

Business-oriented software process improvement based on CMMI using QFD

Yan Sun, Xiaoqing (Frank) Liu *

Computer Science Department, Missouri University of Science and Technology, Rolla, MO 65409, United States

ARTICLE INFO

Article history:

Received 17 November 2008
Received in revised form 24 July 2009
Accepted 7 August 2009
Available online 14 August 2009

Keywords:

Software Process Improvement
CMMI
Quality Function Deployment
Prioritization
Relationship matrix
Process requirements

ABSTRACT

Software Process Improvement (SPI) has become the key to the survival of many software development organizations. As a follow-up of a previous paper on SPI for the CMM using Quality Function Deployment (QFD), a new SPI framework integrating QFD with the CMMI is developed. Similar to the earlier SPI framework for the CMM, the proposed SPI framework based on the CMMI using QFD aims to achieve three objectives: (1) to map process requirements, including business requirements, to the CMMI with the help of QFD; (2) to develop a method based on QFD for the integration and prioritization of requirements from multiple perspectives (groups); and (3) to be able to prioritize software process improvement actions based on process requirements.

Process requirements from multiples groups of stakeholders (perspectives), including business goals, are integrated and prioritized. SPI actions are linked to these process requirements using QFD. Thus, the priorities of actions reflect the priorities of process requirements. By executing the actions with the highest priorities, the highest satisfaction level of process requirements can be achieved. One unique feature of the framework for the CMMI Continuous representation is that the priority values of these actions can be compared across PAs, even when the PAs attempt to reach different capability levels.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Many software companies rely 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 [1,20], ISO 15504 [2,19], the Capability Maturity Model (CMM) [3,4] and the Capability Maturity Model Integrated (CMMI) [5,15,23] from the Software Engineering Institute (SEI). However, during process improvement, standards and models should not be used independently from business and other requirements in an organization.

These models and standards share some common concerns in terms of quality and process improvement. However, their emphases are different. For instance, ISO standard addresses the minimum criteria for a quality system while the CMM and the CMMI emphasize continuous improvement. It is unfair to make a judgment on which one is better [1,6]. However, because of the level of details and breadth covered, the CMM and the CMMI have become very popular worldwide.

Like all the other standards and models on software process improvement, the CMM and the CMMI address the question of “what to do” or improvement approaches while leaving “how to do it” or specific best practices to organizations [21]. Therefore,

some methodology is needed to transform the CMM and the CMMI activities into a set of actions which are detailed enough to be followed by software engineers. Our previous study has proposed a CMM-based SPI framework, which uses QFD to map process requirements to a list of prioritized detailed actions through the CMM [7,8]. However, SEI has stopped its support to the CMM and recommended the CMMI instead. Thus, in this paper, the CMM-based SPI framework is extended to support the CMMI, especially the CMMI Continuous representation. Following the same general flow, business and other process requirements of an organization are mapped to the CMMI and activities, and action plans are developed to satisfy those requirements and the CMMI with the help of Quality Function Deployment (QFD).

As mentioned by Serrano, SPI is a relatively new area and many of the SPI ideas originated from the theories and methodologies in the manufacturing world [18]. QFD is one of the tools originally used by manufacturing and can be potentially helpful to SPI. Since 1966, Quality Function Deployment (QFD) has been used worldwide in many industry and sector to prioritize spoken and unspoken customer needs; to translate these needs into actions and designs such as technical characteristics and specifications; and to build and deliver a quality product or service by focusing on achieving a common goal of customer satisfaction.

In this framework, QFD is used to help an organization achieve three objectives. First, business and other requirements within an organization are mapped to the CMMI Process Areas (PAs) and Practices. A connection is established so that the organization

* Corresponding author. Tel.: +1 573 341 4848; fax: +1 573 341 4501.
E-mail address: fliu@mst.edu (X. (Frank) Liu).

can see clearly how the CMMI helps with its business. Second, business needs and software process requirements from various sources are integrated and prioritized. Third, QFD helps transform requirements of the organization into process actions through PAs. This directly results in the improvement of the organization process.

1.1. Related work

Several attempts have been made to integrate QFD into SPI. Ita Richardson proposed a four-stage model for software process improvement in small companies [9,10]. The measurements in this generic model are based on self assessment of the software process. Therefore, this model can be adopted by small businesses without extra efforts and investments. In larger companies, however, the organizational structures become more complex, which makes self assessment in this model more difficult. Thus, there is a need for methods developed specifically for larger organizations with more complex structures.

Zultner's Business Process Reengineering model with QFD divided the Business Process Reengineering into four major phases: analysis of the current development process, generation of new process concepts (alternatives), selection of the best new development process, and implementation of the selected new process [11–13]. Zultner's approach uses either the major competitor's performance or creative thoughts of employees, but not existing process standards that are widely used in the industry, as the source of process improvement. Although this approach may help address specific issues in an organization, it is difficult to apply it in different situations or environments to produce consistent process improvement results based on creative thoughts as suggested by the approach when compared with a method using a popular reference model.

SAP also uses QFD in SPI [14]. Although this approach considers the participation of multiple stakeholders, it treats requirements from all stakeholders as equally important and does not consider the relationships among multiple perspectives. In addition, the process improvement actions are directly related to process requirements. No standards, such as ISO or the CMM, are considered throughout the workflow. Therefore, although the action plan is prioritized, the priorities may be unreliable.

There have been a few studies using QFD together with CMMI [24,25]. They provide theoretical basis for the validity of this approach. But they either lack details on how to use QFD on CMMI or need more empirical case study for support.

In a previous study, we have introduced a SPI framework based on the CMM using QFD as a tool to connect process requirements and actions [7,8]. Considering the fact that SEI has stopped its support to the CMM and recommended the CMMI model [5,15] instead, a SPI framework based on the CMMI is needed. The CMMI was developed to solve the problem of using multiple CMM models for different areas of application [3,4]. The new integrated model (the CMMI-SE/SW) uses Process Areas (known as PAs), which is different from the previous model (CMM), and covers both system engineering and software engineering, rather than only software engineering in the SW-CMM.

1.2. Our research approach

In this paper, a SPI framework is developed to derive action plans based on software process requirements with the help of QFD and in accordance with the CMMI. It integrates the best features of the existing methodologies, such as using QFD to translate process requirements into the action plan and integrating the process requirements from multiple groups of stakeholders, and addresses the limitation of the previous studies, such as omitting

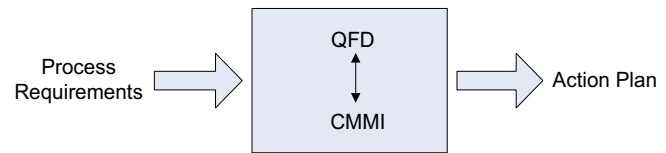


Fig. 1. High-level SPI framework.

the differences among different groups of stakeholders and lack of conformance to reference models.

The CMMI is chosen in this framework because of its popularity in the industry and proven effectiveness. SEI has reported the effectiveness of the CMMI by comparing data from 35 organizations. Tangible benefits such as cost, schedule, productivity, quality, customer satisfaction, and return on investment (ROI) were obvious [16].

With the CMMI used in the framework, the SPI can be sketched from a high level as shown in Fig. 1 below. The action plan is generated based on the process requirements through the CMMI using QFD. This guarantees that the actions are in accordance with the CMMI and, at the same time, satisfy the process requirements from organization.

2. Framework for software process improvement using QFD

In the framework, QFD is used to help with the SPI based on the CMMI. Three objectives can be achieved. First, business and other requirements within an organization are mapped to the CMMI Process Areas and Practices. There have been studies showing that it is important to link business goals to process improvement goals [22]. A connection is established so that the organization can clearly see how the CMMI helps with its business goals. Second, software process requirements from multiple perspectives are prioritized so that requirements with more and stronger impacts on other requirements can receive higher priority values. Third, QFD helps transform requirements of the organization into process actions through Process Areas (PAs) and Practices in the CMMI. Therefore, the ordering of the actions taken is based on how they are related to both the software process requirements and the corresponding Practices in the CMMI. For instance, an action (A1) derived using this approach is strongly related to Practice 1 in the CMMI, while another action (A2) is strongly related to Practice 2. Suppose that according to the mapping developed from this framework, it is found that Practices 1 reflects the requirements more than Practice 2 does. As a result, A1 should have priority over A2. This guarantees that the actions are in accordance with the CMMI and, at the same time, the execution order of these actions better satisfy the process requirements from the organization. This directly results in the improvement of the organizational process.

The framework is designed in such a way that the process requirements can be reflected through our framework all the way down to the action plans. The requirements from multiple perspectives are correlated with each other using the priority assessment technique in [17]. As a result, the priority value of each requirement is adjusted after the impacts from the other requirements are assessed.

One important point that needs to be noted is that the matrices used for the mappings among requirements, from requirements to PAs/Practices, as well as from Practices to actions can be very large. In order to ensure that the mappings and values assigned to the mappings make sense, all stakeholders (or representatives of stakeholders) need to work together and reach consensus. A house of quality represented by a matrix establishes a connection between two perspectives from “what to do” to “how to do”. Stakeholders must be identified from these two perspectives to

identify and assess mapping relationships in the matrix. For example, stakeholders from business requirements perspective are usually business executives and managers. Stakeholders from PAs/Practices are CMMI process improvement specialists. They work together to assess impact of PAs/Practices on business requirements. This is a classical issue in QFD applications and it has been successfully addressed in this manner in many QFD applications [8,9,13,26]. It needs to be pointed out that the whole mapping process can be time-consuming. This is the cost of overhead introduced to the whole process improvement. However, this will also help prioritize and identify key actions in process improvement and lead to better final result.

In order to incorporate both the Continuous representation and the Staged representation in the CMMI, the SPI framework based on the CMMI contains two portions: (1) SPI framework for the CMMI Continuous representation and (2) SPI framework for the CMMI Staged representation.

2.1. SPI framework for CMMI Continuous representation

The SPI framework for the CMMI Continuous representation differs a lot from the CMM framework [7,8]. However, the same techniques of correlation-based prioritization with the help of QFD are used in the framework. In the Continuous representation of the CMMI, the capability levels are assigned to individual PAs. Different PAs can be at different capability levels. Each PA has two types of goal: (1) Generic Goals and (2) Specific Goals. Generic Goals try to institutionalize the capability levels in the CMMI, with one Generic Goal for each level. Specific Goals describe the Practices that must be implemented to satisfy the process area. These goals are satisfied by including Generic Practices and Specific Practices. Fig. 2 illustrates how the Practices and the actions are prioritized in the SPI framework for the CMMI Continuous representation using QFD. The process requirements are used to in the prioritization of both PAs and Practices (as shown as Step 1 and Step 2 in Fig. 2). The first step is to calculate the priority values of PAs. Then the Practices are prioritized from both the process requirements and PAs. Therefore, there are two lines in Fig. 2 for Step 2. Depending on which PA a Practice is from, the priority value of that Practice, which is calculated from the requirements (Step 2 with solid line), is multiplied by the PA priority (Step 2 with dotted line). Finally, the action priority values are calculated from the Practice priority values (Step 3 in Fig. 2).

As shown in Fig. 2, the PAs are prioritized based on those process requirements (1). Thus, the PAs that help achieve higher overall satisfaction of process requirements get higher importance. When resources are limited such that it is impossible to improve all PAs, the most important ones can be selected for improvement.

In order to make improvements on the PAs, Generic Practices for Generic Goals and Specific Practices for Specific Goals at various capability levels are prioritized at the next phase (2). The priorities of Practices at different capability levels are determined by their correlations with the same set of process requirements. Because in the CMMI Continuous representation, different PAs can have different of capability levels, the prioritization of Practices should be done for individual PAs. Thus, in this framework for the CMMI

Continuous representation, the Practices in each level of individual PAs are prioritized separately. The Practices that aim to achieve higher overall satisfaction of key goals receive higher importance values. The priority values for each PA calculated in the previous phase (1) are used in the calculation of priorities of Practices. In the last phase (3), separate sets of action plans are derived from Practices in each of the PAs for different capability levels. The actions that help to support more important Practices receive higher priorities.

As a result, the process requirements are reflected in PAs, Practices, and the actions. The actions both follow the process capability standards in the CMMI and satisfy the process requirements. Those actions with higher importance values help to achieve higher process requirements satisfaction.

Fig. 3 shows a more detailed SPI framework based on the CMMI Continuous representation. Phase 1, which is based on the requirements prioritization method from multiple perspectives [17], is exactly the same with the SPI framework based on the CMM [8]. Various perspectives are represented as P_1 through P_n . Each perspective contains multiple requirements. The software process requirements in perspective 1 are represented as R_{1-1} , R_{1-2} , etc. These perspectives of software process requirements can then be prioritized based on their relative importance within the organization and integrated into one single set of requirements.

For instance, requirements from three (3) perspectives are to be prioritized and integrated. All requirements from Perspective 1 are mapped with all requirements from Perspective 2 using a matrix. The correlation between each requirements-pair is identified and used in the calculation of priority values. Similarly, the same mapping and calculation is done between all requirements from perspectives 1 and 3, perspectives 2 and 3. Each requirement from these three perspectives now has two priority values, obtained from the mapping to the requirements from the other two perspectives. Some normalization method can be applied and each requirement will receive a final priority value. These final priority values of the requirements are comparable across different perspectives. Thus, the requirements can be integrated into one single set and used in the rest of the SPI framework. Based on the priority values, some requirements are more important than others. For more details of the requirements prioritization across perspectives, please refer to our previous study [17].

In Fig. 3, these integrated requirements are represented as R_1 through R_m , where m is the total number of software process requirements from all perspectives. The prioritization ensures that requirements from different perspectives are comparable with each other, and the integration reflects the correlations among requirements from different perspectives. The deliverable of this phase is a set of prioritized and integrated software process requirements, which serves as the input to Phase 2.

The second through fourth phases of this framework are applied to the PAs in the CMMI Continuous representation. In the CMMI Continuous representation, different capability levels are applied to different PAs, therefore, the framework for the CMM model [8] cannot be applied. Instead of mapping the prioritized and integrated requirements from Phase 1 to all the goals in a particular maturity level, they are linked to each of the PAs in Phase 2 and,

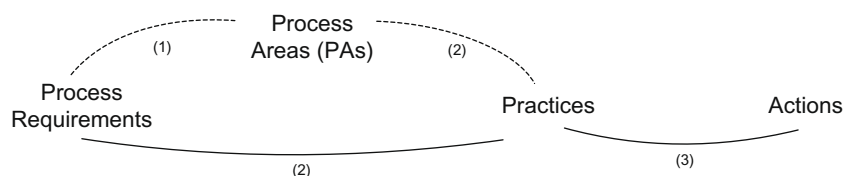


Fig. 2. Priority calculation in SPI framework based on CMMI Continuous representation using QFD.

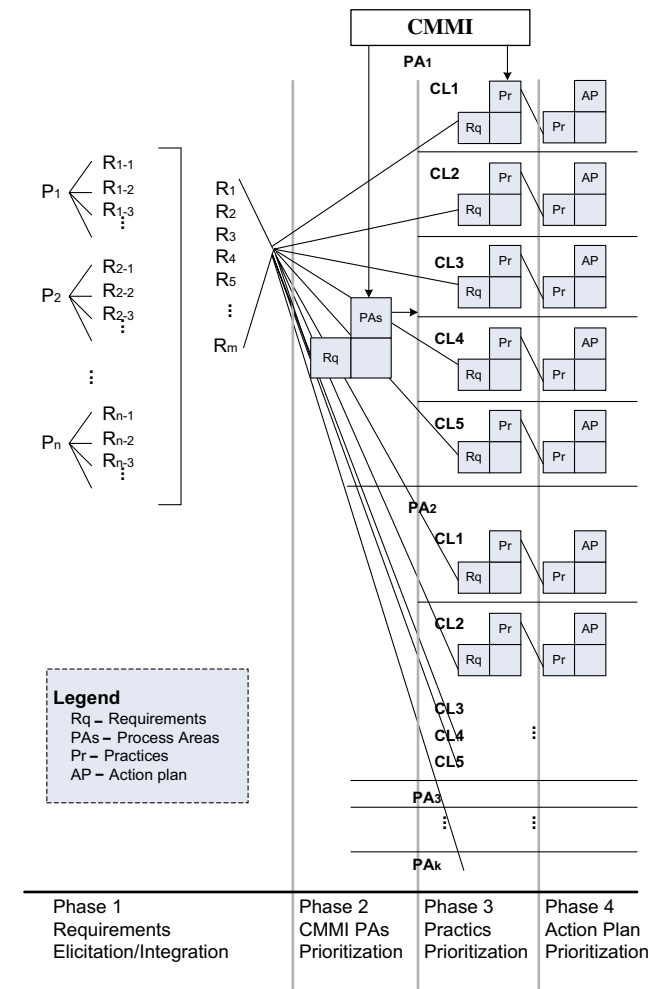


Fig. 3. Software Process Improvement through CMMI Continuous representation using QFD.

depending on the target capability level, linked to each of the Practices in that level in Phase 3 using relationship matrices. In addition to the correlation values between process requirements and Practices, the priority value for each PA also participates in the calculation of the prioritization of Practices in that PA for a particular capability level. Finally, the prioritized Practices are transformed into prioritized action plans using House of Quality (HoQ).

In Phase 2, which is “CMMI PA prioritization,” all PAs are selected and prioritized based on the requirement priorities derived from Phase 1. For instance, there are a total of m requirements from Phase 1 and a total of n PAs in the CMMI Continuous representation to be considered. These requirements and PAs are entered into a matrix and their correlations are identified. The priorities of the PAs are then calculated based on (1) their correlations with the requirements, and (2) the priority values of the requirements. Some PAs will be more important than others. If a PA_i is more closely related to the requirements with high priorities, it should be more important than other PAs and deserve more attention.

This phase helps achieve three important objectives. First, the organization needs to comply with the CMMI standard. At the same time, the organization needs to ensure that by improving process areas to higher capability levels, the process is also satisfying the business and other requirements within the organization. In Phase 2, relationship matrices are used to establish connections between the requirements from the organization and each of the

PAs. This matrix demonstrates that complying with the CMMI standard also helps satisfy the business and other requirements in the organization. Second, the PAs are prioritized so that only the most critical ones will be selected to improve if resources are limited. Third, the final set of action plans needs to be prioritized based on the priorities of requirements so that more important actions receive more resources. The PAs serve as the bridge between requirements and the action plan. By prioritizing the PAs, requirements from the organization can be transformed to the Practices in Phase 3, and finally to the action plans in Phase 4. In this way, a set of actions can be executed not only to reach higher capability levels in various PAs, but also to satisfy organizational process requirements.

Phase 3 of our framework, which is “practice prioritization,” involves the prioritization of Practices for a particular capability level within each PA. The prioritization is carried out on the basis of the deliverables from Phase 2. According to the CMMI specifications, all these Practices for a capability level within a PA have to be performed in order for that PA to reach that particular capability level. However, they do not necessarily require the same amount of resources. These Practices serve as a bridge between the requirements and the final actions, and it is necessary to know how these Practices reflect the software process requirements. In order to show the connections between the requirements and the final action plans, these Practices have to be prioritized based on their correlations with requirements as well as the priority values of the PAs they belong to, which are now also reflecting requirements priorities.

Similar to the previous phases, a matrix is used to map Practices from a PA to all the requirements obtained in Phase 1. Their correlations are identified and the priority values are calculated from the requirement priorities and the correlations. Because in this phase we are looking at the Practices from different PAs, we need to include the PA priorities into consideration. Thus, the priority value of a Practice needs to be adjusted by the priority value of the PA it comes from. For instance, all Practices from PA_i are mapped to all the requirements from Phase 1 using a matrix. The priority values of the Practices are calculated. If a $Practice_j$ has correlations with all the important requirements, it should have a higher priority value. This priority value is then adjusted by the priority value of PA_i , which is the highest among all PAs. The final value, which reflects both requirements importance and PA importance, will again be the highest among all Practices.

In Phase 4 of the framework, which is “action plan development and prioritization,” sets of actions are derived from the prioritized Practices for the desired capability levels of various PAs. These actions are mapped to Practices using a matrix. Their priority values can be calculated based on the priority values of the Practices as well as their correlations with these Practices. These actions should reflect the requirements integrated in Phase 1. Meanwhile, they also state what needs to be executed in order to reach a particular capability level of a particular PA. These actions guide the process improvement.

For instance, an action_k has correlations with many of the important Practices such as Practice_j. This action should also receive a high priority value. Thus, more resources should be assigned to actions with high priorities like action_k because these actions are indirectly correlated to the important PAs such as PA_i , and also indirectly correlated to the important requirements.

As shown in the above framework, by incorporating requirements from the organization into action plans through the goals and the Practices, the connection between the objectives of the organization and PA capability levels becomes clear. Furthermore, because the priority values of actions are calculated from the Practice priorities, which in turn reflects requirements importance and PA importance, the actions can be compared globally across PAs,

even when different PAs are aiming to reach different capability levels.

2.2. SPI framework for CMMI Staged representation

Because of the resemblance between the CMMI Staged representation and the CMM, the SPI framework for the CMMI Staged representation, as shown in Fig. 4, is almost identical to the SPI model based on the CMM [7,8].

For each of the four maturity levels, the set of requirements with adjusted priorities are related to the goals. The goals are prioritized based on those process requirements. Thus, the goals that achieve higher overall satisfaction of process requirements get higher importance.

The CMMI Staged representation has Generic Practices categorized into four Common Features as well as the Specific Practices which correspond to the “Activities Performed” Common Feature in the CMM. The priorities of Practices are determined by their correlations with goals. In this SPI framework, the Generic Practices in each Common Feature and the Specific Practices are prioritized separately based on the priorities of the goals. Practices that aim to achieve higher overall satisfaction of goals receive higher importance values. Separate sets of action plans are derived from the Generic Practices in each of the Common Features as well as from the Specific Practices. The actions that help to support more important Practices receive higher priorities.

As a result, the process requirements are reflected in PA goals, Practices, and the actions. The actions both follow the process maturity standards in the CMMI Staged representation and satisfy the process requirements. Those actions with higher importance values help to achieve higher process requirements satisfaction.

In Fig. 2, Phase 1 is exactly the same with the SPI framework based on the CMM [8]. Various perspectives are represented as P_1 through P_n . Each perspective contains multiple requirements.

The software process requirements in perspective 1 are represented as R_{1-1} , R_{1-2} , etc. These perspectives of software process requirements can then be prioritized based on their relative importance within the organization and integrated into one single set of requirements. A simple example has been given in Section 2.1.

In Phase 2, which is “CMMI goal prioritization,” the goals of all PAs in a particular maturity level are selected and prioritized based on the requirements from Phase 1. All goals are mapped to all requirements using a matrix. For instance, $Goal_i$ has correlation with all the important requirements. Based on the correlations, $Goal_i$ should receive a high priority value. By prioritizing the goals, requirements from the organization can be transformed to the Practices in Phase 3, and finally to the action plans in Phase 4. In this way, a set of actions can be executed not only to achieve a specific maturity level in the CMMI, but also to satisfy organizational process requirements.

Phase 3 of the framework, which is “practice prioritization,” involves the prioritization of Practices within all PAs of a specific level. The prioritization is carried out on the basis of the deliverables from Phase 2. According to the CMMI specifications, all these Practices have to be performed in order to reach that particular maturity level. These Practices serve as a bridge between the requirements and the final actions, and it is necessary to know how these Practices reflect the software process requirements. In order to show the connections between the requirements and the final action plans, these Practices have to be prioritized based on the goals, which are now reflecting requirements priorities. The mapping between the goals and Practices has been clearly shown in the CMMI documentation [15]. These mappings as well as the priority values of goals are used in the calculation of Practices. If a $Practice_j$ has strong correlations with important goals such as $Goal_i$, it should receive a high priority value, too.

In Phase 4 of the framework, which is “action plan development and prioritization,” a set of actions is derived from the prioritized

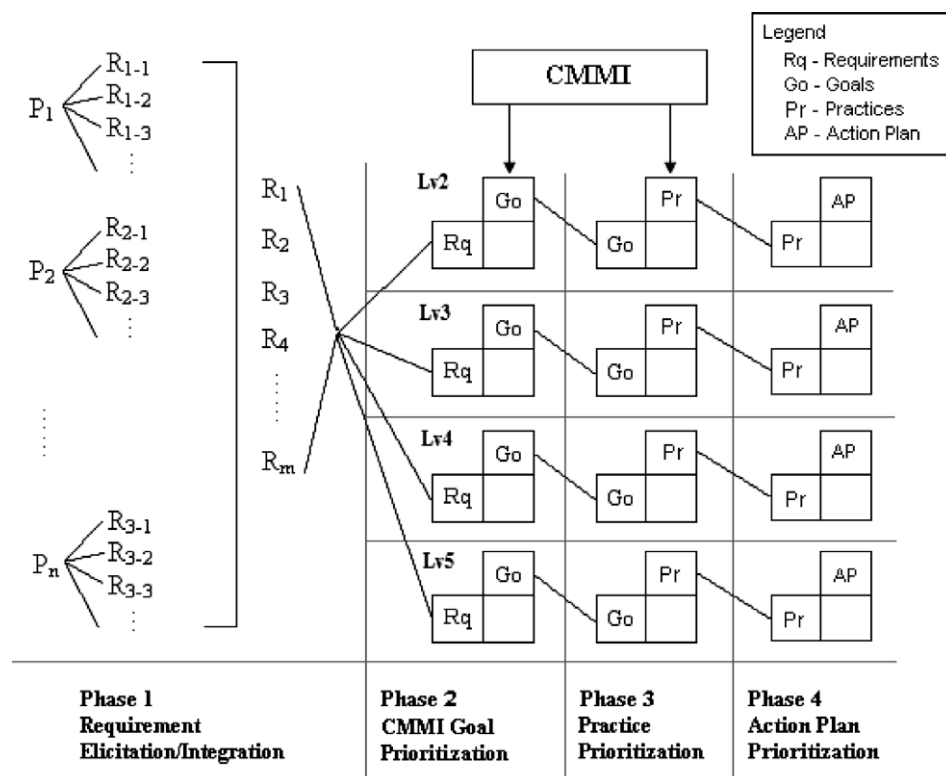


Fig. 4. Software Process Improvement through the CMMI Staged representation using QFD.

Practices. A matrix is used to correlate Practices with actions. If an action_k is strongly correlated to the important Practices such as Practice_j, it will receive a high priority value.

From the sequence of correlations, the final set of actions should reflect the requirements integrated in Phase 1. Action_k has a high priority value because it is indirectly correlated to the important goals such as Goal_i and the important requirements. Meanwhile, they also state what needs to be executed in order to reach a particular CMMI maturity level. These actions guide the process improvement. Thus, more resources should be assigned to those actions with high priorities.

By incorporating requirements from the organization into action plans through goals and Practices, the connection between the objectives of the organization and the CMMI maturity levels becomes clear.

Because of the similarity between this framework for the CMMI Staged representation and the SPI framework based on the CMM in the previous paper [8], the detailed matrices are not repeated. More emphasis is given to the CMMI Continuous representation in this paper.

3. Matrices in our framework based on CMMI Continuous representation

Due to the similarity between the CMMI Staged representation and the CMM, the matrices used in the SPI framework based on the CMMI Staged representation are identical to those used in the framework based on the CMM and thus skipped in this paper. In this section, four similar but different matrices used in the framework based on the CMMI Continuous representation are introduced.

Our working examples discussed hereafter are developed partially based on an application developed in a sponsored project with an industrial company presented in our previous paper [8]. Mapping relationships in new matrices developed in this paper are identified by authors through playing roles of stakeholders and assessing impact of PAs/Practices on requirements, as well as impact of actions on Practices based on our experience in assessing these kinds of relationships in the previous application developed with our industrial partner.

3.1. Requirements impact matrix (RI Matrix) in Phase 1

In the SPI framework for the CMMI, the requirements prioritization still uses the method as introduced in the paper of priority assessment from multiple perspectives [17]. Before the integration of requirements using RI Matrix, every perspective receives a perspective weight based on its relative importance to the organization; at the same time, each requirement can be assigned a local priority value within the perspective it belongs to. The local priorities and perspective weights are assigned.

After all requirements receive their normalized global priorities, RI Matrix is used to integrate and prioritize these requirements. Because at the requirements integration and prioritization phase, there is no difference between the SPI frameworks for the CMM and the CMMI, we can still use the examples in the CMM paper [8].

3.2. Requirements-process areas impact matrix (RPA Matrix) in Phase 2

In this phase, the Requirements-Process Areas Impact Matrix (RPA Matrix) is used to prioritize goals on the basis of the adjusted priorities (APs) of requirements that come from Phase 1. The correlations between the requirements and the PAs are reflected in the matrix, and a value indicating the relative importance for each PA is calculated.

Fig. 5 shows an example of the RPA Matrix using the CMMI Continuous representation. The following five steps guide through the process of building the Requirements-Goals Impact (RG) Matrix:

1. Enter the integrated requirements (deliverables from Phase 1) along with their adjusted priorities (noted as Requirements Importance in the table) into the rows.
2. Enter all PAs in the CMMI of a particular maturity (for Staged representation)/capability (for Continuous representation) level into the columns. Considering the fact that one matrix containing too many items is hard to read, one big matrix can be broken into multiple smaller matrices, each of which contains a group of PAs.
3. Determine the correlation between each requirement and each PA. The same set of symbols representing the weights of 9, 3, and 1¹ in the paper for SPI based on the CMM [8] is used in this matrix. It should be noted that we play roles of stakeholders and determine the appropriate values to enter in the matrix in this hypothetical application example. In practice, stakeholders or their representatives should work together to determine the correlation values.
4. Calculate the weighted importance values for PAs (FPA) using Eq. (1):

$$FPA_i = \sum_{j=1}^M AP_j^* IR(PA_i, R_j), \quad (1)$$

where: FPA_i is the weighted priority value of PA_i.

AP_j is the adjusted priority value of requirement j calculated in Phase 1. IR is the impact correlation value between a requirement-PA pair.

5. Normalize the weighted importance values.

In order to illustrate the above steps using an example, the eighteen requirements with adjusted priorities from Phase 1 were chosen and entered into the rows of Fig. 5.

Eight (8) PAs are used in this example. After determining and entering the correlations between the requirements and PAs into the matrix, the weighted importance of each PA can be calculated using Eq. (1). For instance, in Fig. 5, the PA “Integrated Teaming” has correlations with three requirements. Thus, its weighted importance value is calculated as follows:

$$\begin{aligned} FPA_1 &= \sum_{j=1}^{18} AP_j^* IR(PA_1, R_j^*) = 0.0559 * 9 + 0.0418 * 3 + 0.0233 * 3 \\ &= 0.6984. \end{aligned}$$

After all three weighted importance values are calculated, these values are normalized to obtain the normalized weighted importance. The goals used in this matrix together with their normalized importance values (NPA) serve as the input to Phase 3.

3.3. Requirements-Practices Impact Matrix (RPr Matrix) in Phase 3

In Phase 3 of the framework, the Requirements-Practices Impact Matrix (RPr Matrix) decides the importance of Practices in a particular capability level of a PA based on their relationships with the prioritized requirements as well as the priority value of the PA they belong to. The Practices from the target capability level of a PA

¹ QFD uses the weights of 9, 3, and 1 to represent strong, medium, and weak correlations. This framework follows the same way to determine correlations in all matrices introduced in this study. In the matrices, these weight values are represented as symbols. Please refer to the legend of Fig. 5 for the mapping between symbols and numerical weight values.

Requirements - PAs									
	Requirements Importance	Project Planning	Project Monitoring and Control	Supplier Agreement Management	Integrated Project Management for IPPD	Risk Management	Integrated Teaming	Integrated Supplier Management	Quantitative Project Management
R1: Increase profits	0.3517			○					○
R2: Lead in competition	0.2401	▽							○
R3: Reduce cost of development	0.1845			▽		●			○
R4: Reduce time to develop	0.0740	○	●	▽	▽	▽		▽	○
R5: Reduce marketing time	0.0556	▽							
R6: Improve quality	0.0401		○	▽	▽	●		▽	●
R7: Within Budget	0.2673	●	●	▽					○
R8: On schedule	0.1935	●	●						●
R9: High customer satisfaction	0.1493	▽	●			●			●
R10: Increase productivity	0.0559	●					●		▽
R11: Manage project aggressively	0.0418	○	○	○	○	○	○	○	●
R12: High conformance to software engineering standard	0.0233	○					○		○
R13: Low failure rate	0.0978		○	▽		●		▽	○
R14: Low defect rate	0.0664		○	▽		●		▽	○
R15: High reliability	0.0499		○	▽		●		▽	○
R16: High requirement satisfaction	0.0585	●	○						●
R17: High maintainability	0.0373	○						▽	●
R18: High usability	0.0133	●							
Weighted Importance (FPA)		6.2707	7.2204	1.9605	0.2395	5.4914	0.6884	0.4909	6.3333
Normalized Importance (NPA)		1.7621	2.0289	0.5509	0.0673	1.5431	0.1962	0.1379	1.7796
		6.8824							6.8824

Fig. 5. Requirements-Process Areas Impact (RPA) Matrix for CMMI Continuous representation.

are put into the same matrix with the prioritized requirements derived in Phase 1.

Because the goals reflect the process requirements, by relating Practices with the goals, the priorities of Practices should also reflect the requirements priorities. An example of the RPr Matrix based on the CMMI Continuous representation is shown in Fig. 6.

The following six steps are followed in order to develop an RPr Matrix:

1. Enter the integrated requirements (deliverables from Phase 2) along with their adjusted priorities into the rows.
2. List the Practices in the CMMI into the columns of the matrix.
3. Determine the correlation between each requirement and each PA. The same set of symbols representing the weights of 9, 3,

and 1 is used in this matrix. Again, stakeholders or their representatives should determine the correlation values in practice.

4. Calculate the weighted importance value for Practices using Eq. (2):

$$FPr_i = \sum_{j=1}^N AP_j^* IR(Pr_i, R_j), \quad (2)$$

where: FPr_i is the weighted priority value of Practice i , AP_j is the adjusted priority value of requirement j calculated in Phase 1.

IR is the impact correlation value between a requirement-Practice pair.

5. Normalize all weighted importance values of Practices to obtain normalized priority values (NPr).

Standard 9-3-1																						
Strong ◆ 9.0																						
Moderate ◇ 3.0																						
Weak ▼ 1.0																						
		Requirements Importance	SP1.1 Monitor Project Planning Parameters	SP1.2 Monitor Commitments	SP1.3 Monitor Project Risks	SP1.4 Monitor Data Management	SP1.5 Monitor Stakeholder Involvement	SP1.6 Conduct Progress Reviews	SP1.7 Conduct Milestone Reviews	SP2.1 Analyze Issues	SP2.2 Take Corrective Action	SP2.3 Manage Corrective Action	GP2.1 Establish an Organizational Policy	GP2.2 Plan the Process	GP2.3 Provide Resources	GP2.4 Assign Responsibility	GP2.5 Train People	GP2.6 Manage Configurations	GP2.7 Identify and Involve Relevant Stakeholders	GP2.8 Monitor and Control the Process	GP2.9 Objectively Evaluate Adherence	GP2.10 Review Status with Higher Level Management
R1: Increase profits	0.3517				◇			◇	◇													
R2: Lead in competition	0.2401				◇			◇	◇	◆	◆	◇			◇							
R3: Reduce cost of development	0.1845	▼			◇			◆	◆					◇							◆	
R4: Reduce time to develop	0.0740	▼	◇				◇	◆	◆	◇	◇	▼		◆	◆	▼	◇			◆	◆	
R5: Reduce marketing time	0.0556	◇												◇	◆	◆	◆	▼		◇	▼	▼
R6: Improve quality	0.0401				◆	◇		◆	◆	◆	◆	◆		▼	◇		◇	▼			◆	◆
R7: Within Budget	0.2673	▼			◇			◆	◆												◆	
R8: On schedule	0.1935	▼	◇				◇	◆	◆	◇	◇	▼		◆	◆	▼	◇			◆	◆	
R9: High customer satisfaction	0.1493				◇	▼	◆	◆	◆	◆	◆	◆					▼			◆		◆
R10: Increase productivity	0.0559				◇	◆								◇	◆	◆	◆	▼		◇		
R11: Manage project aggressively	0.0418	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇
R12: High conformance to software engineering standard	0.0233	◇	▼		◆		▼	◆	◆		◇	◇	◆	◆	▼	◇	◇	◆	◆	◆	◆	◆
R13: Low failure rate	0.0978				◆	◇		◆	◆	◆	◆	◆			▼	▼	◇	▼		◇		
R14: Low defect rate	0.0664				◆	◇		◆	◆	◆	◆	◆			▼	▼	◇	▼		◇		
R15: High reliability	0.0499				◆	◇		◆	◆	◆	◆	◆			▼	▼	◇	▼		◇	◇	
R16: High requirement satisfaction	0.0585				◇	▼	◆	◆	◆	◆	◆	◆									◆	
R17: High maintainability	0.0373												◆		◇	◇	◇	◇		◇		
R18: High usability	0.0133							▼	▼								◆					
Weighted Importance (FP)			1.0814	0.9512	6.5448	1.5989	2.8214	12.7555	12.7555	7.3167	7.3167	5.3411	0.6708	3.6707	4.7263	1.7923	2.7000	0.7571	4.2531	8.1472	2.7159	0.3907
Normalized Importance (NP)			0.0122	0.0108	0.0741	0.0181	0.0319	0.1444	0.1444	0.0829	0.0829	0.0605	0.0076	0.0416	0.0535	0.0203	0.0306	0.0086	0.0482	0.0923	0.0308	0.0044
Global Importance (GP)			0.0248	0.0219	0.1504	0.0367	0.0648	0.2931	0.2931	0.1681	0.1681	0.1227	0.0154	0.0843	0.1086	0.0412	0.0620	0.0174	0.0977	0.1872	0.0624	0.0090

Fig. 6. Requirements-Practices Impact (RPr) Matrix for PA "Project Monitoring and Control" in CMMI Continuous representation.

6. Multiply the normalized priority value of the PA with each of the NPr values to obtain the global importance value (GPr) of each Practice.

In Fig. 6, the same eighteen requirements from Fig. 5 are entered in the rows. The corresponding capability level 2 Practices from the second PA in Fig. 5, which is "project monitoring and control", are entered in the columns. The correlations are determined and entered in the matrix. Based on these correlations, the weighted importance value for each of the Practices can be calculated using Eq. (2). For example, the Specific Practice (SP1.5), "Monitor Stakeholder Involvement" in Fig. 8 has strong correlations with process requirements R9 and R16, medium correlations

with R4, R8, and R11, and a weak correlation with R12. Thus, the weighted importance of 1.5 can be calculated using Eq. (2) as following:

$$\begin{aligned}
 FPr_{1.5} &= \sum_{j=1}^{18} AP_j^* IR(Pr_{1.5}, R_j) \\
 &= 0.0740 * 3 + 0.1935 * 3 + 0.1493 * 9 + 0.0418 * 3 \\
 &\quad + 0.0233 * 1 + 0.0585 * 9 \\
 &= 2.8214.
 \end{aligned}$$

After all the weighted importance values are calculated, the weighted priority (FPr) values are normalized to obtain the nor-

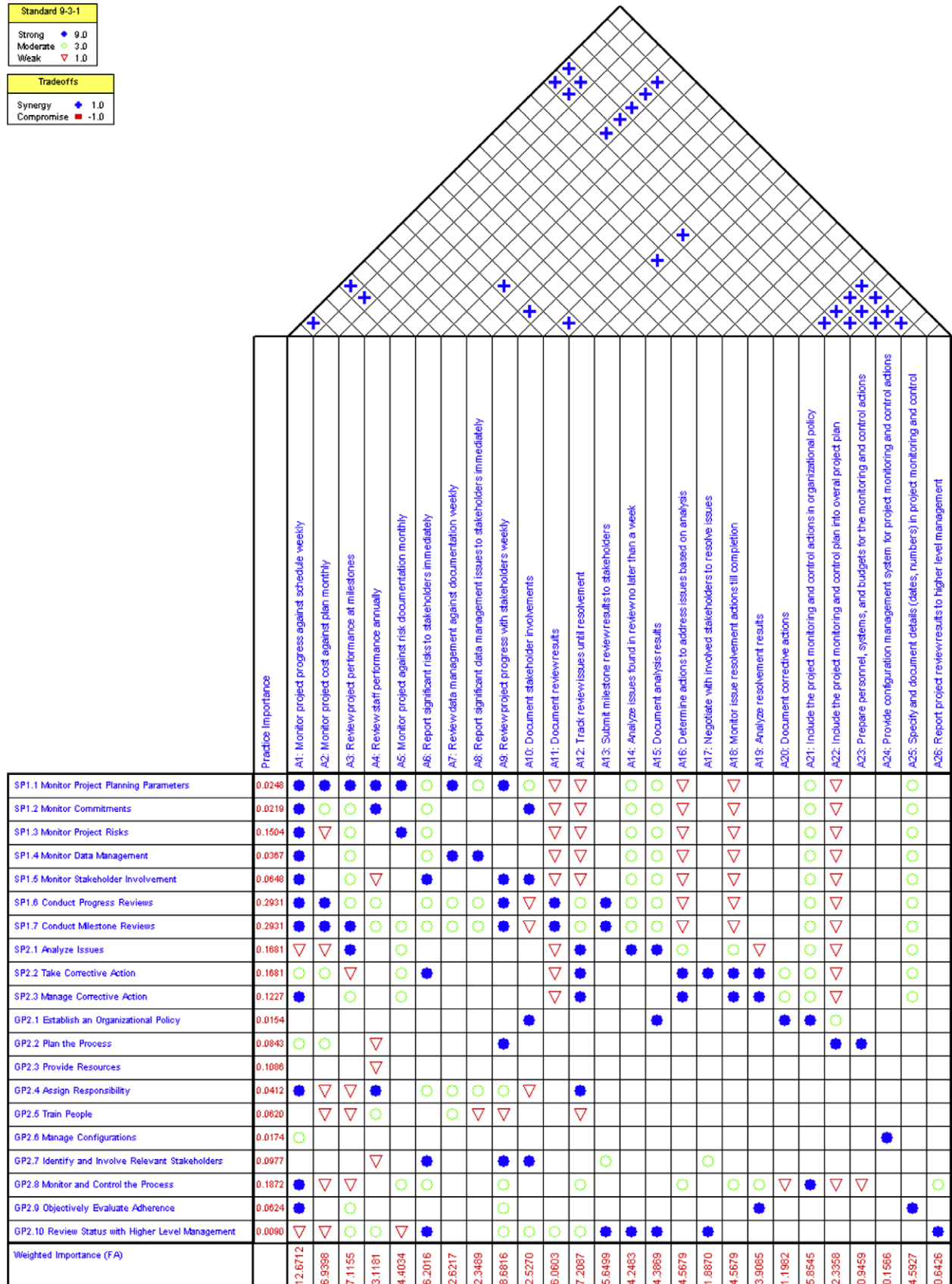


Fig. 7. Action Plan House of Quality Matrix (AP-HoQ Matrix) for PA "Project Monitoring and Control" in CMMI Continuous representation.

malized importance (NPr) and these NPr's are multiplied by 2.0289, which is the Normalized Importance value for the PA "Pro-

ject Monitoring and Control," to obtain the global importance values (GPr).

3.4. Action plan house of quality matrix (AP-HoQ) in Phase 4

Action plans are developed on the basis of Practices, and their correlations are determined using an AP-HoQ matrix. Actions are steps to be followed in software development. Actions tell what steps should be taken in order to achieve the goals. In an AP-HoQ matrix, Practices and actions are related to each other, and the degrees of correlations are calculated. From these correlations, priorities of actions are determined. The deliverable of this matrix tells which actions should be given more and better resources for the

fulfillment of goals and the institutionalization of the improved process. At the same time, these actions with higher priorities help achieve higher satisfaction of the process requirements. The impacts of actions on each other are also determined and represented in the roof of the house of quality. Although the roof is not involved in the calculation of priorities, it can help the process improvement team to decide which set of actions should be executed when choices are to be made. Obviously the actions that contribute more positively to the other actions should be selected over those conflicting with the others.

Standard 9-3-1																				
Strong ◆ 9.0																				
Moderate ○ 3.0																				
Weak ▽ 1.0																				
		Requirements Importance	SP1.1 Determine risks sources and categories		SP1.2 Define risk parameters		SP1.3 Establish a risk management strategy		SP2.1 Identify risks		SP2.2 Evaluate, categorize, and prioritize risks		SP3.1 Develop risk mitigation plans		SP3.2 Implement risk mitigation plans		GP3.1 Establish a defined process		GP3.2 Collected improvement information	
R1: Increase profits	0.3517																			
R2: Lead in competition	0.2401																			
R3: Reduce cost of development	0.1845																	◆	◆	
R4: Reduce time to develop	0.0740																	▽	▽	
R5: Reduce marketing time	0.0556																			
R6: Improve quality	0.0401	◆		○	○							○	○	○	○					
R7: Within Budget	0.2673																			
R8: On schedule	0.1935																			
R9: High customer satisfaction	0.1493																	◆	◆	
R10: Increase productivity	0.0559																			
R11: Manage project aggressively	0.0418	▽	▽	▽	▽				◆	○	○	◆	◆					◆	◆	
R12: High conformance to software engineering standard	0.0233																			
R13: Low failure rate	0.0978	◆	▽	○	○				◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	
R14: Low defect rate	0.0664	◆	▽	○	○				◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	
R15: High reliability	0.0499	◆	▽	○	○				◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	
R16: High requirement satisfaction	0.0595																			
R17: High maintainability	0.0373																			
R18: High usability	0.0133																			
Weighted Importance (FPr)			2.3296	0.1066	0.0266	0.0368	0.0844	0.0841	2.3031	2.1726	2.1726	5.5016	5.5016							
Normalized Importance (NPr)				0.1066	0.0172	0.0368	0.0313	0.0313	0.1054	0.0995	0.0995	0.2518	0.2518							
Global Importance (GPr)				0.1646	0.0266	0.0568	0.0483	0.0483	0.1627	0.1535	0.1535	0.3886	0.3886							

An example of the AP-HoQ matrix based on the CMMI Continuous representation is shown in Fig. 7. The following five steps are followed in order to develop an AP-HoQ matrix:

1. Enter the Practices together with their global importance values (GPr) into the rows of the AP-HoQ matrix.
2. Derive a set of actions from the Practices, and enter them into the columns. These actions represent the steps to be followed to execute the Practices. Various sets of actions are derived separately for different Common Features.
3. Determine the correlation between each Practice and each action. The same set of symbols representing 9, 3, and 1 is used in this matrix.
4. Calculate the weighted importance of actions using Eq. (3):

$$FA_i = \sum_{j=1}^Z GPr_j IR(A_i, Pr_j), \quad (3)$$

where: FA_i is the weighted importance value of Action i ,
 GPr_j is the global importance value of Practice j ,
 and IR is the impact correlation between an action-Practice pair.

5. Determine the correlations between pairs of actions in the roof. If the deployment of one action helps another, then these two actions are said to be positively correlated. In case the deployment of an action is detrimental to another, then a negative correlation is said to exist. A plus sign (+) is used to indicate the existence of a positive correlation between a pair of actions,

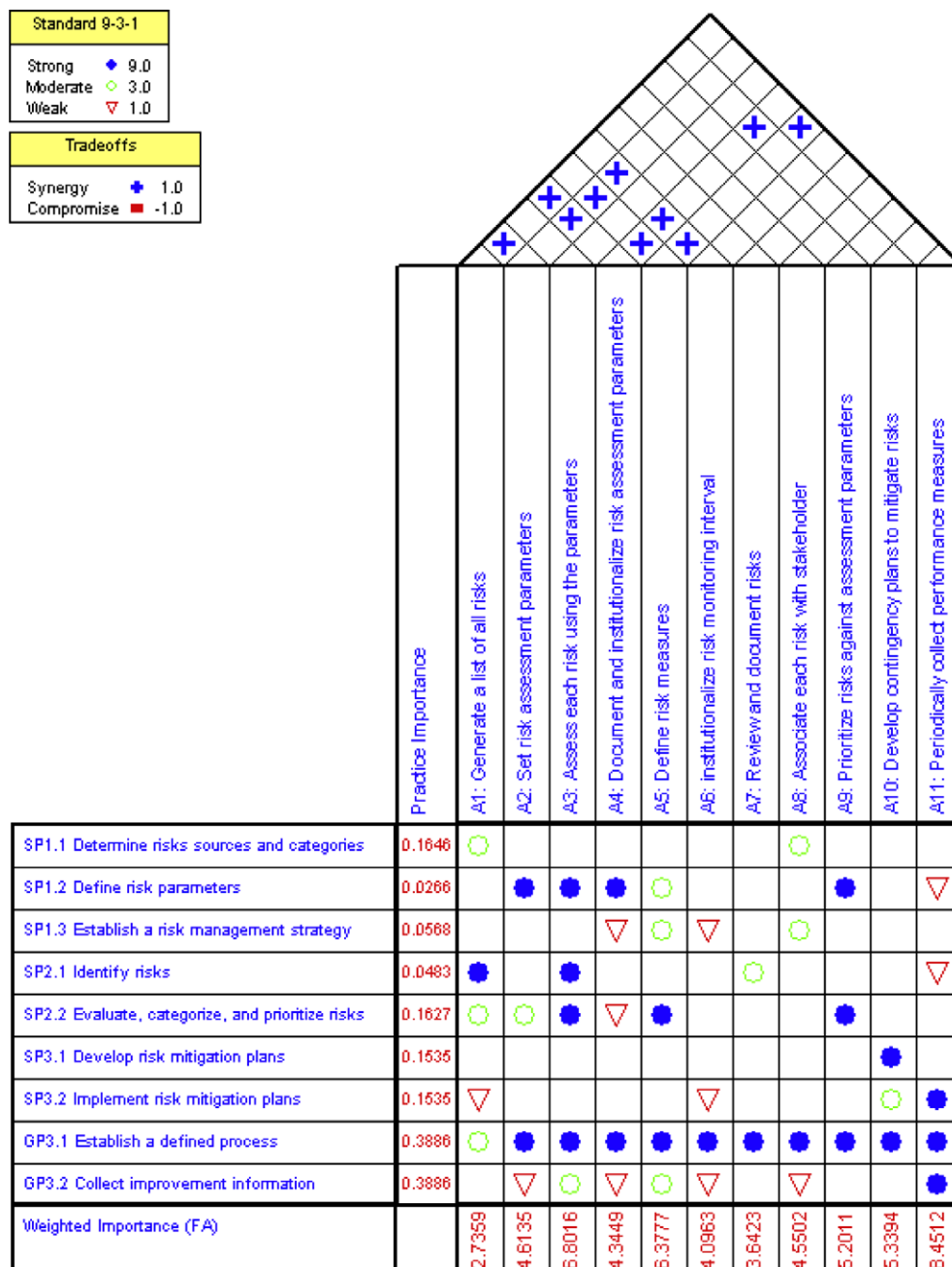


Fig. 9. Action Plan House of Quality Matrix (AP-HoQ Matrix) for PA "Risk Management" in CMMI Continuous representation.

while a negative sign (–) is used to indicate a negative correlation.

The same set of Practices from capability level 2 in PA “Project Monitoring and Control” in Fig. 6 are entered in the rows of Fig. 7, and a number of actions derived from these Practices are entered in the columns. The correlation between each Practice-action pair is determined and entered in the matrix. Based on these correlations and the normalized importance values of Practices, the weighted importance can be calculated for each action using Eq. (3). For instance, the action A8, “Report significant data management issues to stakeholders immediately,” has a strong correlation with Specific Practice SP1.4, medium correlations with SP1.1, SP1.6, SP1.7, and Generic Practice GP2.4, and a weak correlation with GP2.5. The weighted importance of A8 can be calculated using Eq. (3) as following:

$$\begin{aligned} FA_8 &= \sum_{j=1}^{20} NKP_j^* IR(A_8, KP_j) \\ &= 0.0417 * 3 + 0.0617 * 9 + 0.4924 * 3 + 0.4924 * 3 + 0.0692 \\ &\quad * 3 + 0.1042 * 1 \\ &= 3.9466. \end{aligned}$$

After all the weighted importance values are calculated, the action plans can be sorted based on their weighted importance. More important actions should receive more resources and attention because they help achieve higher levels of process requirements satisfaction.

4. Application example

In this section, the SPI framework based on the CMMI Continuous representation using QFD is illustrated. This framework starts from the same set of 18 process requirements from the paper on SPI for the CMM [8]. RI Matrices are skipped in the paper because they are exactly the same as the ones in the original paper.

Figs. 5–7 have shown the mapping from 18 process requirements to nine (9) Process Areas and 20 capability level 2 Practices, and from the Practices to final actions. The weighted importance values and the normalized importance values at each phase are calculated based on the correlations. The final set of actions help reaching capability level 2 in the PA of “Project Monitoring and Control” and at the same time help satisfying the business requirements.

In the CMMI Continuous representation, various PAs can have different capability levels. In this section, the SPI of another PA at capability level 3 is used, together with the matrices in Section 3, as an example to illustrate the framework. Fig. 5 has illustrated the PA prioritization using RPA Matrix and Fig. 6 shows the Practices prioritization in the PA of “Project Monitoring and Control,” aiming at capability level 2. The Normalized Practice priorities are multiplied by the PA priority value to obtain global importance values. These prioritized Practices are then mapped to actions in Fig. 7.

The same calculations are applied to another PA, “Risk Management,” which is aiming to reach capability level 3. After the same PA prioritization steps in Fig. 5, the capability level 3 Practices in “Risk Management” PA are prioritized in the RPr Matrix as shown in Fig. 8.

The 18 process requirements are entered in the rows of the matrix while the Practices in the PA are entered in the columns. The correlation between each Requirement-Practice pair using the 9–3–1 values is entered in the appropriate place in the matrix. For each Practice in the matrix, based on the requirement importance values and the correlation values, the weighted importance value

is calculated using Eq. (2). For instance, in Fig. 8, the Weighted Importance value for the Practice SP1.1 is calculated as:

$$\begin{aligned} FPr_{1.1} &= \sum_{j=1}^{18} AP_j^* IR(Pr_{1.1}, R_j) \\ &= 0.0401 * 9 + 0.0418 * 1 + 0.0978 * 9 \\ &\quad + 0.0664 * 9 + 0.0499 * 9 \\ &= 2.3296. \end{aligned}$$

After all Weighted Importance values in the matrix are calculated, they are normalized and then multiplied by the Normalized Importance value of “Risk Management” PA, which is 1.5431, to obtain the final Global Importance values. The Global Importance value of SP1.4 in this example is 0.1646.

Following the prioritization of the Practices in “Risk Management” PA, actions are developed from the Practices and they are prioritized using the AP-HoQ as shown in Fig. 9. The Practices in “Risk Management” PA and their Global Importance (GPr) values from Fig. 8 are entered in the rows of Fig. 9. The derived actions entered in the columns. The correlation between each Practice-action pair is determined based on the 9–3–1 values. Based on the impor-

Table 1

Prioritized actions from “Project Monitoring and Control” and “Risk Management”.

Actions	Weighted importance
A1: Monitor project progress against schedule weekly	12.6712
A9: Review project progress with stakeholders weekly	8.6816
A11: Periodically collect performance measures	8.4512
A12: Track review issues until resolved	7.2087
A3: Review project performance at milestones	7.1155
A2: Monitor project cost against plan monthly	6.9398
A3: Assess each risk using the parameters	6.8016
A5: Define risk measures	6.3777
A6: Report significant risks to stakeholders immediately	6.2016
A11: Document review results	6.0603
A21: Include the project monitoring and control actions in organizational policy	5.8545
A13: Submit milestone review results to stakeholders	5.6499
A10: Develop contingency plans to mitigate risks	5.3394
A9: Prioritize risks against assessment parameters	5.2011
A2: Set risk assessment parameters	4.6135
A25: Specify and document details (dates, numbers) in project monitoring and control	4.5927
A16: Determine actions to address issues based on analysis	4.5679
A18: Monitor issue resolution actions till completion	4.5679
A8: Associate each risk with stakeholder	4.5502
A5: Monitor project against risk documentation monthly	4.4034
A15: Document analysis results	4.3869
A4: Document and institutionalize risk assessment parameters	4.3449
A14: Analyze issues found in review no later than a week	4.2483
A6: institutionalize risk monitoring interval	4.0963
A19: Analyze resolved results	3.9085
A7: Review and document risks	3.6423
A4: Review staff performance annually	3.1181
A1: Generate a list of all risks	2.7359
A7: Review data management against documentation weekly	2.6217
A10: Document stakeholder involvements	2.527
A8: Report significant data management issues to stakeholders immediately	2.3489
A22: Include the project monitoring and control plan into overall project plan	2.3358
A17: Negotiate with involved stakeholders to resolve issues	1.887
A20: Document corrective actions	1.1982
A23: Prepare personnel, systems, and budgets for the monitoring and control actions	0.9459
A26: Report project review results to higher level management	0.6426
A24: Provide configuration management system for project monitoring and control actions	0.1566

tance values of Practices and the correlations, the Weighted Importance values of all actions are calculated using Eq. (3).

For instance, the Weighted Importance of A1 is calculated as:

$$\begin{aligned} FA_1 &= \sum_{j=1}^6 NKP_j^* IR(A_1, KP_j) \\ &= 0.1646 * 3 + 0.0483 * 9 + 0.1627 * 3 \\ &\quad + 0.1535 * 1 + 0.3886 * 3 \\ &= 2.7359. \end{aligned}$$

After all Weighted Importance values are calculated, the actions can be prioritized based on the importance values. Those actions with higher importance values deserve more attention and resources in SPI and thus the whole SPI project can reach a higher level of process requirements satisfaction.

In this application example, the prioritization of Practices and actions in two PAs are introduced. These two PAs try to reach two different capability levels—"Project Monitoring and Control" PA aims at Level 2 and "Risk Management" PA aims at Level 3. Even though they are at different capability levels, the actions can be prioritized across PAs. Table 1 below shows the ranked list of actions from these two PAs in Sections 3 and 4.

5. Conclusion

Some common limitations of existing SPI models and standards are the specification of "what to do" but not "how to do it." This paper extends our previous work based on the CMM [8] to the CMMI, using QFD as a tool to connect requirements within an organization to the action plans for its process improvement. The new framework discusses in detail how to prioritize and integrate requirements, how to map requirements to various components in the CMMI, and how to prioritize action plans.

The proposed framework has three major benefits: (1) mapping process requirements, including business requirements, to the CMMI with the help of QFD; (2) a method, based on QFD, for the integration and prioritization of requirements from multiple perspectives (groups); and (3) prioritization of software process improvement actions based on process requirements.

Using this framework, the process requirements are linked to the CMMI PAs and Practices. Through these links, software development organizations can see the direct benefit to the business by reaching a higher level in the CMMI.

In addition, while the prioritized action plans are the final deliverable of the SPI, the stakeholder requirements serve as the root of the prioritization. This is reasonable because all SPIs ought to satisfy the requirements from certain stakeholders.

When action plans are related to these stakeholder requirements, the priority values of requirements are transformed into priority values of action plans. As shown in the application examples, the actions are related with process requirements through the CMMI. By simply executing the action plans with higher priorities before others, one can always achieve a higher satisfaction level of requirements in an optimized way. Because the capability levels in different PAs are relatively independent in the calculation of action priorities, these action priorities can be compared across PAs, no matter these PAs are aiming at the same or different capability levels. This is exactly the advantage of the CMMI Continuous representation and the application example shows that our SPI framework supports this feature and it works as expected.

References

- [1] M.C. Paulk, A Comparison of ISO 9001 and Capability Maturity Model for Software, Technical Report, CMU/SEI-94-TR-12, ESC-TR-94-12, July, 1994.
- [2] ISO/IEC TR 15504: Information Technology – Software Process Assessment, 1998 (parts 1–9; part 5 was published in 1999), <<http://www.iese.fhg.de/SPICE>>.
- [3] M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber, Capability Maturity Model for Software, Version 1.1. Technical Report, CMU/SEI-93-TR-024, ESC-TR-93-177, February, 1993.
- [4] M.C. Paulk, C.V. Weber, S.M. Garcia, M.B. Chrissis, M. Bush, Key Practices of the Capability Maturity Model, Version 1.1. Technical Report, CMU/SEI-93-TR-025, ESC-TR-93-178, February, 1993.
- [5] M.B. Chrissis, M. Konrad, S. Shrum, CMMI: Guidelines for Process Integration and Product Improvement, second ed., Addison-Wesley Professional, 2006.
- [6] K.E. Emam, D.R. Goldenson, An Empirical Review of Software Process Assessments, NRC/ER-1065, November 1999, <http://it-iti.nrc-cnrc.gc.ca/publications/nrc-43610_e.html>.
- [7] X. Liu, Y. Sun, Y. Kyoya, K. Noguchi, QFD application in software process management and improvement based on CMM, in: Proceedings of the Third Workshop on Software Quality, St. Louis, Missouri, 2005.
- [8] X. Liu, Y. Sun, G. Kane, Y. Kyoya, K. Noguchi, Business-oriented software process improvement based on CMM using QFD, Software Process: Improvement and Practice 11 (6) (2006) 573–589.
- [9] I. Richardson, E. Murphy, K. Ryan, Development of generic quality function deployment matrix, Quality Management Journal 9 (2) (2002) 25–43.
- [10] I. Richardson, Quality function deployment – a software process tool? in: Third Annual International QFD Symposium, Linköping, Sweden, October 1997.
- [11] R.E. Zultner, Business process reengineering with quality function deployment: process innovation for software development, in: 7th Symposium on QFD (ISBN1-889477-07-9), 1995.
- [12] R.E. Zultner, Quality Function Deployment (QFD) for Software. American Programmer, 1992.
- [13] R.E. Zultner, Quality function deployment (QFD) for software: structured requirements exploration, in: G.G. Schulmeyer, J.I. McManus (Eds.), Total Quality Management for Software, Van Nostrand Reinhold, New York, 1992.
- [14] A. Hierholzer, G. Herzwurm, H. Schlang, Applying QFD for software process improvement at SAP AG, in: Proceedings of the World Innovation and Strategy Conference, Sydney, Australia, August 2–5, 1998, pp. 85–95.
- [15] CMMI Product Team, CMMI for Development, Version 1.2 (CMMI-DEV, V1.2), Technical Report, CMU/SEI-2006-TR-008, ESC-TR-2006-008, 2006.
- [16] D.L. Gibson, D.R. Goldenson, K. Kost, Performance Results of CMMI-Based Process Improvement, Technical Report, CMU/SEI-2006-tr-004, August 2006.
- [17] X. Liu, Y. Sun, C. Veera, Y. Kyoya, K. Noguchi, Priority assessment of software process requirements from multiple perspectives, Journal of Systems and Software 79 (11) (2006) 1649–1660.
- [18] M. Serrano, State of the art and future of research in software process improvement, in: Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04), Hong Kong, September 28–30, 2004.
- [19] Y. Choi, E. Lee, S. Ha, The management of software processes with software process improvement tools based on ISO 15504, in: The 7th International Conference on Advanced Communication Technology, vol. 2, Issue , 0-0 0, 2005 (ICACT 2005), pp. 933–936.
- [20] D. Stelzer, W. Mellis, G. Hertzwurm, Software Process Improvement via ISO 9000? Results of Two Surveys among European Software Houses, Software Process – Improvement and Practice, vol. 2, 1996, pp. 197–210.
- [21] D. Card, Research directions in software process improvement, in: Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04), Hong Kong, September 28–30, 2004.
- [22] B. Shen, T. Ruan, A case study of software process improvement in a Chinese small company, International Conference on Computer Science and Software Engineering 2 (12–14 December) (2008) 609–612.
- [23] B.C. Mary, K. Mike, S. Sandy, The Capability Maturity Model Integration: Guideline for Process Integration and Product Improving, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2002.
- [24] A. Becker, R. Prikladnicki, J. Audy, Strategic alignment of software process improvement programs using QFD, in: Proceedings of the First International Workshop on Business Impact of Process Improvements, May 13, 2008, Leipzig, Germany.
- [25] Z. Hu, Value-centric process improvement for small organizations by using QFD and CMMI, in: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, 2005.
- [26] Y. Akao, Quality Function Deployment: Integrating Customer Requirements into Product Design, Productivity Press, Cambridge, MA, 1990.