus, more resource should be assigned to those actions with high priorities.

4. Matrices in our framework

Four different matrices are used in the four phases of our framework.

4.1 Phase 1 – Requirements Impact Matrix

We use a relationship matrix called Requirements Impact Matrix (RI Matrix) to help with the requirements integration. But before that, a requirements prioritization technique can be used to integrate all requirements into one single set. The following steps are followed to produce perspective weights and requirement local priorities.

1. Establish a linkage between each pair of perspectives in the set by identifying the relative dominance values $rd_{1,2}$ between perspectives 1 and 2 using the following equation [13]:

   $$rd_{1,2} = n_2/n_1$$

   where: n_1 is satisfaction degree of perspective 1,
   n_2 is satisfaction degree of perspective 2.
2. Calculate the local priority of each perspective. For n perspectives, P_1, P_2, ,…, P_n, in a decreasing order of importance, let $rd_{i,j}$ denote the relative dominance of the perspective P_i over P_j, and let W_{Pi} denote the numeric priority of perspective R_i. First, P_n is assigned with a priority (W_{Pn}) of one. All remaining perspective priorities can be determined recursively using [13]:

   $$\text{For } 1 \leq i < n,\ W_{Pi-1} = W_{Pi} * rd_{i-1,i}$$
3. Within perspectives, requirements are prioritized using the same method described in steps 1 and 2 to derive the local priorities.

Local requirements priorities are normalized within perspectives before being multiplied by its perspective weight to produce initial global priorities [12,13]. These initial global priorities of requirements are used in the RI Matrix for requirements integration.

The following steps calculate weighted priorities of requirements from one perspective with effects from another using RI Matrix as shown in Figure 2.

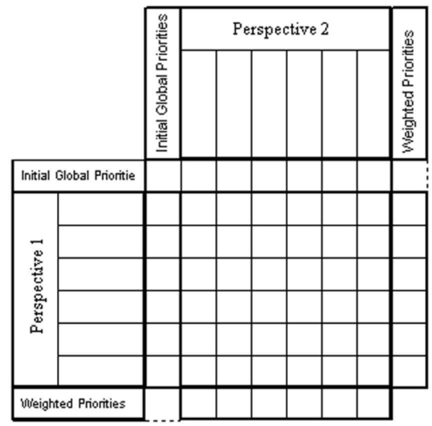

Figure 2. Requirements Impact (RI) Matrix

1. Enter all requirements from one perspective with their initial global priorities into rows.
2. Enter all requirements from another perspective into columns, with their initial global priorities entered in the cells immediately below them.
3. Determine the impact relations between requirements from different perspectives. The types of impact relationships are shown in Table 1.

Table 1. Types of impact relationships

Impact	Symbol	Value
Strong	●	9
Medium	▢	3
Weak	▽	1

4. Calculate the weighted priorities using Equation.

$$WP^{i,k}_j = W^i_j * \sum_{l=1}^{M_k} W^k_l * IR(R^i_j, R^k_l) \qquad \text{(Equation 1)}$$

where: $WP^{i,k}_j$ is the weighted priority for requirement j in perspective i with effects from perspective k;
$IR(R^i_j, R^k_l)$ is the impact relationship between requirement j from perspective i and requirement l from perspective k;
W^i_j is the initial global priority of requirement j from perspective i.

An RI Matrix is constructed between each pair of perspectives. After all RI Matrices are constructed, the summations of all weighted priorities (WP) for the same requirement become the final weight. These final weights are normalized into normalized weights (NW) before being combined with the initial global priorities using an adjustment factor α between 0 and 1:

$$AP_j = W_j + \alpha * NW_j \qquad \text{(Equation 2)}$$

The result from the above equation is the adjusted priority (AP) which is delivered to phase 2.

4.2 Phase 2 - Requirements-Goals Impact Matrix

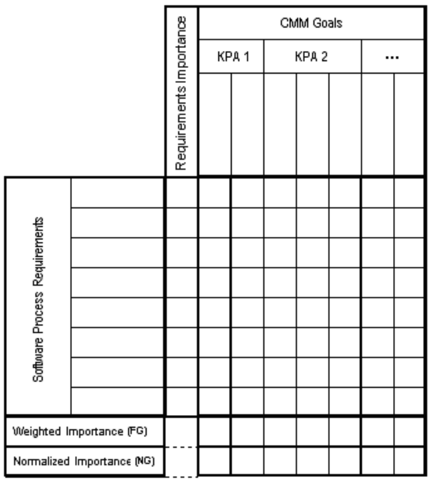

Figure 3. Requirements-Goals Impact (RG) Matrix

In this phase we use the Requirements-Goals Impact Matrix (RG Matrix) to prioritize KPA goals on the basis of the adjusted priorities (APs) from phase 1. KPA goals are listed for each KPA in CMM.

Figure 3 illustrates the components of an RG Matrix. The following five steps guides through the process of building the RG Matrix.

1. Enter integrated requirements (from previous phase) and their adjusted priorities into rows.
2. Enter goals in all KPAs of a particular maturity level into columns.
3. Determine the impact relations between each requirement and each KPA goal. The same set of symbols in Table 1 is used in this matrix.
4. Calculate the weighted importance values (FG) using formula:

$$FG_i = \sum_{j=1}^{M} AP_j * IR(G_i, R_j) \quad \text{(Equation 3)}$$

5. Normalize the weighted importance values (NG).

4.3 Phase 3 – Goals-Practices Impact Matrix

Goals-Practices Impact Matrix (GP matrix) maps KPA goals to KPs. The following steps construct a GP matrix as shown in Figure 4.

1. Enter CMM goals in RG matrix together with their normalized importance values (NG) into rows of GP matrix as indicated.
2. List the KPs in the desired common feature as columns of the matrix.

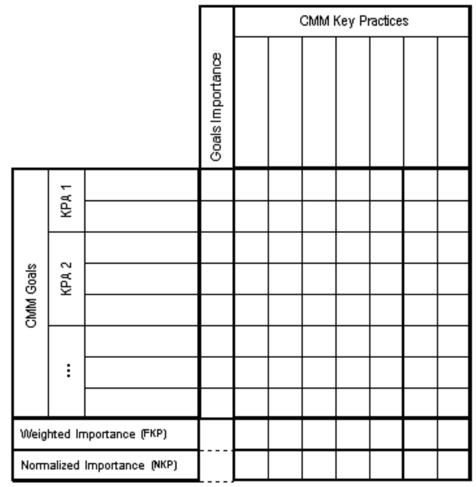

Figure 4 Goals-Practices Impact (GP) Matrix

3. Determine the impact relationship between each goal and each KP. The same symbols in Table 1 are used in this matrix.
4. Calculate the weighted importance values (FKP) of key practices using formula:

$$FKP_i = \sum_{j=1}^{S} NG_j * IR(KP_i, G_j) \quad \text{(Equation 4)}$$

5. Normalize the weighted importance values of key practices into NKPs.

4.4 Phase 4 - Action Plan House of Quality Matrix

Action plans are developed on the basis of key practices and their correlations are determined using AP-HoQ matrix. The following six steps are used to develop an AP-HoQ matrix as shown in Figure 5.

1. Enter CMM key practices together with normalized importance values (NKP) into rows.
2. Derive a set of actions from the key practices in each KPA and enter them into columns as indicated. A set of actions is derived separately for each common feature.
3. Determine the impact relationship between each KP and each action. The same set of symbols in Table 1 is used in this matrix.
4. Calculate the weighted importance of actions using formula:

$$FA_i = \sum_{j=1}^{Z} NKP_j * IR(A_i, KP_j) \quad \text{(Equation 5)}$$

5. Calculate the normalized importance of actions.
6. Determine the impacts of actions on each other in the roof. A plus sign (+) is used to indicate the existence of impacts.

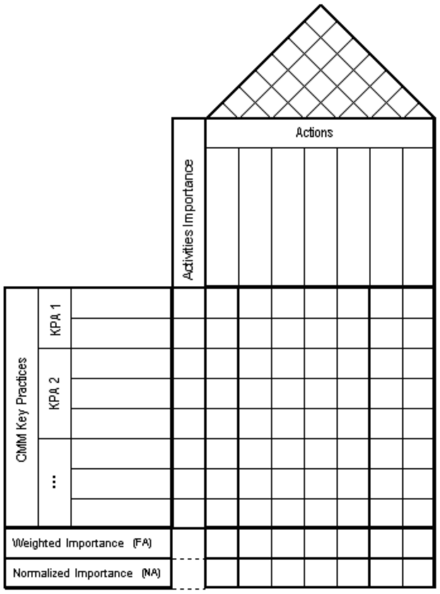

Figure 5. Action Plan House of Quality (AP-HoQ) Matrix

5. Simplified frameworks

One of the advantages of the proposed framework is that, when the schedule and budget become tight, this framework can be modified for a faster process improvement. We give the name "light-weight frameworks" to differentiate the variations which will be introduced in this section from the "comprehensive framework" introduced in the previous two sections. Although these light-weight frameworks help solve the problem of tight schedule and budget, they may not produce results as accurately as the original framework does. It is recommended that when schedule and budget allow, in order to gain accuracy in the final result, an organization makes as few modifications from the comprehensive framework as possible. However, accuracy may not always be the top concern to an organization. In this case, three Light-Weight Frameworks can be applied.

Light-weight framework 1 simplifies phase 1 of the comprehensive framework. Process improvement groups may choose to integrate requirements without using the RI matrix. Arbitrary weights can be assigned to requirements and these arbitrarily ranked requirements are delivered to phase 2. To further speed up phase 1, process improvement groups may even choose to collect requirements from only the most important group instead of from all related groups and no requirements integration is needed.

Light-weight framework 2 simplifies the mapping from requirements to CMM and from CMM to action plans in the comprehensive framework. An organization may choose to use either the KPA goals or KPs instead of both. For instance, requirements are mapped to KPA goals and these goals can be mapped to action plans directly. Phase 1 remains the same as that in the comprehensive framework.

An organization may even choose Light-weight framework 3 by combining the first two light-weight frameworks mentioned above.

6. Evaluation

In order to validate our framework, a case study based on the comprehensive framework was developed. A software development organization is considering improving its software process. Eighteen requirements are collected from three perspectives—business, management, and quality—within the company, with 6 requirements in each perspective. These requirements are integrated using RI Matrices as introduced in section 4.1. Figure 6 shows the integration of business and management perspectives. When the requirements from all three perspectives are integrated, we have the adjusted weights (AW) ready to be delivered to phase 2.The RG, GP and AP-HoQ matrices are then used to transform these requirements into a set of prioritized actions.

	Initial Global Priorities	Within Budget	On schedule	High customer satisfaction	Increase productivity	Manage project aggressive	High comformance to software engineering standard	Weighted Priorities (P1-2)
Initial Global Priorities		0.1077	0.0923	0.0615	0.0308	0.0308	0.0154	
Increase profit	0.1615	●	○	●	●	▽		0.3404
Lead in the competition	0.1154	○	●	●	▽			0.2006
Reduce cost of development	0.1154	●	○		▽	▽		0.1509
Reduce time to develop	0.0462	○	●	▽	○	▽		0.0618
Reduce marketing time	0.0462	○						0.0149
Improve quality	0.0231			●		○	●	0.0181
Weighted Priorities (P2-1)		0.3355	0.2109	0.1689	0.0561	0.0121	0.0032	

Figure 6. Business Requirements and Management Requirements

Based on the normalized weighted importance values in the AP-HoQ matrix, we can sort all action plans in a decreasing order. Since these action plans are derived based on CMM KPs, their execution will help the organization achieve a higher maturity level. At the same time, stakeholder requirements from multiple perspectives have also been reflected in the action plans. The action plans at the top of the list can better satisfy stakeholder requirements and should receive more resources than those listed after them.

From the result of this case study, we found the presented framework helpful in generating action plans for SPI, in prioritizing stakeholder requirements, and in relating stakeholder requirements with these action plans. While the prioritized action plans are the final deliverable of SPI, the stakeholder requirements serve as the root of our framework. This is reasonable because all SPIs have to satisfy requirements from certain stakeholders. By providing a methodology for requirements prioritization, we first identify requirements with more and stronger correlations with other requirements from multiple stakeholders. Satisfying these requirements will also satisfy other requirements to some extent. Therefore, they receive higher priority values in this framework. When action plans are connected to these requirements, the priority values of requirements are transformed into priority values of actions plans.

The benefit of prioritizing the final action plans lies in the fact that sometimes not all actions can be performed within schedule. When this is the case, by executing the action plans with higher priorities first, we can always achieve higher satisfaction level of requirements.

After small scale experiments, our framework will be further validated by the Toshiba Corporation in a large scale environment. We will report the results in the future.

7. Conclusion

Software Process Improvement has received attention from more and more software development organizations nowadays. It is a common practice for an organization, especially a large organization, to select a software engineering standard and model for its process improvement. CMM from SEI as a software process improvement framework has become very influential in software development world. However, CMM only provides practices which are needed to be performed without specifying how. In addition, they are not directly related to business goals and other requirements. Our project successfully addressed this issue by using QFD as a tool to connect requirements within an organization to the action plan for its process improvement through KPA goals and KPs in CMM. In this paper, we discussed in detail how to prioritize and integrate requirements, how to map requirements to CMM goals, how to prioritize CMM KPs, and how to generate and prioritize action plans. Three objectives are achieved through our framework: 1) business and other requirements of an organization are successfully connected with the KPA goals and KPs in CMM; 2) software process requirements from multiple perspectives are successfully prioritized based on their impact correlations with each other; and 3) process improvement actions are successfully prioritized according to organization requirements through KPAs and KPs.

CMMI has been developed to solve the problem of using multiple CMM models for different areas of application. While the basic ideas remain the same in CMM and CMMI, the primary difference is that, in CMMI, especially in the continuous model, process improvements are designated to individual Process Areas (PA) rather than the whole process. Our framework needs to be adjusted accordingly so that a list of action plans is generated for each PA. Considering the connections (similarities) between the two models, this framework adjustment should not be overwhelming. This framework adjustment can be part of our future work.

References

[1] Software engineering – Product quality – Part 1: Quality model, ISO/IEC 9126-1:2001.

[2] Paulk, Mark C., Bill Curtis, Mary Beth Chrissis, Charles V. Weber. “Capability Maturity Model for Software, Version 1.1.” Technical Report. CMU/SEI-93-TR-024, ESC-TR-93-177, February, 1993.

[3] Paulk, Mark C., Charles V. Weber, Suzanne M. Garcia, Mary Beth Chrissis, Marilyn Bush. “Key Practices of the Capability Maturity Model, Version 1.1.” Technical Report. CMU/SEI-93-TR-025, ESC-TR-93-178, February, 1993.

[4] Paulk, Mark C. “A Comparison of ISO 9001 and Capability Maturity Model for Software.” Technical Report. CMU/SEI-94-TR-12, ESC-TR-94-12, July, 1994.

[5] Akao, Yoji, ed., Quality Function Deployment: Integrating Customer Requirements into Product Design, Cambridge, MA, Productivity Press, 1990.

[6] Akao, Yoji, Glenn H. Mazur. “Using QFD to Assure QS9000 Compliance.” 4th International Symposium on Quality Function Deployment, Sydney, 1998.

[7] Zultner, Richard E. “Business Process Reengineering with Quality Function Deployment: Process Innovation for Software Development.” 7th Symposium on QFD (ISBN1-889477-07-9), 1995.

[8] Ita Richardson, Eamonn Murphy, KevinRyan, “Development of Generic Quality Function Deployment Matrix”, Quality Management Journal, Vol. 9, No. 2, April 2002, pp. 25-43.

[9] Andreas Hierholzer, Georg Herzwurm, Harald Schlang, “Applying QFD for Software Process Improvement at SAP AG.”, Proceedings of the World Innovation and Strategy Conference in Sydney, Australia, August 2-5, 1998, S. 85-95

[10] Akao Y., Hayazaki T. “Environmental Management System on ISO 14000 Combined with QFD.” Transactions of the Tenth Symposium on QFD. Novi, Michigan. ISBN 1-889477-10-9

[11] Ita Richardson. “Quality Function deployment – A Software Process Tool?”. Third Annual International QFD Symposium. Linkoping, Sweden, Oct. 1997.

[12] Xiaoqing Frank Liu, Praveen Inuganti, Chandra Sekhar Veera. “An Integration Methodology for Software Quality Function deployment.” Final Project Report to the Toshiba Corporation. 2003.

[13] Xiaoqing Frank Liu. “A Quantitative Approach for Assessing the Priorities of Software Quality Requirements.” The Journal of Systems and Software, 42, (1998), 105-113

[14] Zultner, R.E. “Quality Function Deployment (QFD) for Software.” American Programmer (1992).