Spring 2008

Business-oriented Software Process Improvement based on CMM and CMMI using QFD

Yan Sun

Department: Computer Science

BUSINESS-ORIENTED SOFTWARE PROCESS IMPROVEMENT BASED ON CMM AND CMMI USING QFD

by

YAN SUN

A DISSERTATION

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2008

Approved by:

Xiaoqing (Frank) Liu, Advisor
Bruce M. McMillin
Fikret Ercal
Jennifer Leopold
Sahra Sedigh-Ali

ABSTRACT

Software Process Improvement (SPI) has become the key to the survival of many software development organizations. Many international SPI models/standards are developed for SPI. The Capability Maturity Model (CMM) and Capability Maturity Model Integrated (CMMI) from the Software Engineering Institute are two SPI models. In this study, several existing SPI models and approaches are reviewed, their advantages are identified, and their drawbacks are discussed. A set of new SPI frameworks integrating Quality Function Deployment (QFD) with both CMM and CMMI are developed by combining the best features of previous approaches and addressing their limitations.

The proposed SPI frameworks based on CMM or CMMI using QFD aim to achieve three objectives: 1) to map process requirements, including business requirements, to CMM or CMMI, with the help of QFD; 2) to develop a method based on QFD for the integration and prioritization of requirements from multiple perspectives; and 3) to be able to prioritize SPI actions based on process requirements.

By mapping the process requirements with CMM/CMMI, QFD displays the benefits of satisfying requirements through process improvement. In addition, process requirements from multiple groups of stakeholders (perspectives), including the 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.

ACKNOWLEDGMENTS

This dissertation owes a debt of gratitude to Dr. Xiaoqing (Frank) Liu, chair of my dissertation committee. It was under his continuous guidance and advice that this dissertation was completed. During the four years of Ph.D. studies at the University of Missouri-Rolla, I have learnt so much from him. It was with his help that I achieved what I have today. I am also grateful for Toshiba Corporation, which has generously funded my research projects over the years. These research projects set the foundation of this research. Without their funding, this dissertation could have never been finished.

Also, special thanks to Dr. Bruce M. McMillin, Dr. Fikret Ercal, Dr. Jennifer Leopold, and Dr. Sahra Sedigh Sarvestani. As members of my dissertation committee, they have offered precious time as well as advice that were extremely helpful to the completion of this dissertation.

It is hard to list all those who have worked with me and helped me, but I would like to express my gratitude to everyone in the Computer Science Department at the University of Missouri-Rolla. They have provided various types of help to me in my study and research. It has been a pleasure working with all of them.

Finally, I dedicate this dissertation to my dearest family members—my parents, my sister, and my wife. Without their love, support, and encouragement, life could never be so enjoyable.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

In this era of rapid technological innovation and changes, the key to the survival of a software company is the continuous improvement of its process. When talking about Software Process Improvement (SPI), many software development organizations think about existing models and standards, such as the ISO 9000 series of standards [1], ISO 15504 [2], the Capability Maturity Model (CMM) [3][4] and the Capability Maturity Model Integrated (CMMI) [5] 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, the ISO standard addresses the minimum criteria for a quality system while CMM and CMMI emphasize continuous improvement. It is unfair to make a judgment on which one is better [1][6]. However, considering the more detailed guidance and greater breadth provided by CMM, it may be a better choice for some software development organizations [1][7].

Like all the other standards and models on software process improvement, CMM and CMMI address the question of “what to do” while leaving “how to do it” to organizations. Therefore, some methodology is needed to transform CMM activities or CMMI Practices into a set of actions that are detailed enough to be followed by software engineers.

In this study, frameworks were developed to help map business and other process requirements of an organization to CMM and CMMI elements, and help develop action plans to satisfy those requirements using Quality Function Deployment (QFD).

Since 1966, Quality Function Deployment (QFD) has been used world-wide in nearly every 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.

There are three original contributions in the proposed framework, all with the help of QFD. First, business and other requirements within an organization are mapped to goals and activities in CMM, or Process Areas and Practices in CMMI. A connection is established so that the organization can see clearly how CMM/CMMI helps with its business. Second, business needs and software process requirements from various sources are integrated and prioritized. Third, QFD is used to help transform requirements of the organization into process actions through CMM/CMMI. It will be shown that this directly results in the improvement of the organization process.

2. RELATED CONCEPTS

2.1. SOFTWARE PROCESS IMPROVEMENT AND SOFTWARE PROCESS ASSESSMENT

Starting from the mid 1980s, great attention has been given to the study of software processes. The goal is to analyze the process structures and to find the best way to improve them. Two related terms should be clarified: Software Process Improvement (SPI) and Software Process Assessment (SPA).

Figure 2.1 shows the relationships among the process, process assessment, process improvement, and capability determination. As shown in the figure, process assessment is the starting point of process improvement. The result of process improvement is a changed (and hopefully better) process that can be assessed for further improvement. This relationship between SPI and SPA is concurred upon by the SPI paradigm from the SEI. The SPI paradigm from SEI is illustrated by its IDEAL model [8][9], as shown in Figure 2.2.

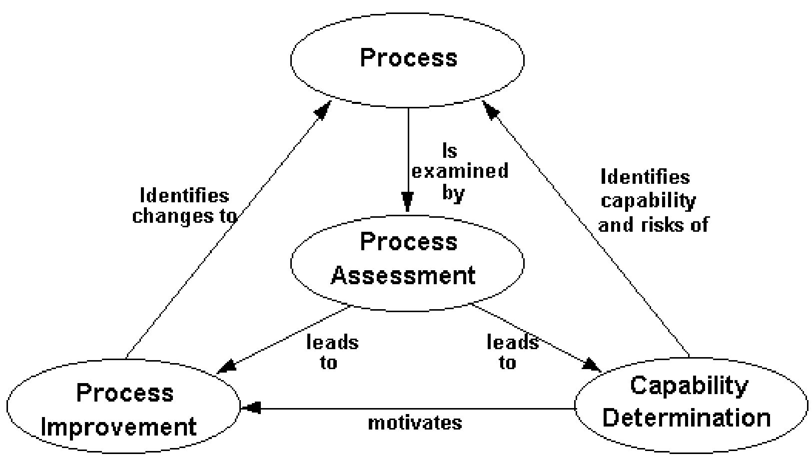


Figure 2.1. Software Process Improvement Context from SPICE [2]

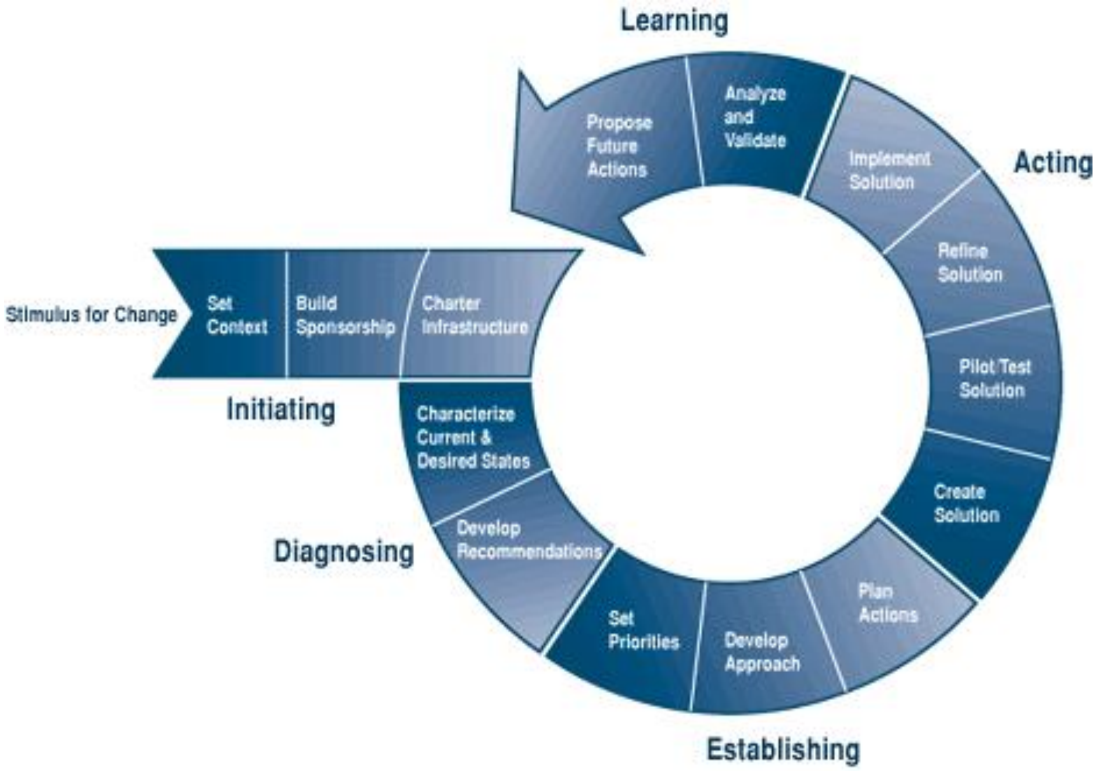


Figure 2.2. IDEAL Model from SEI [8]

The IDEAL model is an iterative approach for software process improvement, comprised of Diagnosing, Establishing, Acting, and Learning stages, which can be repeated as necessary. Of the five stages in the IDEAL model, the assessment of the current process is conducted in the Diagnosing stage.

2.2. QUALITY MODELS

2.2.1. Capability Maturity Model (CMM). The Capability Maturity Model for Software (CMM or SW-CMM) is a model developed by the SEI for judging the maturity of the organizational level software development process and for identifying key practices required to improve the maturity level [3][4][8][10][11].

There are five maturity level defined in CMM, which are shown in Figure 2.3.

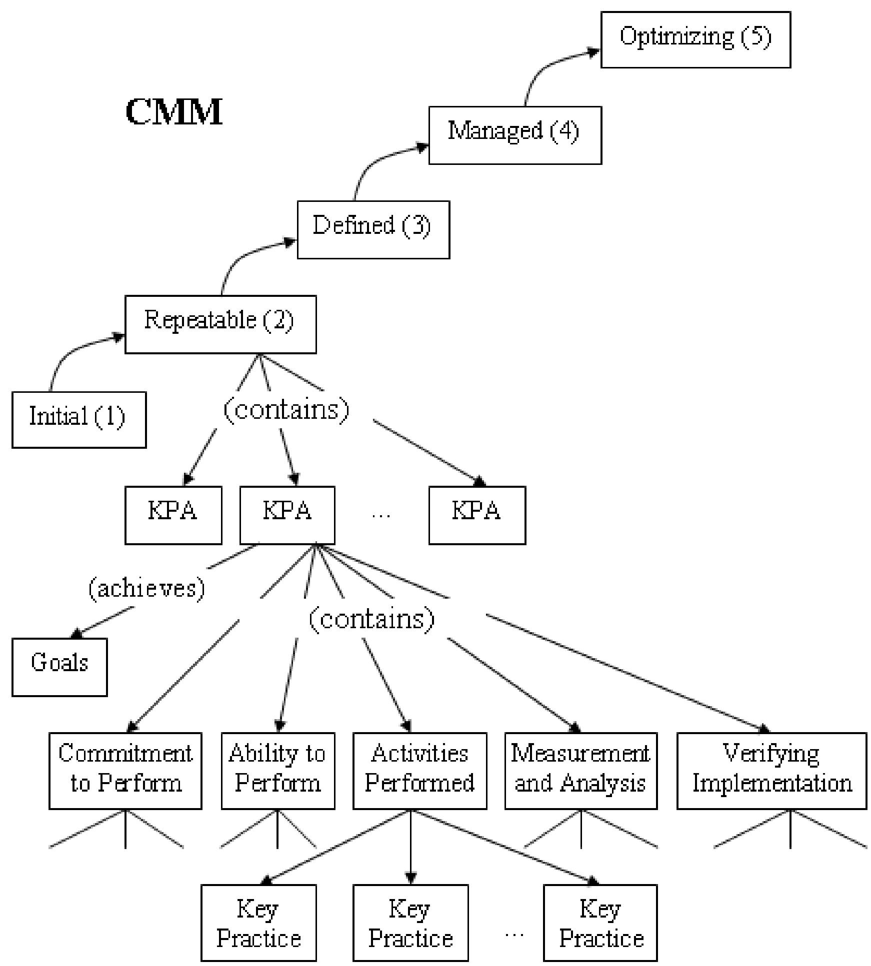


Figure 2.3. Capability Maturity Levels in CMM

Initial is the first level in the CMM model, in which processes are characterized as ad hoc. Only a few processes are defined and the success of a product depends on "heroes."

Repeatable is the second level in the CMM model in which basic management processes are established to track project cost, schedule, etc. This basic set of processes makes it possible to repeat previous project successes with similar applications.

Defined is the third level in the CMM model, in which a software process for management and engineering activities is defined, documented, and institutionalized.

This approved process is tailored for each software development project for quality control.

Managed is the fourth level in the CMM model, in which both product quality planning and software process are quantitatively measured and managed based on collected data.

Optimizing is the fifth and the highest level in the CMM model. In this level, the entire organization is aiming at continuous process improvement by means of identifying weaknesses and improving the process performance.

Each level in CMM other than Level 1 (Initial) contains multiple Key Process Areas (KPAs), each of which in turn contains multiple Key Practices that aim to achieve a set of goals. The goals for each KPA are used to determine whether the KPAs have been implemented within the organization. Key practices in each KPA describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the KPA.

Key practices in each KPA are grouped into the following five common features based on the direction they are targeting:

- Commitment to Perform
- Ability to Perform
- Activities Performed
- Measurement and Analysis
- Verifying Implementation

Key practices in the *Commitment to Perform* common feature describe the actions that must be taken to ensure the establishment and endurance of the process. The *Ability to Perform* common feature includes preconditions that must exist in order to implement the software process competently. The *Activities Performed* common feature describes the roles and procedures necessary to implement a KPA. Key practices in the *Measurement and Analysis* common feature describe the need to measure the process and analyze the measurement. The *Verifying Implementation* common feature deals with the steps required to ensure that activities are performed in accordance with the established process.

In order to reach a level in CMM, all of the KPAs in that level must be satisfied. To satisfy a KPA, all goals have to be achieved, which in turn are completed by performing all key practices in that KPA.

CMM is a widely used standard for many software development organizations all over the world, especially in North America and India. Some other standards and models are also available for software process improvement, such as the ISO 9000 series of standards. It is inappropriate to make a judgment on which is the best model/standard for software process improvement. Each of the models/standards has its own niche and special features. However, CMM has been considered as providing more detailed guidance as well as greater breadth for the adopters as compared to other international SPI standards. As a result, it has become extremely influential in the software industry.

The CMM model identifies what needs to be accomplished for software process improvement by listing the activities under each KPA. However, these activities are not tailored to any particular organization, which means that they sometimes are not specified in enough details to be carried out. In other words, CMM specifies “what to do” but not “how to do it” and why. The implementation is left to individual organizations.

In addition to that, CMM is serving as a standard that many businesses are striving to meet. These organizations try to reach high levels in the CMM model in order to be qualified for contracting bidding. It is less obvious to the business how high levels in CMM model help the business meet requirements from various branches of the organization such as higher management, and software development team. Therefore, in terms of software process improvement, the company needs a methodology to validate CMM, as well as to convert various requirements within the company into action plans, which at the same time helps the organization reach a higher level in CMM model.

2.2.2. Capability Maturity Model Integration (CMMI). CMMI model [5][12][13] was developed to solve the problem of using multiple CMM models for different areas of application [3][4]. The new integrated model (CMMI-SE/SW) uses Process Areas (known as PAs), which defined differently from the previous model (CMM), and covers both system engineering and software engineering, rather than only software engineering in the SW-CMM [5].

There are two representations of CMMI, Continuous [13] and Staged [12]. It allows choosing the order of improvements according to company goals by mitigating the risk. Instead of the five maturity levels in CMM, the CMMI continuous model has six different capability levels with names that are self-explanatory:

0. *Incomplete*
1. *Performed*
2. *Managed*
3. *Defined*
4. *Quantitatively Managed*
5. *Optimizing*

Another major distinction between the CMMI continuous model and CMM is that the continuous representation groups process areas by affinity categories and designates capability levels for process improvement within each process area. Furthermore, each process area has generic and specific goals and generic and specific practices, as shown in Figure 2.4.

The staged representation continues with the CMM structure by introducing five levels of process maturity with names very similar to the maturity levels defined by CMM:

1. *Initial*
2. *Managed*
3. *Defined*
4. *Quantitatively Managed*
5. *Optimizing*

Each of the five maturity levels is for the whole process. Unlike KPAs in the CMM model, in staged CMMI, each level has a different PA and each PA has specific and generic goals, as shown in Figure 2.5. It also specifies practices to achieve specific goals, and generic practices to achieve generic goals. The specific practices are analogous to “Activities Performed” in CMM. Key practices to achieve generic goals are categorized into four common features:

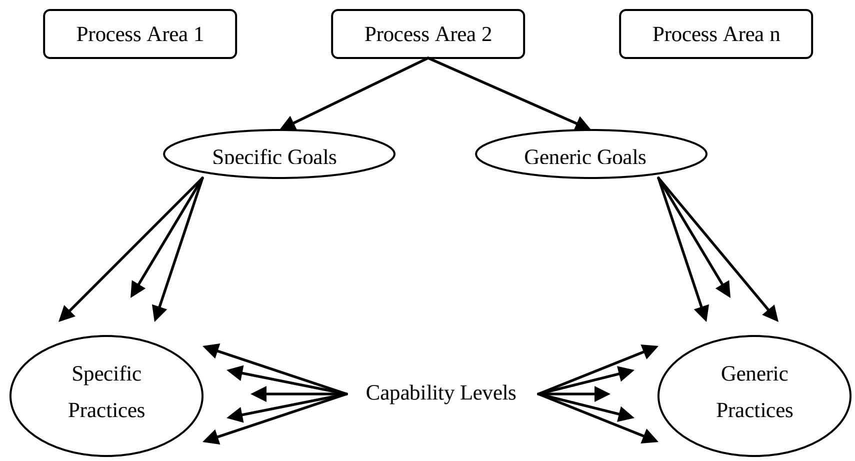


Figure 2.4. CMMI (Continuous) Model Components

- Commitment to Perform
- Ability to Perform
- Directing Implementation
- Verifying Implementation

2.2.3. SPICE (ISO/IEC TR 15504:1998 - Software Process Assessment).

SPICE is an international project on the development of an international standard for Software Process Assessment. The working draft of the standard was completed in 1995 and was published as ISO/IEC 15504:1998 – Software Process Assessment [2].

Similar to the continuous model in CMMI, ISO/IEC 15504 also specifies a "continuous" architecture. In other words, the capabilities defined in this international standard are applied to individual process areas rather than the complete process. A set of practices forms the lowest level of the architecture. The architecture organizes the practices into a number of categories using two different approaches:

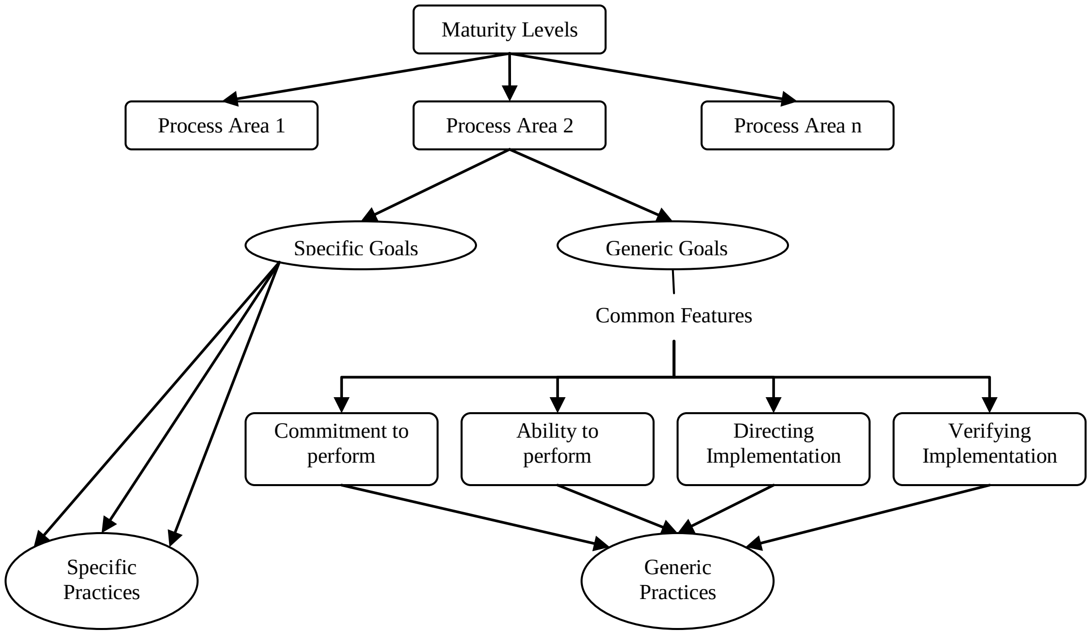


Figure 2.5. CMMI (Staged) Model Components

the essential activities of a specific process (best practices), which are grouped by the type of activity they address into processes and process categories;

generic practices, which are applicable to any process, and represent the activities necessary to manage a process and improve its capability to perform.

The best practices in the standard are organized into the following five process areas:

The **Customer-Supplier** process category consists of processes that directly impact the customer, support development and transition of the software to the customer, and provide for its correct operation and use.

The **Engineering** process category consists of processes that directly specify, implement, or maintain a system and software product and its user documentation.

The **Project** process category consists of processes which establish the project, and co-ordinate and manage its resources to produce a product or provide a service which satisfies the customer.

The **Support** process category consists of processes that enable and support the performance of the other processes on a project.

The **Organization** process category consists of processes that establish the business goals of the organization and develop process, product, and resource assets which will help the organization achieve its business goals.

Similar to CMMI, 5 levels of capability are defined:

Level 0; *Not-Performed:* There is general failure to perform the base practices in the process. There are no easily identifiable work products or outputs of the process.

Level 1; *Performed-Informally:* Base practices of the process are generally performed. The performance of these base practices may not be rigorously planned and tracked. Performance depends on individual knowledge and effort. Work products of the process testify to the performance. Individuals within the organization recognize that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process.

Level 2; *Planned-and-Tracked:* Performance of the base practices in the process is planned and tracked. Performance according to specified procedures is verified. Work products conform to specified standards and requirements.

The primary distinction from the Performed-Informally Level is that the performance of the process is planned and managed and progressing towards a well-defined process

Level 3; *Well-Defined:* Base practices are performed according to a well-defined process using approved, tailored versions of standard, documented processes.

Level 4; *Quantitatively-Controlled:* Detailed measures of performance are collected and analyzed. This leads to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known.

Level 5; *Continuously-Improving:* Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organization. Continuous process improvement against these goals is enabled by quantitative feedback from performing the defined processes and from piloting innovative ideas and technologies.

With the process areas and capability levels defined, the architecture of the international standard can be illustrated by Figure 2.6. The processes are listed vertically. Each of the processes can be assessed to determine the capability level reached.

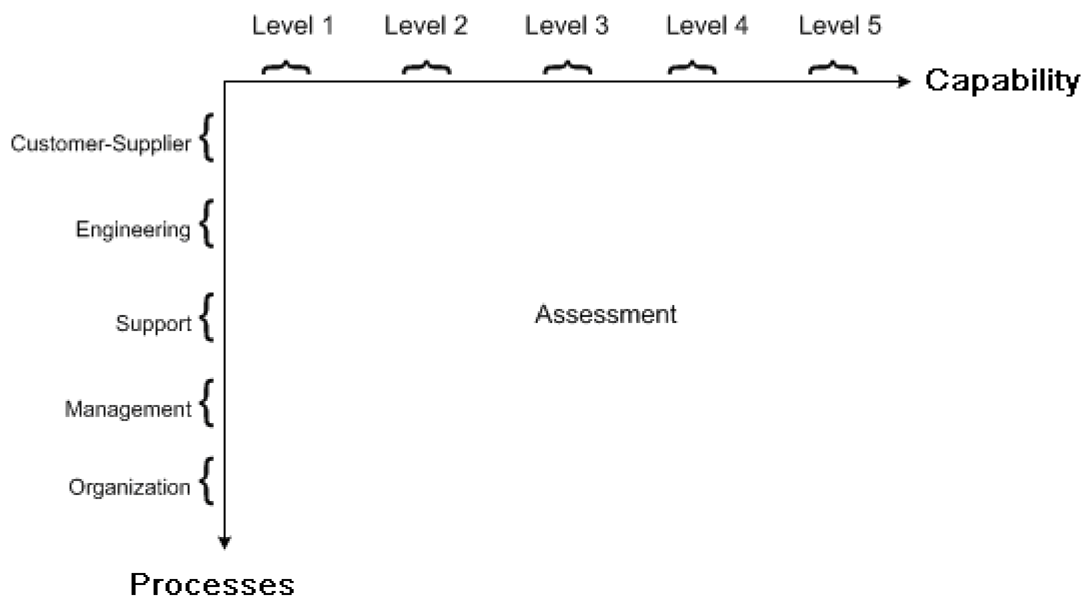


Figure 2.6. The Architecture of ISO/IEC 15504

2.2.4. ISO 9000 Series of Standards. The ISO 9000 series of standards is a set of documents dealing with quality systems that can be used for external quality assurance purposes [1]. Of the ISO 9000 series, ISO 9001, “Quality Systems-Model for quality assurance in design/development, production, installation, and servicing,” is the standard that is pertinent to software development and maintenance. The standard was designed to follow a process management approach, which requires that the processes be managed to satisfy a number of requirements. These requirements for the Quality Management System are outlined in sections 4 through 8 of ISO 9001, which are listed below.

Section 4: General Requirements

Requirements for the overall Quality Management System.

Section 5: Management Responsibility

Requirements for management and their role in the Quality Management System.

Section 6: Resource Management

Requirements for resources, including personnel, training, the facility, and work environment.

Section 7: Product Realization

Requirements for the production of the product or service, including planning, customer related processes, design, purchasing, and process control.

Section 8: Measurement, Analysis and Improvement

Requirements on monitoring processes and improving those processes.

Through these requirements, ISO 9001 requires that 1) the quality policy be defined, documented, understood, implemented, and maintained; 2) responsibilities and authorities for all personnel specifying, achieving, and monitoring quality be defined; and 3) in-house verification resources be defined, trained, and funded.

The biggest similarity between ISO 9001 and CMM is that both emphasize the documentation that contains the guidance for what should be done. There are also differences between them. Although there are specific issues that are not adequately addressed in CMM, in general the concerns of ISO 9001 are encompassed by the CMM. The converse is less true. This is reflected in the fact that ISO 9001 describes the minimum criteria (requirements) for an adequate quality management system rather than process improvement [1].

2.3. QUALITY FUNCTION DEPLOYMENT (QFD)

While all the quality models provide guidance for either the assessment of the current process or the achievement of a better process, they all share one common characteristic—these models defines “what to do” but leaving “how to do it” to individual companies. It is desirable to have a means to guide the companies in the development of action plans for SPI. These actions should be based on the software process requirements from relevant sources. Quality Function Deployment (QFD) is an appropriate tool for the translation of customer needs into products. Thus, it is helpful in serving the purpose of deriving action plans for SPI from software process requirements.

QFD, which was developed in the late 1960s in Japan by Professor Shigeru Mizuno and Yoji Akao, was introduced to the rest of the world, including European countries and the United States, in the early 1980s. It is a methodology for building the voice of the customer, both spoken and unspoken, into a product. The difference between QFD and other quality methodologies resides in the fact that, unlike traditional quality systems which aim at minimizing negative quality in a product, QFD adds values to the product by means of maximizing the positive quality [14]. Nowadays, QFD has been applied to virtually every industry and business, including software development [15][16].

The tools used in QFD are the Seven Management and Planning Tools, which are listed below:

1. Relations Diagram
2. Affinity Diagram
3. Tree Diagram
4. Matrix Diagram
5. Matrix Data Analysis Chart
6. Process Decision Program Chart
7. Activity Network

One important technique in QFD is the House of Quality (Figure 2.7). It is a table that connects the Voice of the Customer and the Voice of the Engineer. The House of Quality contains six major components:

1. **Customer requirements** (WHAT’s). A structured list of requirements derived from customer statements.

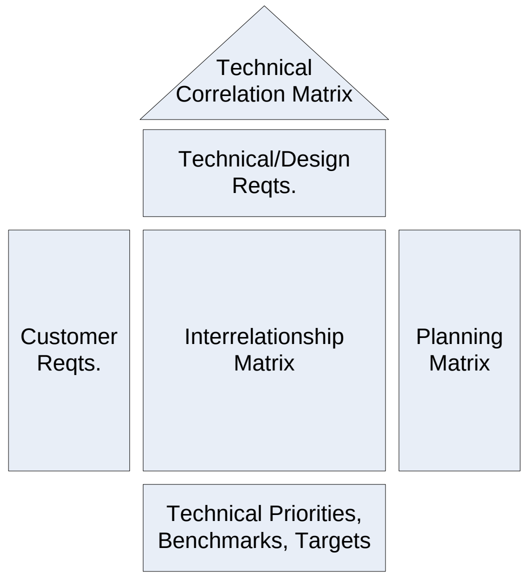


Figure 2.7. House of Quality in QFD

2. **Technical requirements** (HOW’s). A structured set of relevant and measurable product characteristics.

3. **Planning matrix**. Illustrates customer perceptions observed in market surveys. Includes relative importance of customer requirements, company and competitor performance in meeting these requirements.

4. **Interrelationship matrix**. Illustrates the QFD team's perceptions of interrelationships between technical and customer requirements. An appropriate scale is applied, which is illustrated by using symbols or figures. To fill this portion of the matrix involves discussions and consensus within the team, which can be time consuming. Concentrating on key relationships and minimizing the numbers of requirements are useful techniques to reduce the demands on resources.

5. **Technical correlation (Roof) matrix**. Used to identify where technical requirements support or impede each other in the product design. Can highlight innovation opportunities.

6. **Technical priorities, benchmarks and targets**. Used to record:

    - The priorities assigned to technical requirements by the matrix.
    - Measures of technical performance achieved by competitive products.
    - The degree of difficulty involved in developing each requirement.

When Professors Mizuno and Akao proposed the idea of QFD, this methodology was meant to include two components: a) Quality Deployment (QD) or Product Focused QFD; and b) Narrow definition QFD or Process Focused QFD [17][18]. The first component, as its name indicates, focuses on improving the quality of products by translating customer requirements into product features. This has been widely adopted by many industries world-wide. The second component, which focuses on improving the quality of processes, was designed to assure that organizational processes and actions are in compliance with established standards such as ISO 9000, ISO14000, and any other standards. For software companies, this "narrow definition QFD" can help them improve software development processes to the level specified in standards such as ISO 9001, CMM, etc. Unfortunately, this component has been neglected by most QFD followers in the business, especially in the field of software development [17][19].

2.4. PRIORITY ASSESSMENT

Similar to software implementation, requirements analysis is also important in software process improvement, especially in the prioritization of requirements. One of the difficulties in requirements prioritization is the fact that requirements are coming from different stakeholders with different interests. It should be understood that when different groups of stakeholders are involved, a key factor to successful software projects involves effectively negotiating the requirements among these stakeholders who have different roles and responsibilities (Boehm and In, 1996). It is essential to decide what is important before these requirements are incorporated into the software development process. By addressing the high-priority requirements before considering the low-priority ones, one can significantly reduce both the costs and duration of a project (Hofmann and Lehner, 2001). However, there is a lack of methodologies on the priority assessment of requirements from multiple perspectives. Requirements prioritization in the industry has been reported to be very informal and dependent upon experience and tacit knowledge (Lehtola et al., 2004). It is difficult enough for a stakeholder to decide which of his requirements are most important; achieving consensus among multiple stakeholders with diverse expectations and combining requirements from different sources is even more challenging (Wiegers, 1999; Lehtola et al., 2004).

The same holds true for the software process improvement. To alleviate risks at a later stage in software process improvement, one should initially identify the most important requirements that are mutually satisfactory to all stakeholders, which may include clients, end-users, developers, managers, quality assurance staff, and many other interested parties. Several factors concerning different stakeholders should be considered in priority assessment, such as business value, cost to deliver, risks, and the relation to other requirements.

There are several requirements prioritization methods proposed to date (Karlsson et al., 1998), each of which uses one or more different types of analytic or mathematical approaches to assist with requirements prioritization.

One such approach is the Analytic Hierarchy Process (AHP) proposed by Saaty, which uses exhaustive pair-wise evaluation by hierarchy level (Saaty, 1994; Saaty 1996). This approach has been commented on as being complicated and time-consuming, thus, it is impractical for large projects with many requirements (Finnie et al., 1995). Understanding this disadvantage of AHP, several researchers proposed to reduce the number of comparisons (Carmone et al., 1997; Harker, 1987; Karlsson, 1996; Karlsson et al., 1997; Shen et al., 1992). However, by reducing the number of comparisons, judgment errors may remain unidentified and the consistency may be decreased (Karlsson et al., 2004). Even worse, the reduced number of comparisons may still be overwhelming in practice (Lehtola and Kauppinen, 2004). In addition, AHP does not capture the correlations among requirements. There are also variations of AHP that have been proposed by researchers, such as Cased-Based Ranking (CBR), which uses machine learning techniques to overcome the shortcomings of computation explosion (Avesani et al., 2005).

Zultner proposed an interesting software requirements prioritization technique that involves multiple stakeholders or customers (Zultner, 1997). However, he also used AHP to determine the priorities of multiple customers. Karlsson used the AHP concept and developed a cost-value approach for prioritizing requirements (Karlsson and Ryan, 1997). He compared the requirements based on their relative importance and relative cost to implement in a pair-wise fashion. The resultant relative priorities were plotted on a cost-value diagram showing which requirements were to be used for current release of the software product. Even though Karlsson's techniques have advantages, when there are large sets of requirements, this technique may still cause a computational explosion. Also, this technique does not work for the prioritization of requirements from multiple perspectives, which may better reflect stakeholder needs in practice (Park et al., 1999).

Instead of using AHP, Frank Moisiadis presented a Requirements Prioritization Tool (RPT) (Moisiadis, 2002). RPT prioritizes requirements based on business goals and stakeholder viewpoints. A graphical fuzzy rating scale is used to elicit stakeholders' ratings, and dependencies among requirements are used for requirement prioritization. Although Moisiadis listed the limitations of commonly used requirements prioritization approaches such as QFD (Akao, 1990) and AHP, RPT does not overcome these limitations, which include the use of subjective ratings and ordinal scales. Furthermore, relationships between requirements from multiple perspectives are ignored.

There is also a model for distributed collaborative prioritization of software requirements (Boehm and Ross, 1989; Park et al., 1999), where disparately located stakeholders negotiate the relative priorities using priority bins. Although the model can identify conflicts between requirements during the renegotiation process as the software evolves, it does not address the interdependencies between prioritized sets of software requirements.

Siv Sivzattian and Bashar Nuseibeh proposed a portfolio-based approach to prioritize and select requirements (Sivzattian and Nuseibeh, 2001). This approach selects requirements based on the trade-off between effort and return. However, treating individual requirements as capital assets and applying the "U.S. capital market risk-free rates" and "average return rate" to the prioritization of requirements deserves more explanation and validation. Again, one disadvantage of this approach is that it is very difficult to apply. Furthermore, this approach does not consider the integration of requirements and priority assessment from various groups of stakeholders.

Some other researchers proposed the integration of multiple views in requirements specification using the term "viewpoint" (Easterbrook and Nuseibeh, 1995; Finkelstein and Fuks, 1989; Kotonya and Sommerville, 1992; Nuseibeh et al., 1997, 2000). Viewpoint-oriented analysis allows various specification methods to be used by owners of different viewpoints. While facilitating the tasks for each viewpoint, the integration of various viewpoint specifications still remains a challenging research area. In addition, prioritization of requirements from multiple perspectives is not considered in these studies.

A previous paper (Liu, 1998) proposed a model for requirements priority assessment. This model helps prioritize requirements from stakeholders who share the same concerns for the software product. However, this study was not able to prioritize requirements from multiple groups of stakeholders with different concerns because this was not the original focus.

3. EXISTING SPI METHODOLOGIES USING QFD

3.1. RICHARDSON'S APPROACH

Ita Richardson proposed a SPI model for small businesses, which uses the House of Quality in QFD to transform the business process requirements into the action plan [20]. As shown in Figure 3.1, the first stage of the model is optional. Businesses will be asked to focus on their business requirements, and to provide three measurements: current performance, planned performance, and the importance of this particular requirement on their business. The outcome from this section of the model would then be used in implementation Stage 2.

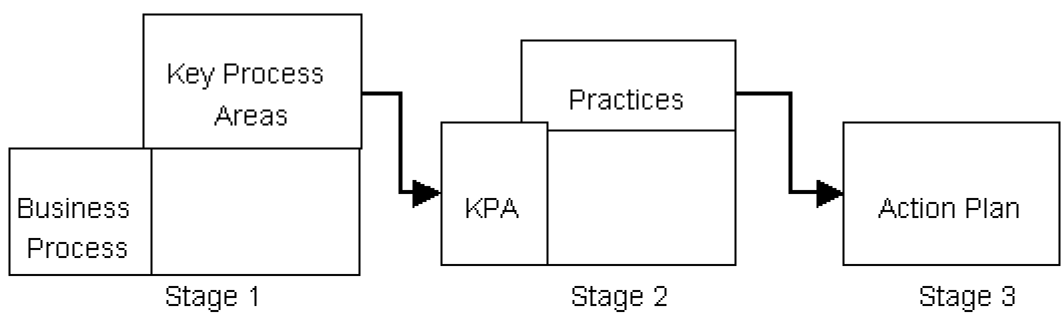


Figure 3.1. A SPI Model Using QFD for Small Businesses by Richardson

In Stage 2, companies can indicate how they currently perform in each Key Process Areas (KPA). They are also required to give measurements for their planned performance and the importance of that key process area to their business.

Once self-assessment has been input to SPI/HoQ in Stage 2, the priority practices to be pursued are now identified. These are then incorporated within a software process improvement action plan for implementation within the organization.

The above model was later improved into a four-stage model, as shown in Figure 3.2, for software process improvement in small companies [21].

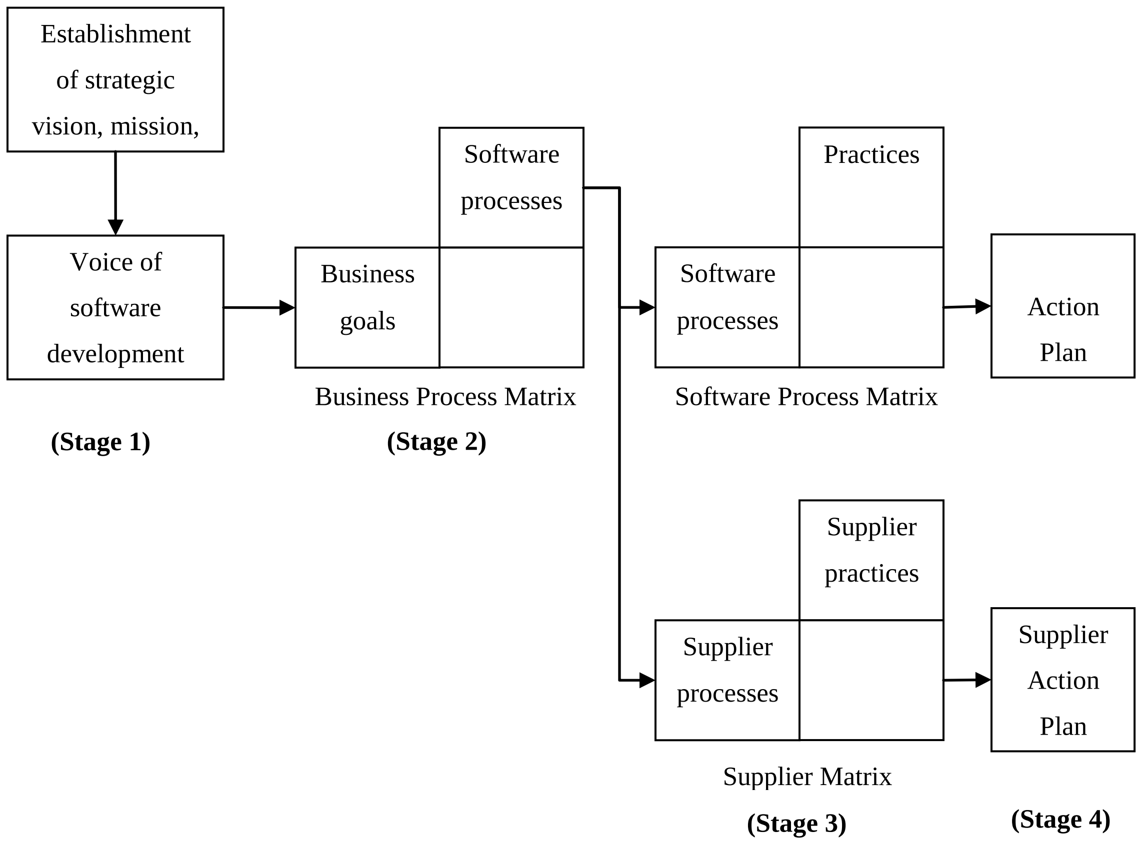


Figure 3.2. A Generic SPI/QFD Model by Richardson

In stage 1, business goals are identified from mission statement of the company. The identified goals represent the voice of this software development company. In stage 2, business goals are correlated with software processes using business process matrix. The measurements collected when identifying business goals are used to calculate the overall importance of the business goals. The importance of each software process is calculated by using overall importance of business goals and the value of relationships. In stage 3, Software processes and practices are correlated using software process matrix. The relationships used in the matrix are generic, indicating the effect that a practice will have on the process. The importance of each practice is calculated from process importance and relationship values. Similarly, importance of supplier practices is calculated using supplier processes. In stage 4, the action plan is derived. The action plan indicates the prioritized practices. Action plan helps the company to decide the order of important practices to improve upon, in order to influence software processes, and consequently business goals. Similarly, the supplier action plan is derived from supplier matrix.

Richardson's generic model [21] is useful for small companies. Small indigenous companies can establish practices that can be improved inexpensively and easily with little upfront investment. As this model is generic, small companies can use the matrices readily without extra efforts and investments.

Although the generic model generates a prioritized action plan, the measurements are based on self-assessment of the software process. In larger companies, however, the organizational structures become more complex, which makes self-assessment in this model more difficult. Also it does not deal with interrelationships between the practices in action plan, which correspond to roof of house of quality in QFD.

3.2. ZULTNER'S APPROACH

Zultner's Business Process Reengineering model with QFD is shown in figure 3.3. Zultner's model 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 [22].

In order to reengineer the business process, the organization must understand what the current process is and what needs to be improved. In the Analysis phase a Customer Needs/Process Requirements HoQ is used to find out what needs to be done in order to satisfy the customer requirements. A set of targets are identified and sent to the

Selection phase. At the same time, tasks in the current process of software development are visualized with the help of a Data Flow Diagram.

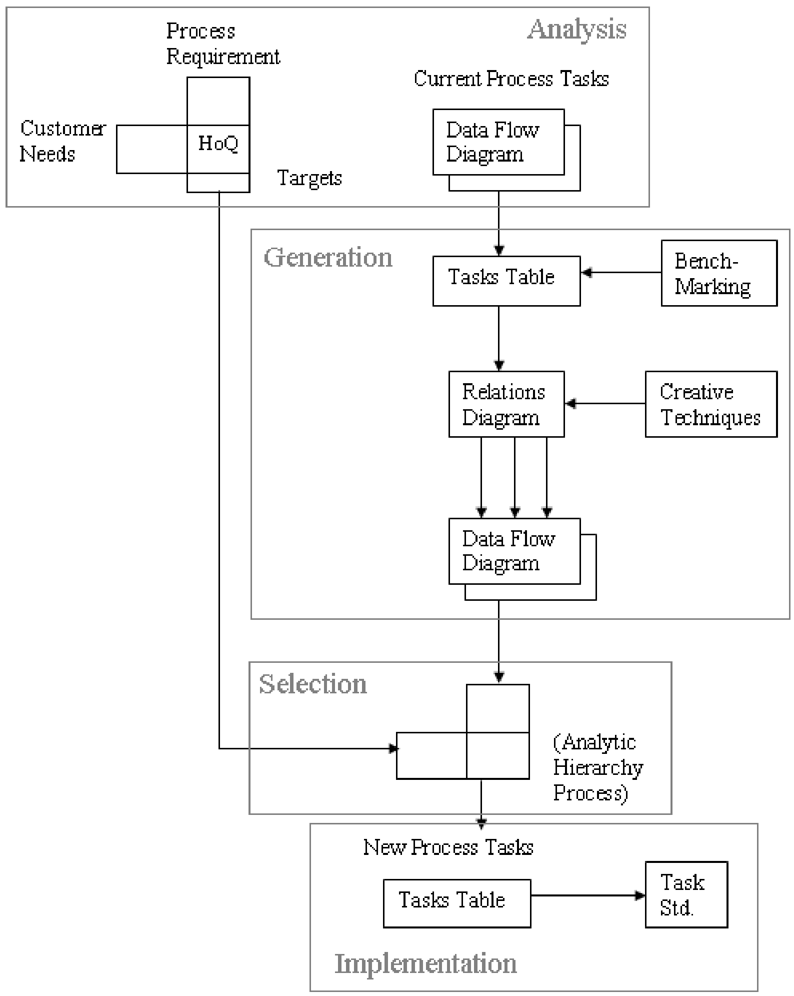


Figure 3.3. Zultner's QFD Process for Business Process Reengineering

New business process concepts are produced in the Generation phase. There are two possible sources for new process concepts. One is the comparison of process tasks from Analysis phase and benchmark results of competitors; the other one is creative ideas. New process alternatives are documented using whatever tools and techniques the new process employs.

In the Selection phase, new process alternatives are selected using Process Requirements/Alternatives matrix. The criteria in the Process Requirements/Alternatives matrix are the process requirements from the Customer Needs/Process Requirements HoQ in Analysis phase. With the help of Analytic Hierarchy Process (AHP), a ratio-scale priority of each alternative is generated. It shows precisely how much better the best alternative is over other choices.

Once the new process is selected, process tasks are generated and defined using tasks table in Implementation phase.

Zultner’s approach uses the seven managing and planning tools, or the seven “new” tools, as the basic toolset. QFD ensures that customer requirements are integrated into process requirements, and from process requirements into the new process. The new process is implemented by a set of tasks. Therefore, the final set of process tasks guarantee to be much better than the initial tasks from the customers’ point of view.

Zultner’s approach uses either the major competitor’s performance or creative thoughts of employees, but not existing standards which are widely used in a particular industry, as the source of process improvement. Although this approach may help address specific issues in an organization, it is difficult to apply this approach in different situations or environments to produce consistently efficient process improvement results when compared with a method using a popular reference model. This is because elements in this approach, such as creative thoughts, are not always dependable.

3.3. SAP’S QFD FOR SPI

The workflow of SAP’s approach for software process improvement as shown in Figure 3.4 starts with interviewing multiple stakeholders, such as developers, quality/ development/ product managers and program directors [23].

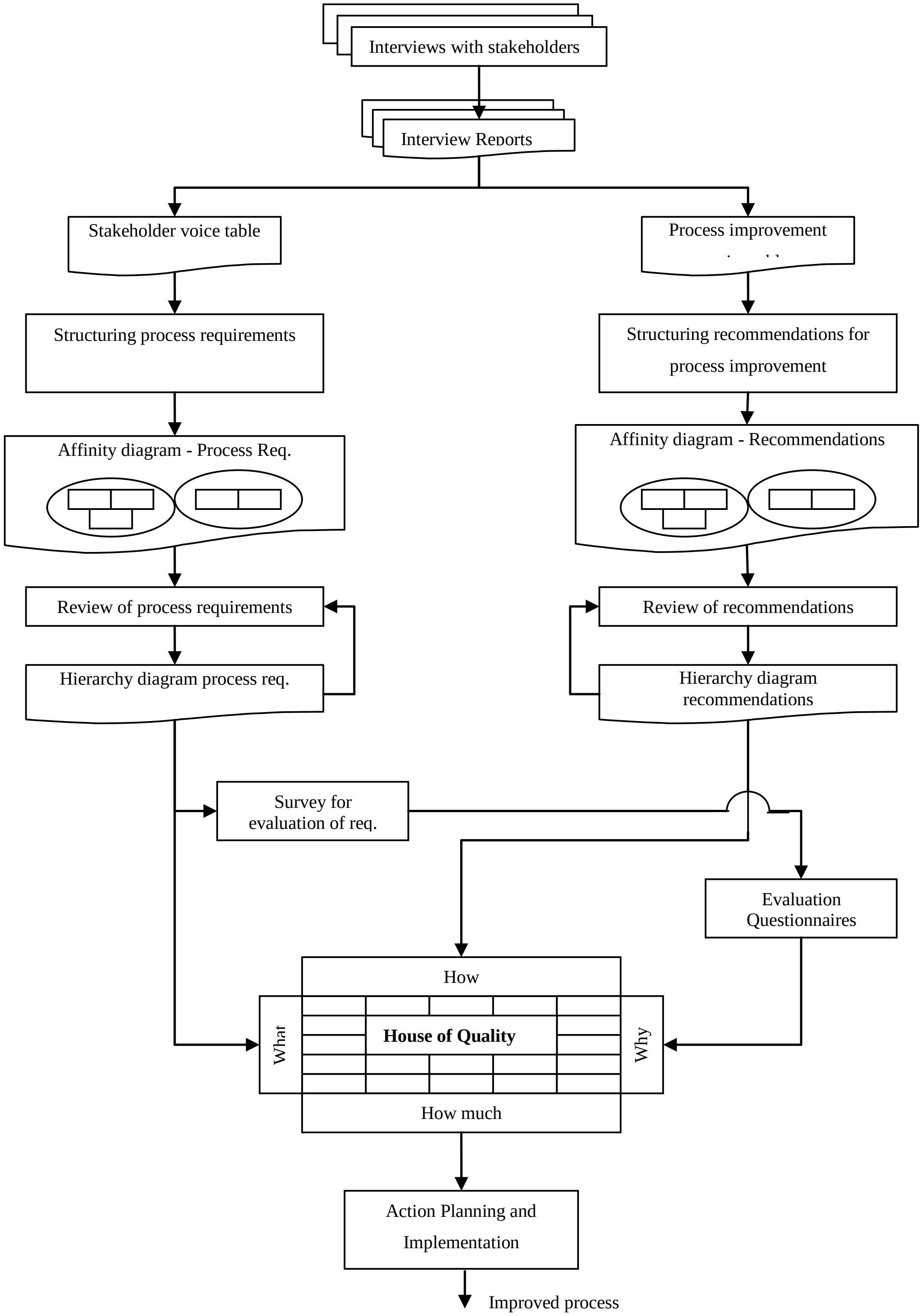


Figure 3.4. SAP's QFD for SPI

The outcome of interviews is the current problems and improvements necessary in software process. The problems form process requirements, and improvements form actions. Entries in each of these are grouped together on the basis of similar characteristics, using Affinity Diagrams. These groups are arranged hierarchically using Hierarchy Diagrams. The requirements are then prioritized using the Analytic Hierarchy Process (AHP) approach [24]. Global priorities are calculated as product of local priorities and group priorities. This forms the inputs to the house of quality matrix.

The process requirements form the “What” part of the house of quality, whereas improvements form the “How” part. Relationships are established between process requirements and improvements. The correlation ratings may take positive as well as negative 1, 3, 9 values. The process improvements get prioritized as outcome of this matrix. Prioritized improvements are used to build action plan.

SAP’s approach considers multiple stakeholders to form a set of process requirements. The advantage is that, it does not rely solely on business requirements. Furthermore, the requirements prioritization scheme is better, as it uses both hierarchy and groupings of requirements.

Although SAP’s approach considers multiple stakeholders, it treats requirements from all the stakeholders as equally important. It does not consider relationships between multiple perspectives. Also, the process improvements, which represent actions, are directly related to process requirements. Both are obtained from stakeholders. Quality models/standards, such as ISO or CMM, are not considered throughout the workflow. Thus, although the action plan is prioritized, the order of actions may be unreliable.

4. PROPOSED APPROACH

In this dissertation, SPI frameworks are developed to derive action plans based on software process requirements with the help of QFD and in accordance with CMM [25][26] and CMMI. The proposed frameworks integrate 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 the differences among different groups of stakeholders and lack of conformance to reference models.

CMM and CMMI are chosen in this framework because of their popularity in the industry and proven effectiveness. CMM, for many years, has shown positive results in terms of both tangible benefits such as cost, schedule, product quality, productivity, and amount of rework [27][28][29][30] and intangible benefits such as improvements in the quality of work life, organization communications; organization learning and efficiencies; the ability to attract, retain, and develop software professionals; and the coherency of its organization culture [31]. Similarly, SEI also reported the effectiveness of CMMI by comparing data from 35 organizations. Tangible benefits such as cost, schedule, productivity, quality, customer satisfaction, and return on investment (ROI) were obvious [32].

With either CMM or CMMI used in the framework, the SPI can be sketched from a high level as shown in Figure 4.1 below. The action plan is generated based on the process requirements through CMM or CMMI using QFD. This guarantees that the actions are in accordance with CMM/CMMI and, at the same time, satisfy the process requirements from organization.

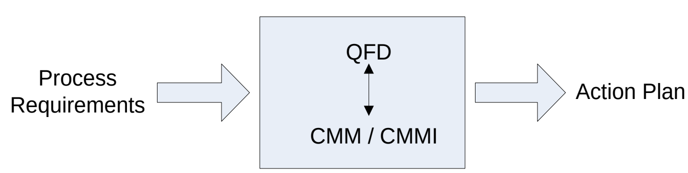


Figure 4.1. High-Level SPI Framework

In the rest of the dissertation, a unique method for requirements priority assessment will be introduced. This method serves as the starting point of the whole SPI frameworks. With the proposed requirements prioritization method, requirement priorities consider local weights, perspective weights, as well as correlation analysis results. This will increase the accuracy of require priorities and consequently increase the accuracy of the final deliverables of the SPI framework.

Following the priority assessment method, the detailed SPI framework for both CMM and CMMI will be introduced. Application examples will follow to illustrate the frameworks.

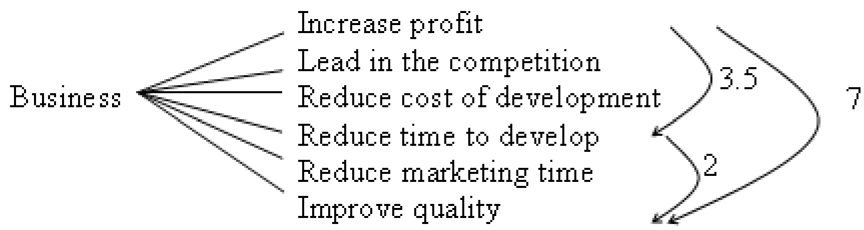
5. REQUIREMENTS PRIORITIZATION

5.1. PRIORITIZATION FRAMEWORK

During the requirements elicitation phase of a software process improvement project, the software requirements are collected from multiple groups of stakeholders, each having their own perception of the software process to be improved. The collection of requirements from each group of the stakeholders is called a perspective in this framework. It is important to realize that these perspectives of requirements are both independent to each other, in the sense that they are all important and none should be left out, as well as related to each other, in the sense that satisfying one might indirectly affect another. Problems emerge when various perspectives of software process requirements are put together because obviously they are not all equal. In order to ensure that the improved software process reflects the most critical needs from various perspectives, Correlation-Based Priority Assessment (CBPA) [33] is developed to prioritize and integrate these requirements so that the best available resources can be allocated to the most critical requirements.

The prioritization is performed both within each perspective and across various perspectives. Initially, requirements and the perspectives they come from are organized in the form of hierarchies (Figure 5.1a). Different perspectives form the roots in these hierarchical structures. These hierarchical structures can be prioritized using the prioritization scheme shown in Figure 5.1.

The prioritization can be performed either absolutely or relatively. In the case of absolute evaluation, each requirement/perspective is assigned an ordinal scale value between 1 and 5 points, which indicates how important it is to the stakeholders. This process requires less effort compared with the relative evaluation, especially when there are long lists of requirements elicited. However, using the absolute evaluation, stakeholders have a tendency to assign high values to all requirements under evaluation, which ultimately affects the quality of the software process. Hence, a relative evaluation as shown below is recommended instead.



(a) Requirements hierarchy and relative dominance

Perspective	Requirements	Local	Raw Initial Global	Initial Global
Business 3	Increase profit	7	3*7=21	0.1615
	Lead in competition	5	3*5=15	0.1154
	Reduce cost of development	5	3*5=15	0.1154
	Reduce time to develop	2	3*2=6	0.0462
	⋮			
Mngt 2	Within budget	7	2*7=14	0.1077
	On schedule	6	2*6=12	0.0923
	High customer satisfaction	4	2*4=8	0.0615
	Increase productivity	2	2*2=4	0.0308
⋮	⋮			

(b) Local priorities and initial global priorities

Figure 5.1. Prioritizing a Hierarchy of Requirements

Using relative evaluation, one must first identify the relative dominance values between perspectives and between requirements. Based on these relative dominance values, each perspective receives a perspective weight; at the same time, each requirement can be assigned a local priority value within the perspective it belongs to. The local requirement priorities and the corresponding local perspective weights are multiplied to produce the global priorities of the requirements, which are then normalized. The details are introduced as follows:

Step 1: Establish a linkage between each pair of perspectives by identifying the degree to which one perspective is relatively more dominant than the other. Considering the fact that some of the requirements may conflict with each other, it is possible that some of the requirements, or a group of requirements as a perspective, cannot be completely satisfied. In such cases, one can only say that they are satisfied to a certain degree, and this satisfaction degree can be adjusted. If the stakeholders can specify the percentage of the satisfaction degree of a perspective (P_i) that is needed to increase in order to compensate for the decrease in the satisfaction degree of another perspective (P_j), then the relative dominance of P_i over P_j, denoted by $rd_{i,j}$, can be obtained using the following equation [34]:

$$rd_{i,j} = n_j/n_i, \quad (1)$$

where n_i is the percentage of increase in the satisfaction degree of perspective P_i, and n_j is the percentage of decrease in the satisfaction degree of the perspective P_j.

Within each perspective, relative dominance values are identified for each pair of requirements using the same method in the above paragraphs. For instance, the stakeholders agree that in order to compensate for a decrease of 20% in the satisfaction degree of the requirement “reduce time to develop” in the Business Perspective, an increase of 10% in the satisfaction degree of “improve quality” is needed. Therefore, as shown in Figure 13(a), the relative dominance value of “reduce time to develop” over “improve quality” is 20% / 10% = 2. Similarly, an increase of 20% in the satisfaction degree of “increase profit” compensates the decrease of 70% in the satisfaction degree of “improve quality.” Thus, the relative dominance value of “increase profit” over “improve quality” is 70% / 20% = 3.5.

The numeric representation of the relative dominance *rd* typically comes from a consensus by all stakeholder representatives. If the relative dominance values given by stakeholder representatives vary, then discussions are needed and a uniformly agreed upon value must be generated. It is also important to ensure that the relative dominance values are consistently assigned. The relative dominance values are said to be consistent if and only if the following exists (Liu 1998a):

$$1 \leq i, j, k \leq n \text{ such that } rd_{i,k}=rd_{i,j}*rd_{j,k} \quad (2)$$

As shown in Figure 5.1(a), if one can determine a relative dominance value of seven (7) for "increase profit" over "reduce time to develop," then one can show that the relative dominance values above are consistently assigned because 7 = 3.5 * 2, in which 3.5 is the *rd* of "increase profit" over "improve quality," and 2 is the *rd* of "improve quality" over "reduce time to develop." The relative dominance values will be used to assign local priority values to perspectives and requirements.

Step 2: After establishing the relative dominance of one perspective over the other, the local priority of each perspective can be calculated. In order to help with the understanding of this scheme, the local priorities of different perspectives are called "perspective local priorities." Suppose that there are n perspectives, P_1, P_2, …, P_n, in a decreasing order of importance. Let W_{Pi} denote the numeric priority of perspective P_i. First, W_{Pn} (the priority of perspective P_n, which has the lowest importance) is assigned a base value of one (1). Then the perspective local priorities of all the remaining perspectives can be determined recursively using the following equation [34]:

$$\text{for } 1 < i \leq n,\ W_{Pi-1} = W_{Pi} * rd_{\,i-1, I} \tag{3}$$

In Figure 5.1, the priority value of the management perspective has been calculated as two (2). Because a relative dominance value of 1.5 has been determined for the business perspective over the management perspective, one can calculate the priority value for the business perspective by using the Equation (3):

$$W_{P\text{-}business} = W_{P\text{-}mgnt} * 1.5 = 2 * 1.5 = 3$$

Notice that the result of the above equation is not an ordinal value assigned arbitrarily by stakeholders. Instead, it is calculated using the relative dominance value between two perspectives.

Step 3: Within each perspective, the requirements are prioritized using the same method described in Step 2 to derive the requirement local priorities. Starting by assigning a value of one (1) to the requirement with the lowest priority, the local priorities of all the remaining requirements in the same perspective can be determined recursively using Equation (4):

$$\text{for } 1 < i \leq n,\ W_{R_{i-1}} = W_{R_i} * rd_{\,i-1,\,I} \tag{4}$$

In Figure 5.1(b), the priority value of the requirement of “improve quality” in the Business Perspective has been calculated as two (2). Because a relative dominance value of 3.5 has been determined for “increase profit” over “improve quality,” the priority value for “increase profit” by can be calculated by the following Equation (4):

$$W_{increase\ profit} = W_{improve\ quality} * 3.5 = 2 * 3.5 = 7$$

Step 4: For each requirement, calculate the raw initial global priority, which reflects both its local priority and the priority of the perspective it belongs to. For requirement X from perspective Y, the raw initial global priority is calculated by multiplying its requirement local priority and the perspective local priority as follows:

$$\text{Raw Initial Global Priority}(X) = \text{Perspective Local Priority}(Y) * \text{Requirement Local Priority}(X) \tag{5}$$

For instance, in Figure 5.1(b), the raw initial global priority for the requirement “Increase profit” (with a local priority of 7) in the Business Perspective (with a perspective local priority of 3) can be calculated using Equation (5):

$$\text{Raw initial Global Priority}_{\,(Increase\ profit)} = 3 * 7 = 21$$

Step 5: The raw initial global priorities of all requirements are normalized across perspectives to obtain the initial global priorities. This research did not normalize the local priorities within perspectives because the locally normalized priorities will be affected by the number of requirements in the perspectives. If the numbers of requirements in two perspectives differ a lot, then the normalized local priority values from the two perspectives will not be comparable. Thus, this research normalized raw initial global priorities across perspectives so that the perspective priorities were retained and the normalized results were not affected by the number of requirements in each perspective.

For instance, in the Business Perspective, the raw initial global priority for the requirement “increase profit” was calculated to be 21. This was done for all other requirements. These raw initial global priorities from all perspectives were normalized to 1. After normalization, the initial global priority of the requirement “increase profit” became 0.1615.

5.2. INTEGRATION AND ASSESSMENT OF REQUIREMENT PRIORITIES FROM TWO PERSPECTIVES

This section discusses an approach in CBPA to integrate prioritized stakeholder requirements from two perspectives into a single concise set of prioritized requirements.

A simple relationship matrix as shown in Figure 5.2 can form a basis for integrating stakeholder requirements from two perspectives. When two perspectives are identified, the prioritized requirements from both perspectives are integrated and re-prioritized using the relationship matrix by establishing their correlations on each other. The following steps discuss the components that constitute the relationship matrix and how they are completed.

1. Enter the stakeholder requirements from two perspectives into the columns and rows of the relationship matrix.
2. Enter the initial global priorities: These are the sets of normalized initial global priorities obtained in the previous section.
3. Determine the correlation relationships: Every requirement from Perspective 1 is carefully examined against every requirement from Perspective 2, and a correlation relationship is assigned using the symbols shown in Table 5.1. Various criteria can be adopted to determine the correlations depending on the schedule, budget, and existing approaches being used by the project.

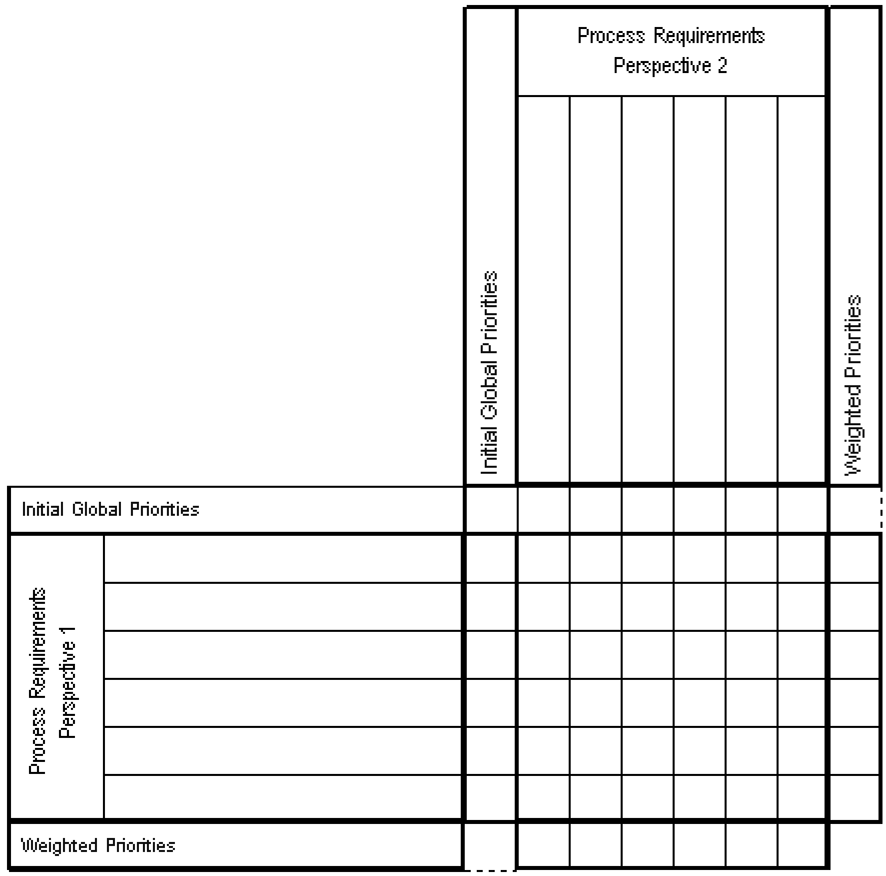


Figure 5.2. Relationship Matrix

Carlshamre et al. have proposed a set of five different interdependency types between requirements [35]. The relationships can be either one of the AND, REQUIRES, or TEMPORAL interdependencies, which according to Carlshamre are the three types with the highest priorities. Because the requirements prioritization scheme is intended to facilitate the requirements analysis, this study purposely avoids the use of function-driven approaches to calculate the impact relationships because they involve multiple factors related to the interdependencies of requirements, and there is no consensus on such a set of relevant factors. Choosing a function-driven approach will complicate the prioritization scheme, which is exactly the opposite of the aim of this research. The correlation types from QFD as shown in Table 5.1, on the other hand, have been widely accepted by industrial and academic practitioners.

Table 5.1. Types of Correlation Relationships

Impact	**Symbol**	**Value**
Strong	●	9
Medium	○	3
Weak	▽	1

4. Calculate weighted priorities: For requirement X from one of the two perspectives, the weighted priority is calculated by Equation (6):

$$\textit{Weighted Priority}(X) = \Sigma_y (\textit{Initial priority}(X) * \textit{correlation}(X, Y) * \textit{Initial priority}(Y)) \quad (6)$$

where Initial priority(X) is the initial global priority of requirement X, and correlation (Y, X) is the correlation value between requirement X and requirement Y in another perspective.

After the weighted priority values are calculated for all the requirements in the relationship matrix, they are normalized. For requirement X from either Perspective 1 or Perspective 2, the normalized priority is calculated by Equation (7):

$$\textit{Normalized Priority}(X) = \textit{Weighted priority}(X) / (\Sigma_{k(P1)} \textit{Weighted Priority}(k_{(P1)}) + \Sigma_{k(P2)} \textit{Weighted Priority}(k_{(P2)})) \quad (7)$$

In the above equation, $k_{(P1)}$ and $k_{(P2)}$ are the number of requirements in perspectives 1 and 2, respectively.

There are two approaches for the normalization of weighted priorities: a) normalization within each perspective, and b) normalization across all perspectives. Each of the two approaches has its own advantages and disadvantages.

- *Normalization within each perspective:* In this approach, weighted priorities of requirements are normalized to 1 within each perspective. As a result, the requirements are comparable with each other within their own perspectives, and the perspectives are kept separate from each other. Because the perspective priorities have been used in Section 2 to produce the raw initial priority values for individual requirements, it is inappropriate to multiply perspective priorities with the normalized weighted priorities of requirements again. As a result, the total influence from each perspective is the same, which means the perspective priorities are lost and the normalized priorities of requirements from one perspective cannot be compared with those from other perspectives. In other words, the requirements from different perspectives can not be merged to form a single set of requirements. Therefore, this approach was not adopted.
- *Normalization across all perspectives:* In this approach, weighted priorities across all perspectives are normalized to 1. Thus, the normalized priorities of requirements from different perspectives can be compared with each other. The resultant normalized priority indicates the influence of a particular requirement in the complete set of requirements from all perspectives. Both perspective priorities and local priorities for individual requirements are preserved. However, perspective boundaries are lost once all requirements are normalized. The requirements prioritization and assessment framework of this research adopts this approach because in the methodology perspective priorities need to be retained along with individual requirement priorities when all requirements from different perspectives are integrated into a single set.

Normalized priorities capture the relationships between requirements from different perspectives, and these can be used to adjust the initial global priorities with the help of an adjustment factor α. The α value indicates the importance of the correlations between requirements relative to the initial global priorities, and it ranges from 0 to 1. If the correlations between requirements are considered to be as important as the initial priorities, then α = 1 is used in the calculation of adjusted priorities. As the relative importance of the correlations decreases, the α value decreases accordingly until it become 0, which means that the correlations are negligible compared with the initial priorities. The systems analysts in charge of requirements prioritization should decide the alpha value based on their experience with previous projects and their understanding of the current project. From either Perspective 1 or Perspective 2, the adjusted priority for requirement X is calculated by the following equation:

$$\textit{Adjusted Priority}(X) = \textit{Initial global priority}(X) + \alpha * \textit{Normalized priority}(X) \tag{8}$$

As the above equation shows, the initial global priority serves as the base value of the adjusted priority. The normalized priority, which represents the impacts from the requirements of the other perspectives, is the amount to be added to the base value. Also, the α value controls the percentage of the additional amount that should be used in the adjusted priority. For instance, suppose a requirement has an initial global priority of 0.34 and its normalized priority is calculated as 0.25. When α =1, the adjusted priority of this requirement is the sum of its initial global priority and it normalized priority as shown using Equation (8):

$$\textit{Adjusted Priority} = 0.34 + 1 * 0.25 = 0.59$$

On the other side of the spectrum, when α = 0, the adjusted priority is simply the initial global priority, as shown using Equation (8):

$$\textit{Adjusted Priority} = 0.34 + 0 * 0.25 = 0.34$$

As shown in the above example, the adjusted priority value of the same requirement is higher when the adjustment factor is larger. When a requirement has no relationship with any of the requirements from the other perspective, one can still use the same equation to calculate the adjusted priority, which is exactly the initial global priority of that requirement regardless of what α value is used, because the normalized priority is always zero (0).

The α value should be determined by consensus from the stakeholders. The initial global priorities of requirements can be used directly for requirements integration by skipping the correlation analysis, which has the same effect of assigning zero (0) to α. However, this will ignore the correlations between requirements from different perspectives. Initial global priorities represent the importance of requirements within their own perspectives, yet, some requirements may correlate with requirements of other perspectives. When such correlations are considered to be important, these requirements should get higher importance globally. Satisfying such requirements may help satisfy other correlated requirements to some extent. Thus, actions to satisfy such requirements should receive more resources to maximize the customer satisfaction and to improve the process. If these correlations are not considered when requirements are prioritized and integrated, then some requirements may receive more resources and attention than they deserve, while other requirements may get less. On the other hand, with these correlations the prioritization of requirements becomes more accurate.

After re-assessing the priorities from two perspectives, the two individual sets of prioritized requirements can be integrated by using the adjusted priorities calculated. The deliverable is a single set of ranked software process requirements.

5.3. PRIORITIZATION AND ASSESSMENT OF REQUIREMENT PRIORITIES FROM MORE THAN TWO PERSPECTIVES

In practice, it is common for a software process improvement project to have more than two different perspectives of requirements. Integration of requirements from three perspectives is more complicated than the integration of two perspectives because only one relationship matrix cannot capture the correlations among three different perspectives. Below is the discussion on how to integrate software process requirements from more than two perspectives in CBPA.

5.3.1. Identification of All Perspective Pairs. First, correlations between requirements from each pair of perspectives are generated using relationship matrices. A table is used to identify all perspective pairs. As illustrated in Figure 5.3, all perspectives are listed in both columns and rows. Because the correlation between one perspective and itself is not considered, the diagonal is not used. Only half of the table is used (above or below the diagonal) because each relationship matrix captures the impact relationship in both directions (from one perspective to the other and vise versa). According to Figure 15, if there are N perspectives, then N*(N-1)/2 relationship matrices are needed. Typically, the number of perspectives is small in practice with not too many relationship matrices constructed. Each one of the shaded cells in Figure 5.3 represents a relationship matrix that will be used as illustrated in Figure 5.2. For instance, M_{1-2} represents the relationship matrix between requirements from Perspective 1 and Perspective 2. It will generate the weighted priorities for requirements in Perspective 1 with influence from Perspective 2, and the other way around.

5.3.2. Combining Weighted Priorities. The construction of each individual relationship matrix is exactly the same as that introduced in the previous section. The initial global priorities come from the calculation as introduced in Section 5.1. When all relationship matrices are completed, all weighted priorities must be combined before calculating the adjusted priorities. Given a total of N different perspectives, there will be N-1 weighted priorities for each requirement, one from each matrix. Each weighted priority indicates the degree of impact on that requirement from a different perspective. All weighted priorities of one requirement are added to produce the final priority. In this way, the final priorities can be calculated for all requirements from all perspectives. Let $WP_j^{i,k}$ be the weighted priority of requirement j in perspective i with impact from perspective k, which is obtained from the relationship matrix. If there are N perspectives, then the final priority of requirement j is calculated by adding all its weighted priorities using the following equation.

$$FP_j^i = \sum_{k=1}^{N} WP_j^{i,k} \quad (k \neq i) \tag{9}$$

	P_1	P_2	P_3	…	P_N
P_1		$M_{1\text{-}2}$	$M_{1\text{-}3}$	$M_{1\text{-}\ldots}$	$M_{1\text{-}N}$
P_2			$M_{2\text{-}3}$	$M_{2\text{-}\ldots}$	$M_{2\text{-}N}$
P_3				$M_{3\text{-}\ldots}$	$M_{3\text{-}N}$
…					$M_{\ldots\text{-}N}$
P_N					

Figure 5.3. Perspective Integration

Suppose that there are three perspectives and a requirement with an initial global priority of 0.1615 receives two weighted priority values of 0.3404 and 0.1119 from the two relationship matrices between its perspective and the other two perspectives. The final priority is calculated as 0.3404 + 0.1119 = 0.4523.

After the final priorities of all requirements are calculated, they are normalized in the same way as the weighted priorities were normalized in the previous section. An adjustment factor α is used together with the normalized priorities and the initial global priorities to calculate the adjusted priority value with the help of the same equation as introduced in Section 5.2. Suppose that after normalization, the requirement above with the final priority of 0.4523 receives a normalized priority of 0.1902. With an α value of one (1), the adjustment priority of this requirement is calculated as follows:

$$\textit{Adjusted Priority} = 0.1615 + 1 * 0.1902 = 0.3517$$

The advantage of this requirement integration method for three or more perspectives, like the integration and assessment for two perspectives, is that the final priorities reflect both local priorities, which are relative priorities of requirements within each perspective, and perspective priorities. Furthermore, when three or more perspectives are involved, the final priority of each requirement will not be affected by the order of impacts considered from other perspectives. No matter which impact is calculated first, the final priority values will not change as long as the impact values between requirements are not altered. In other words, the requirements integration results will not be affected by the order in which relationship matrices are constructed because each relationship matrix is constructed using only the initial global priorities of the requirements in the two perspectives involved. Because the initial global priorities are calculated before the construction of relationship matrices, the results obtained from one relationship matrix will not affect the results of another one.

6. SPI FRAMEWORK BASED ON CMM USING QFD

6.1. THE FRAMEWORK

CMM is used in the framework as the reference model because of its popularity in the industry. Although the support for CMM from SEI has discontinued and CMMI has been recommended since then, it takes time for many companies currently using CMM to switch to CMMI.

QFD is used to help an organization achieve three objectives. First, business and other requirements within an organization are mapped to CMM goals and activities. A connection is established so that the organization can clearly see how CMM helps with its business goals. Second, software process requirements from multiples 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 Key Process Areas (KPAs) and Key Practices (KPs) in CMM. Therefore, the ordering of the actions taken is based on how they are related to both the software process requirements and the corresponding KPs in CMM. For instance, an action (A1) derived using this approach is strongly related to a KP (KP1) in CMM, while another action (A2) is strongly related to KP2. Suppose that according to the mapping developed from this framework, it is found that KP1 reflects the requirements more than KP2 does. As a result, A1 should have priority over A2. This guarantees that the actions are in accordance with CMM 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 the proposed framework all the way down to the action plans. The requirements from multiple perspectives are correlated with each other using the priority assessment technique introduced in Section 5. As a result, the priority value of each requirement is adjusted after the impacts from the other requirements are assessed.

The set of requirements with adjusted priorities are related to the key goals in CMM KPAs. The goals are prioritized based on those process requirements. Thus, the goals that achieve higher overall satisfaction of process requirements get higher importance. In order to achieve these goals, CMM has KPs categorized into five common features. Both the common features and the KPs contained in them can have different priorities. The priorities of the common features are determined by their natures in CMM. For instance, “Commitment to Perform” should be considered before “Verifying Implementation.” The priorities of KPs in various common features, on the other hand, are determined by their correlations with KPA goals. Thus, the KPs in each common feature are prioritized separately based on the priorities of the goals. KPs that aim to achieve higher overall satisfaction of key goals receive higher importance values. Separate sets of action plans are derived from KPs in each of the common features. The actions that help to support more important KPs receive higher priorities.

As a result, the process requirements are reflected in KPA goals, KPs, and the actions. The actions both follow the process maturity standards in CMM and satisfy the process requirements. Those actions with higher importance values help to achieve higher process requirements satisfaction.

As illustrated in Figure 6.1, this framework starts with the elicitation and integration of requirements. In this phase, the requirements for the improvement of the organizational process are gathered from various branches/departments, including the business goals from the executive board. For instance, one of the business goals may state that “Our product should lead in the competition,” or a software process requirement from the management level may be that “The employee productivity should be increased.” Depending on which branches and departments they come from, these software process requirements are grouped into perspectives with each branch/department being a perspective.

In Figure 6.1, 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 [34]. In Figure 6.1, 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 the next phase.

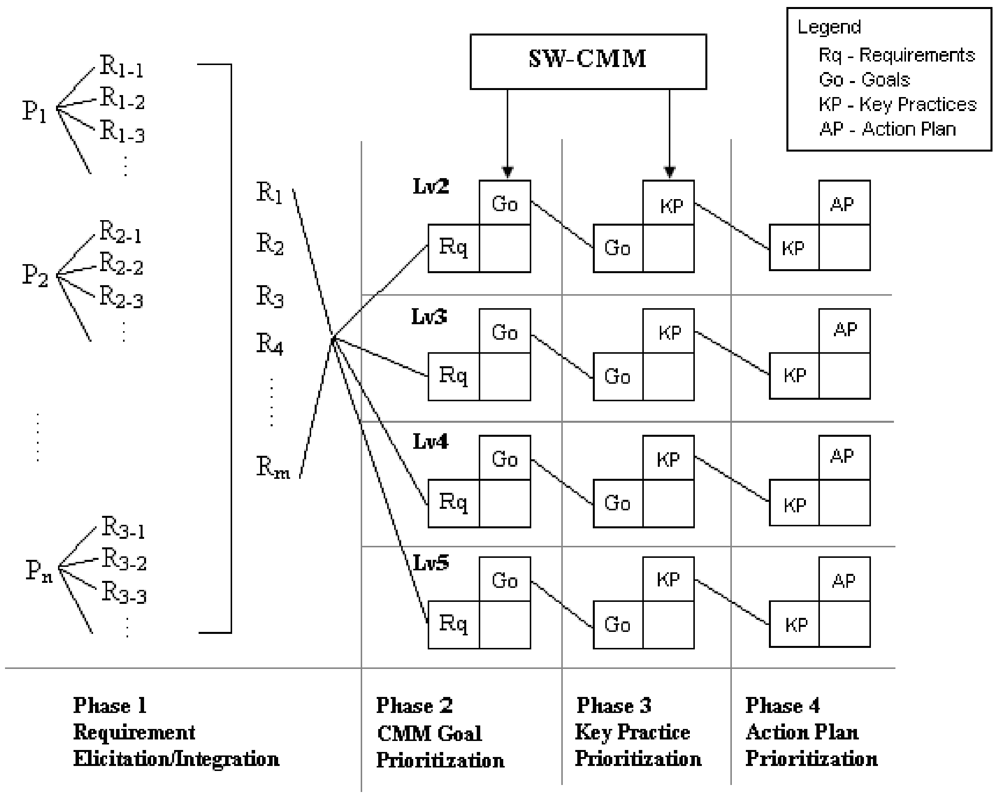


Figure 6.1. Software Process Improvement through CMM Using QFD

The second through fourth phases of this framework are applied to Level 2 to Level 5 of the CMM model. The prioritized and integrated requirements from Phase 1 are linked to all KPA goals in each of the four levels in CMM using relationship matrices (as introduced in Section 3). These prioritized KPA goals are used as the basis for the prioritization of KPs. Finally, the prioritized KPs are transformed into prioritized action plans using House of Quality (HoQ).

In the second phase, which is “CMM goal prioritization,” the goals of all KPAs in a particular CMM level are selected and prioritized based on the requirements from the previous phase. There are two objectives of this framework and this phase is significant in terms of achieving both. First, the organization needs to comply with the CMM standard. At the same time, the organization needs to ensure that by reaching a particular maturity level, the process is also satisfying the business and other requirements within the organization. In Phase 2, a relationship matrix is used to establish connections between the requirements from the organization and KPA goals in CMM. This matrix demonstrates that complying with the CMM standard also helps satisfy the business and other requirements in the organization. Second, the final set of action plans needs to be prioritized based on the priorities of requirements so that more important actions receive more resources. KPA goals serve as the bridge between requirements and the action plan. By prioritizing KPA goals, requirements from the organization can be transformed to the KPs in the third phase, and finally to the action plans in the final phase. In this way, a set of actions can be executed not only to achieve a specific maturity level in CMM, but also to satisfy organizational process requirements.

The third phase of the proposed framework, which is “key practice prioritization,” involves the prioritization of KPs within all KPAs of a specific level. The prioritization is carried out on the basis of the deliverables from Phase 2. According to CMM specifications, all these KPs have to be performed in order to reach that particular maturity level. However, these KPs serve as a bridge between the requirements and the final actions, and it is necessary to know how these KPs reflect the software process requirements. In order to show the connections between the requirements and the final action plans, these KPs have to be prioritized based on KPA goals, which are now reflecting requirements priorities. The mapping between KPA goals and KPs has been provided in Appendix E of the 1995 SEI CMM book [11], and it can be modified if necessary.

In the fourth phase of the framework, which is “action plan development and prioritization,” a set of actions is derived from the prioritized KPs. These actions should reflect the requirements integrated in the first phase. Meanwhile, they also state what needs to be executed in order to reach a particular CMM maturity level. These actions guide the process improvement. Thus, more resources should be assigned to those actions with high priorities.

The above framework addresses the problem that CMM specifies only "what to do" but not "how to do." By incorporating requirements from the organization into action plans through KPA goals and KPs, the connection between the objectives of the organization and CMM maturity levels becomes clear.

6.2. MATRICES IN FRAMEWORK BASED ON CMM USING QFD

Four different matrices are used in the framework based on CMM. This section introduces these matrices used in each of the four phases.

6.2.1. Requirements Impact Matrix (RI Matrix) in Phase 1. The requirements prioritization technique introduced in Section 4.1 can be used in Phase 1 of the framework to integrate requirements from all perspectives into one single set. 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 by following the five steps in Section 5.1.

After all requirements receive their normalized global priorities, RI Matrix is used to integrate and prioritize these requirements. Figure 6.2 shows an example of the RI Matrix. It uses the relationship matrix as introduced in Section 5.2. Depending on the number of perspectives, these requirements are integrated by following either Section 5.2 or Section 5.3.

6.2.2. Requirements-Goals Impact Matrix (RG Matrix) in Phase 2. In this phase, the Requirements-Goals Impact Matrix (RG Matrix) is used to prioritize KPA goals on the basis of the adjusted priorities (APs) of requirements that come from the previous phase. Their correlations with the requirements are reflected in the matrix, and a value indicating the relative importance for each goal statement is calculated.

Figure 6.3 shows an example of the RG Matrix using CMM Level 3 KPAs. The following five steps guide through the process of building the Requirements-Goals Impact (RG) Matrix:

	Initial Global Priorities	Low failure rate	Low defect rate	High reliability	High requirement satisfaction	High maintainability	High usability	Weighted Priorities (P1-3)
Initial Global Priorities		0.0462	0.0308	0.0308	0.0231	0.0154	0.0077	
Increase profit	0.1615	○	○	○	●	●	○	0.1119
Lead in competition	0.1154	●	○	○	●		○	0.0960
Reduce cost of development	0.1154				○	○		0.0133
Reduce time to develop	0.0462				○		○	0.0043
Reduce marketing time	0.0462	▽	○		▽			0.0075
Improve quality	0.0231	○	●	●	●	○	○	0.0224
Weighted Priorities (P3-1)		0.0757	0.0363	0.0320	0.0746	0.0288	0.0080	

Figure 6.2. Requirements Impact (RI) Matrix

1. Enter the integrated requirements (deliverables from the previous phase) along with their adjusted priorities into the rows.
2. Enter the goals of all KPAs of a particular maturity level in CMM into the columns. These goals are grouped based on the KPAs from which they come.

One big matrix can be broken into multiple smaller matrices, each of which contains grouped goals from one or more KPAs.

	Requirements Importance	Organization Process Focus (KPA 1)	G1: Process development and improvement activities are coordinated	G2: Strengths and weaknesses of software processes are identified	G3: Organization-level process developt and improv activities are planned
R1: Increase profits	0.3517			○	○
R7: Within Budget	0.2673		○		▽
R2: Lead in competition	0.2401		○	●	●
R8: On schedule	0.1936		●	○	○
Weighted Importance (FG)			3.2646	3.7968	4.0641
Normalized Importance (NG)			0.2934	0.3413	0.3653

Figure 6.3. Requirements-Goals Impact (RG) Matrix

3. Determine the correlation between each requirement and each goal. The same set of symbols in Table 5.1 is used in this matrix.
4. Calculate the weighted importance values for goals (FG) using the following equation:

$$FG_i = \sum_{j=1}^{M} AP_j * IR(G_i, R_j) \tag{10}$$

5. Normalize the weighted importance values.

In order to illustrate the above steps using an example, the four requirements with the highest adjusted priorities from Phase 1 were chosen and entered into the rows of Figure 6.3. For simplicity's sake, only three key goals from CMM Level 3 were selected and entered into the columns. After determining and entering the correlations between the requirements and key goals into the matrix, the weighted importance of each goal can be calculated using Equation (10). For instance, the weighted importance value of G1 is calculated as follows:

$$FG_1 = \sum_{j=1}^{4} AP_j * IR(G_1, R_j)$$

$$= 0.3517*0 + 0.2673*3 + 0.2401*3 + 0.1936*9 = 3.2646$$

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 (NG) serve as the input to the next phase.

6.2.3. Goals-Practices Impact Matrix (GP Matrix) in Phase 3. In the third phase of the framework, the Goals-Practices Impact Matrix (GP matrix) decides the importance of KPs in CMM based on their relationships with the prioritized goals. Because the KPs are categorized into common features in CMM, KPs in each common feature are prioritized using a separate GP matrix. Common features are five groups of KPs that are used by organizations to institutionalize their processes. Typically they are all mandatory for the achievement of a particular CMM maturity level. Therefore, it does not make sense to discard any common features or the KPs in them. The only reason to prioritize these KPs in the common features is to reflect the process requirements priorities and pass them to the actions in the next step.

Because the goals reflect the process requirements, by relating KPs with the goals, the priorities of KPs should also reflect the requirements priorities. An example of the GP matrix based on CMM Level 3 KPAs following Figure 6.3 is shown in Figure 6.4. The following five steps are followed in order to develop a GP matrix:

	Goals Importance	KP1: Process is assessed periodically, action plans developed to address assessment findings	KP2: Use of organization's software process database is coordinated at organization level	KP3: Software engineering methods and tools are integrated into defined software process
Organization Process Focus (KPA 1)				
G1: Process development and improvement activities are coordinated	0.2934	○	○	
G2: Strengths and weaknesses of software processes are identified	0.3413	●	▽	
G3: Organization-level process development and improvement activities are planned	0.3653	●	○	○
Weighted Importance (FKP)		5.0478	2.3174	1.0959
Normalized Importance (NKP)		0.5966	0.2739	0.1295

Figure 6.4. Goals-Practices Impact (GP) Matrix

1. Enter the prioritized goals in the RG matrix together with their normalized importance values (NG) into the rows.
2. List the KPs in CMM into the columns of the matrix.
3. Enter the correlation between each goal and each KP based on Goal-KP mapping provided in Appendix E of SEI CMM book. However, they treat all correlations as equally important. This can be modified by introducing "strong," "medium," and "weak" correlations, as shown in Table 5.1.

4. Calculate the weighted importance values (WKP) of the KPs using Equation (11):

$$FKP_i = \sum_{j=1}^{s} NG_j * IR(KP_i, G_j) \tag{11}$$

5. Normalize all weighted importance values of KPs into NKPs.

The same three key goals in Figure 6.3 are entered in the rows of Figure 6.4. Only three KPs are used in this example for simplicity's sake. The correlations are determined and entered in the matrix. Based on these correlations, the weighted importance value for each of the KPs can be calculated using Equation (11). For instance, the weighted importance for KP1 is calculated as follows:

$$FKP_1 = \sum_{j=1}^{3} NG_j * IR(KP_1, G_j)$$

$$= 0.2934*3 + 0.3413*9 + 0.3653*9 = 5.0478$$

After all three weighted importance values are calculated, the KP Weight (WKP) values are normalized to obtain the normalized importance.

6.2.4. Action Plan House of Quality Matrix (AP-HoQ) in Phase 4. Action plans are developed on the basis of KPs, 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, KPs 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.

An example of the AP-HoQ matrix based on CMM Level 3 KPAs is shown in Figure 6.5. The following six steps are followed in order to develop an AP-HoQ matrix:

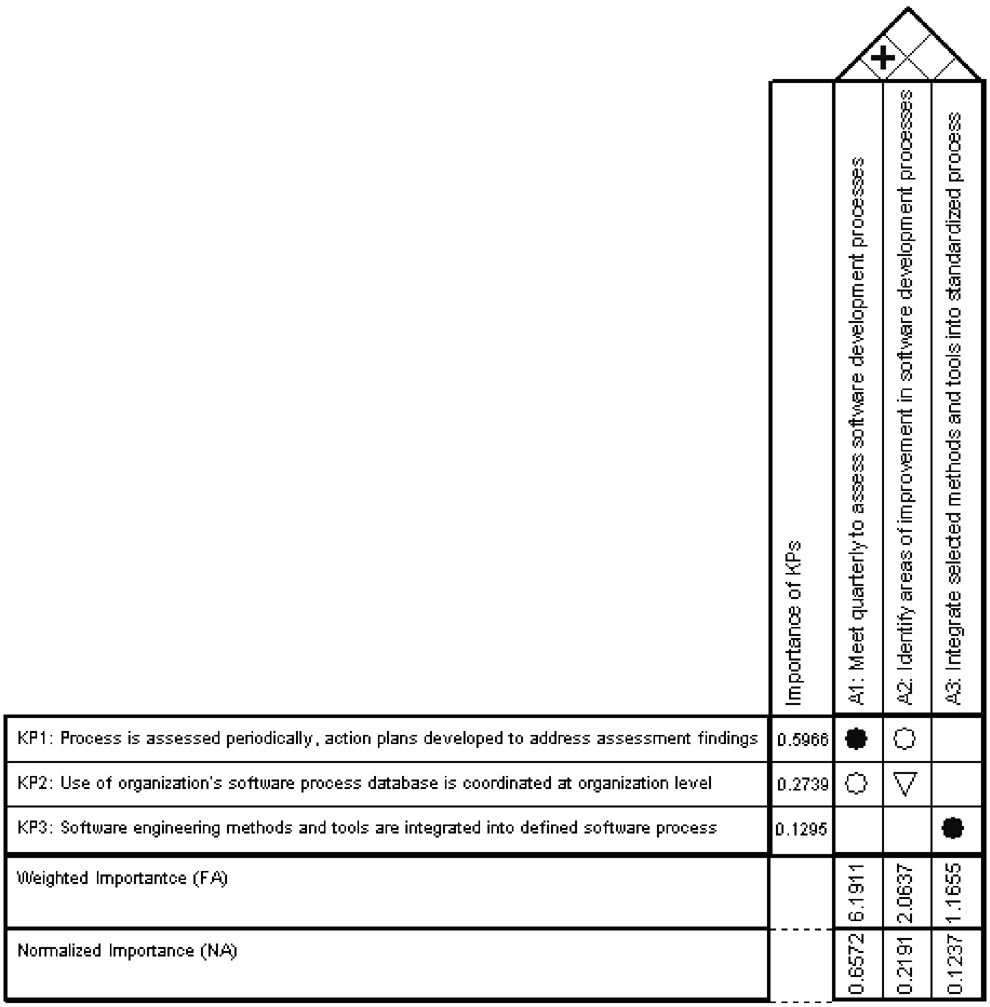


Figure 6.5. Action Plan House of Quality Matrix (AP-HoQ Matrix)

1. Enter the KPs together with their normalized importance values (NP) into the rows of the AP-HoQ matrix.

2. Derive a set of actions from the KPs, and enter them into the columns. These actions represent the steps to be followed to execute the KPs. Various sets of actions are derived separately for different common features.
3. Determine the correlation between each KP and each action. The same set of symbols in Table 5.1 is used in this matrix.
4. Calculate the weighted importance of actions using Equation (12):

$$FA_i = \sum_{j=1}^{z} NKP_j * IR(A_i, KP_j) \tag{12}$$

5. Calculate the normalized importance of actions.
6. 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 a positive correlation between a pair of actions, while a negative sign (-) is used to indicate a negative correlation.

The same three KPs in Figure 6.4 are entered in the rows of Figure 6.5, and three actions derived from these KPs are entered in the columns. The correlation between each KP-action pair is determined and entered in the matrix. Based on these correlations and the normalized importance values of KPs, the weighted importance can be calculated for each action using Equation (12). For instance, the weighted importance of A1 can be calculated as:

$$FA_1 = \sum_{j=1}^{3} NKP_j * IR(A_1, KP_j)$$

$$= 0.5966*9 + 0.2739*3 + 0.1295*0 = 6.1911$$

After all the weighted importance values are calculated, they are normalized to obtain the normalized importance.

7. SPI FRAMEWORK BASED ON CMMI USING QFD

Our SPI framework also works with CMMI, which is gaining popularity in the industry. Again, QFD is used to help with the SPI based on CMMI. The same objectives as mentioned in Section 6.1 still apply. First, business and other requirements within an organization are mapped to CMMI Process Areas and practices. A connection is established so that the organization can clearly see how CMMI helps with its business goals. Second, software process requirements from multiples 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 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 CMMI. For instance, an action (A1) derived using this approach is strongly related to Practice1 in CMMI, while another action (A2) is strongly related to Practice2. Suppose that according to the mapping developed from this framework, it is found that Practices1 reflects the requirements more than Practice2 does. As a result, A1 should have priority over A2. This guarantees that the actions are in accordance with 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 the proposed framework all the way down to the action plans. The requirements from multiple perspectives are correlated with each other using the priority assessment technique introduced in Section 4.1. As a result, the priority value of each requirement is adjusted after the impacts from the other requirements are assessed.

In order to incorporate both the staged model and the continuous model in CMMI, the SPI framework based on CMMI contains two portions: 1) SPI framework for CMMI staged model and 2) SPI framework for CMMI continuous model.

7.1. SPI FRAMEWORK FOR CMMI STAGED MODEL USING QFD

The SPI framework for CMMI staged model, as shown in Figure 7.1, resembles the SPI model based on CMM as introduced in Section 4.2.1.

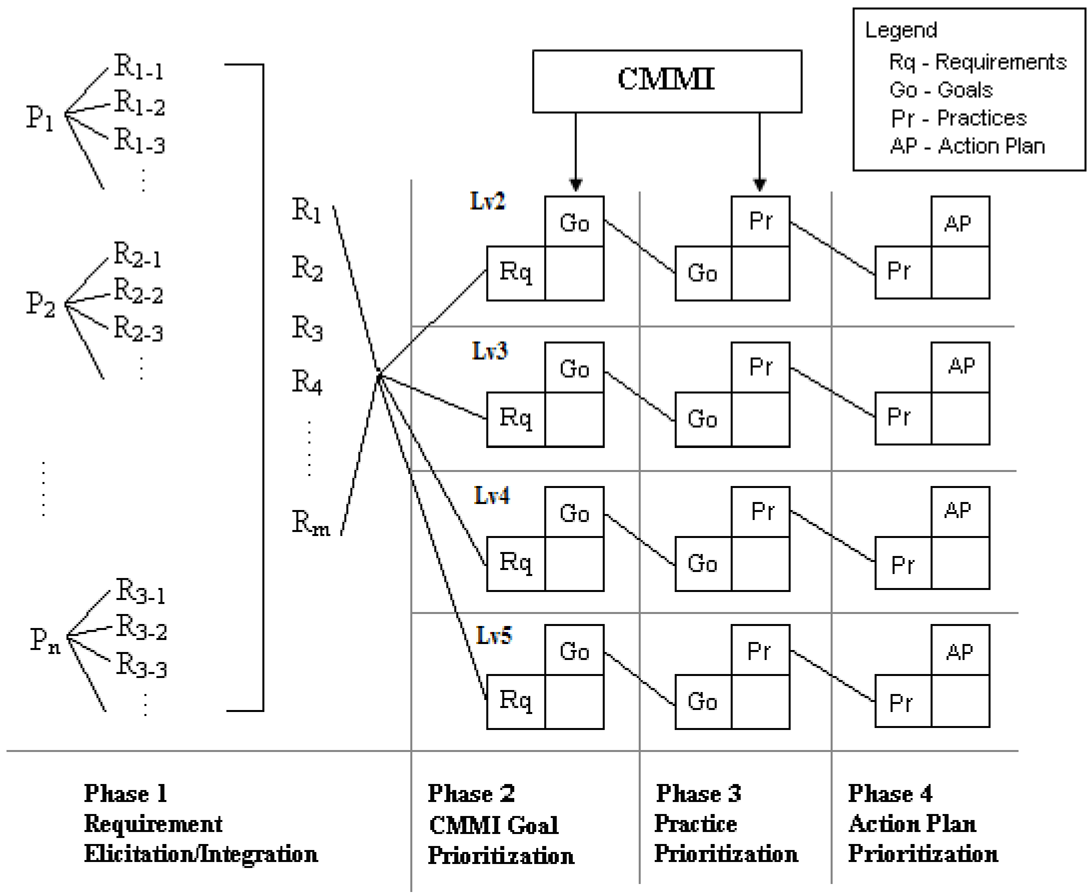


Figure 7.1. Software Process Improvement through CMMI Staged Model Using QFD

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.

In order to achieve these goals, CMMI staged model has generic practices categorized into four common features as well as the specific practices which correspond to the “Activities Performed” common feature in CMM. The priorities of Practices are determined by their correlations with goals. Thus, 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 CMMI staged model and satisfy the process requirements. Those actions with higher importance values help to achieve higher process requirements satisfaction.

Because of the close resemblance between CMMI staged model and CMM, the four phases for the SPI framework based on CMMI staged model as shown in Figure 7.1 are very similar with the SPI framework based on CMM in Section 4.2.1.

In Figure 7.1, phase 1 is exactly the same with the SPI framework based on CMM. 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. In Figure 7.1, 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 the next phase.

The second through fourth phases of this framework are applied to Level 2 to Level 5 of the CMMI staged model. The prioritized and integrated requirements from Phase 1 are linked to all goals in each of the four levels in CMMI staged model using relationship matrices. These prioritized goals are used as the basis for the prioritization of Practices. Finally, the prioritized Practices are transformed into prioritized action plans using House of Quality (HoQ).

In the second phase, which is “CMMI goal prioritization,” the goals of all PAs in a particular maturity level are selected and prioritized based on the requirements from the previous phase. This phase helps to achieve two important objectives. First, the organization needs to comply with the CMMI standard. At the same time, the organization needs to ensure that by reaching a particular maturity level, the process is also satisfying the business and other requirements within the organization. In Phase 2, a relationship matrix is used to establish connections between the requirements from the organization and the goals in CMMI. This matrix demonstrates that complying with the CMMI standard also helps satisfy the business and other requirements in the organization. Second, 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 goals serve as the bridge between requirements and the action plan. By prioritizing the goals, requirements from the organization can be transformed to the Practices in the third phase, and finally to the action plans in the final phase. In this way, a set of actions can be executed not only to achieve a specific maturity level in CMMI, but also to satisfy organizational process requirements.

The third phase 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 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 has been clearly shown in CMMI documentation [12].

In the fourth phase of the framework, which is “action plan development and prioritization,” a set of actions is derived from the prioritized Practices. These actions should reflect the requirements integrated in the first phase. 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.

As shown in the above framework, by incorporating requirements from the organization into action plans through goals and Practices, the connection between the objectives of the organization and CMMI maturity levels becomes clear.

7.2. SPI FRAMEWORK FOR CMMI CONTINUOUS MODEL USING QFD

The SPI framework for CMMI continuous model differs a lot from the staged framework. However, the same techniques of correlation-based prioritization with the help of QFD are used in the framework. In the continuous model of 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 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. Figure 7.2 illustrates how the practices and the actions are prioritized in the SPI framework for CMMI continuous model using QFD. The process requirements are used to in the prioritization of both PAs and Practices. The first step is to calculate the priority values of PAs. Then the Practices are prioritized from both the process requirements and PAs. Depending on which PA a Practice is from, the priority value of that Practices calculated from the requirements is multiplied by the PA priority. Finally, the action priority values are calculated from the Practice priority values.

Thus, as illustrated in Figure 7.2, the PAs are prioritized based on those process requirements and the PAs that help achieve higher overall satisfaction of process requirements get higher importance.

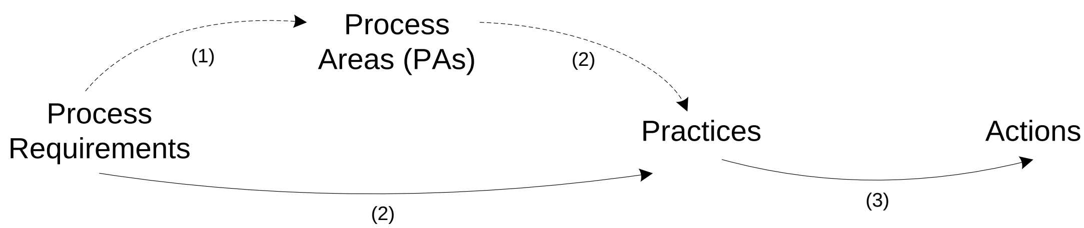


Figure 7.2. Priority Calculation in SPI Framework Based on CMMI Continuous Model Using QFD

In order to make improvements on the PAs, generic practices for the generic goals and specific practices for specific goals at various capability levels are prioritized at the next phase. The priorities of Practices at different capability levels are determined by their correlations with the same set of process requirements. Because in CMMI continuous model, different PAs can have different of capability levels, the prioritization of Practices should be done for individual PAs. Thus, in this framework for CMMI continous model, 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 are used in the calculation of priorities of practices. This will be introduced in more details in Section 7.3.3. In the last phase, 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 CMMI and satisfy the process requirements. Those actions with higher importance values help to achieve higher process requirements satisfaction.

In Figure 7.3, phase 1 is exactly the same with the SPI framework based on CMM. 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

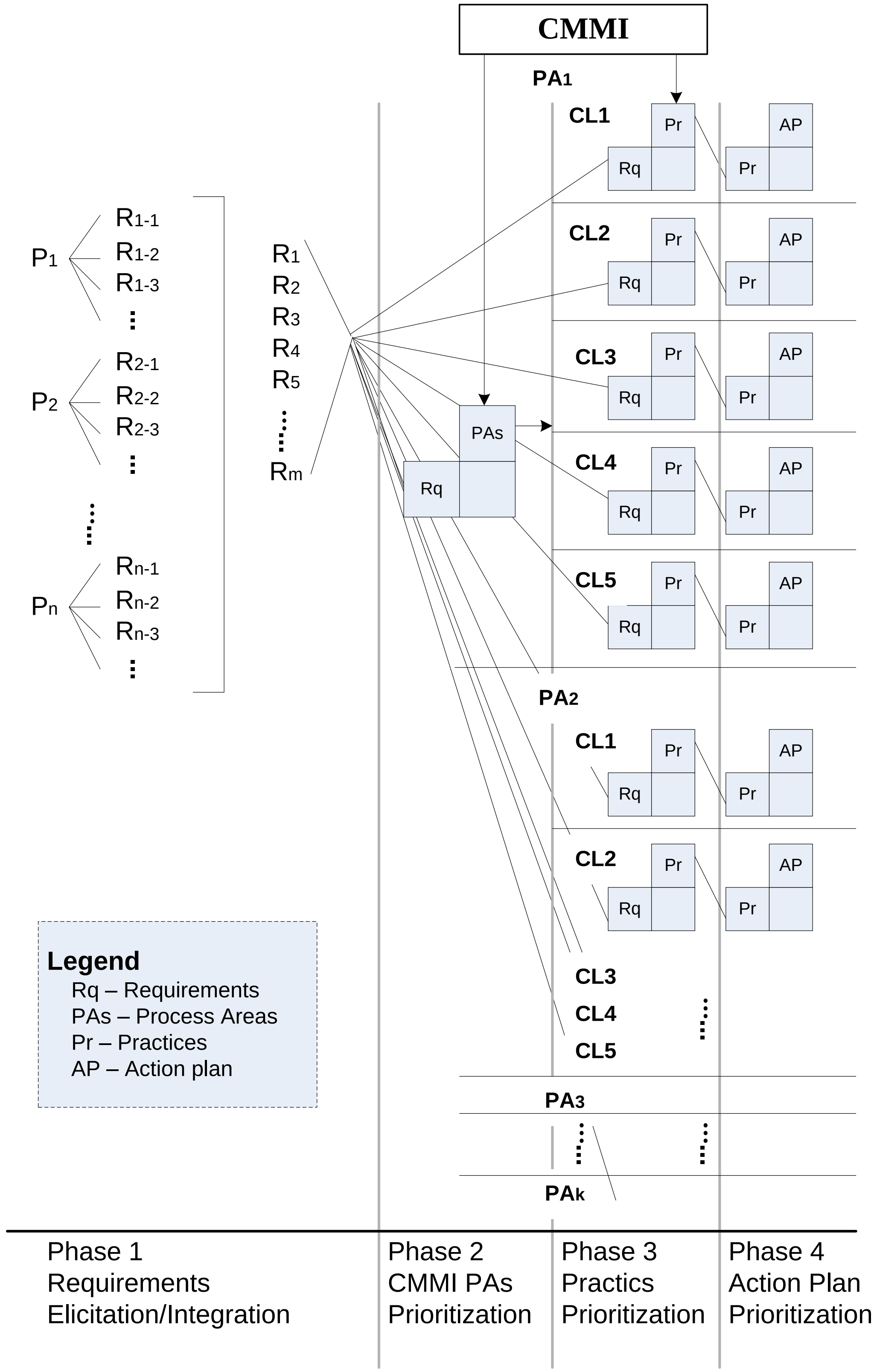


Figure 7.3. Software Process Improvement through CMMI Continuous Model Using QFD

then be prioritized based on their relative importance within the organization and integrated into one single set of requirements.

In Figure 7.3, 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 the next phase.

The second through fourth phases of this framework are applied to the PAs in the CMMI Continuous model. Because in CMMI continuous model, different capability levels are applied to different PAs, the framework for the staged model 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, 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 the second phase, which is “CMMI PA prioritization,” all PAs are selected and prioritized based on the requirement priorities derived from the previous phase. This phase helps achieve two 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 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 the third phase, and finally to the action plans in the final phase. 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.

The third phase of the proposed 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 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.

In the fourth phase 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 should reflect the requirements integrated in the first phase. 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. Thus, more resources should be assigned to those actions with high priorities.

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.

7.3. MATRICES IN FRAMEWORK BASED ON CMMI CONTINUOUS MODEL USING QFD

Due to the similarity between CMMI staged model and CMM, the matrices used in the SPI framework based on CMMI staged model are identical to those used in the framework based on CMM as shown in Section 6.2. In this section, the four different matrices used in the framework based on CMMI continuous model are introduced.

7.3.1. Requirements Impact Matrix (RI Matrix) in Phase 1. The requirements prioritization technique introduced in Section 5.1 can be used in Phase 1 of the framework to integrate requirements from all perspectives into one single set. 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 by following the five steps in Section 5.1.

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 CMM and CMMI, Figure 6.2 can again be used as an example. It uses the relationship matrix as introduced in Figure 5.2. Depending on the number of perspectives, these requirements are integrated by following either Section 5.2 or Section 5.3.

7.3.2. 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 the previous phase. This is a variation of the RG Matrix introduced in Section 6.2.2. 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.

Figure 7.4 shows an example of the RPA Matrix using CMMI continuous model. The following five steps guide through the process of building the Requirements-Goals Impact (RG) Matrix:

1. Enter the integrated requirements (deliverables from the previous phase) along with their adjusted priorities into the rows.

Requirements - PAs

Standard 9-3-1		
Strong	●	9.0
Moderate	○	3.0
Weak	▽	1.0

	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	Requirements Management
R1: Increase profits	0.3517			○					○	
R2: Lead in competition	0.2401	▽								○
R3: Reduce cost of development	0.1846			▽		●			○	○
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	0.6984	0.4909	6.3333	6.8824
Normalized Importance (NPA)		1.7621	2.0289	0.5509	0.0673	1.5431	0.1962	0.1379	1.7796	1.9339

Figure 7.4. Requirements-Process Areas Impact (RPA) Matrix for CMMI Continuous Model

2. Enter all PAs in CMMI of a particular maturity (for staged model)/capability (for continuous model) 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 in Table 5.1 is used in this matrix.
4. Calculate the weighted importance values for PAs (FPA) using the following equation:

$$FPA_i = \sum_{j=1}^{M} AP_j * IR(PA_i, R_j) \tag{13}$$

where: FPA_i is the weighted priority value of PA i,

AP_j is the adjusted priority value of requirement j calculated in the previous phase,

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 Figure 7.4. 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 Equation (13). For instance, the weighted importance value of the PA "Project Planning" is calculated as follows:

$$FPA_1 = \sum_{j=1}^{18} AP_j * IR(PA_1, R_j)$$

$$= 0.2401*1 + 0.0740*3 + 0.0556*1 + 0.2673*9 + 0.1935*9 + 0.1493*1$$
$$+ 0.0559*9 + 0.0418*3 + 0.0233*3 + 0.0585*9 + 0.0373*3 + 0.0133*9$$
$$= 6.2707$$

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 the next phase.

7.3.3. Requirements-Practices Impact Matrix (RPr Matrix) in Phase 3. In the third phase 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.

For the CMMI continuous model, RPr Matrix can be constructed in a way similar to the steps mentioned for GP Matrix as introduced in Section 6.2.3. The Practices from the target capability level of a PA 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 GP matrix based on CMMI continuous model is shown in Figure 7.5.

The following six steps are followed in order to develop a GP matrix:

1. Enter the same set of requirements used in RPA Matrix along with their adjusted priorities into the rows.
2. List the Practices in 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.
4. Calculate the weighted importance values (WP) of the Practices using Equation (14):

$$F\,\mathrm{Pr}_i = \sum_{j=1}^{s} AP_j * IR(\mathrm{Pr}_i, R_j) \tag{14}$$

where: FPr_i is the weighted priority value of Practice I,

AP_j is the adjusted priority value of requirement j calculated in the previous phase,

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

Standard 9-3-1	
Strong ●	9.0
Moderate ○	3.0
Weak ▽	1.0

	Requirements Importance	SP1.1 Estimate the Scope of the Project	SP1.2 Establish Estimates of Work Prod. and Task Attr.	SP1.3 Define Project Life Cycle	SP1.4 Determine Estimates of Effort and Cost	SP2.1 Establish the Budget and Schedule	SP2.2 Identify Project Risks	SP2.3 Plan for Date Management	SP2.4 Plan for Project Resources	SP2.5 Plan for Needed Knowledge and Skills	SP2.6 Plan Stakeholder Involvement	SP2.7 Establish the Project Plan	SP3.1 ReviewPlans that Affect the Project	SP3.2 Reconcile Work and Resource Levels	SP3.3 Obtain Plan Commitment	GP1.1 Perform Base Practices
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 (FPr)		4.4789	1.9690	1.6189	3.3261	9.3081	2.6042	0.6745	1.7296	1.2979	2.4802	2.7692	0.9433	0.6331	0.5258	17.1531
Normalized Importance (NPr)		0.0869	0.0382	0.0314	0.0646	0.1807	0.0506	0.0131	0.0336	0.0252	0.0481	0.0538	0.0183	0.0123	0.0102	0.3330
Global Importance (GPr)		0.1532	0.0674	0.0554	0.1138	0.3184	0.0891	0.0231	0.0592	0.0444	0.0848	0.0947	0.0323	0.0217	0.0180	0.5868

Figure 7.5. Requirements-Practices Impact (RPr) Matrix for PA “Project Planning” in CMMI Continuous Model

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

The same eighteen requirements in Figure 7.4 are entered in the rows of Figure 7.5. The corresponding capability level 1 Practices from the first PA in Figure 7.4, which is “Project Planning”, 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 Equation (14). For instance, the weighted importance for the Specific Practice (SP1.3) is calculated as follows:

$$F\,\mathrm{Pr}_{1.3} = \sum_{j=1}^{18} AP_j * IR(\mathrm{Pr}_{1.3}, R_j)$$

$$= 0.0740*9 + 0.1935*3 + 0.0418*3 + 0.0233*9 + 0.0373*1 = 1.6189$$

After all the weighted importance values are calculated, the weighted priority (FPr) values are normalized to obtain the normalized importance (NPr) and these NPr’s are multiplied by 1.7621, which is the Normalized Importance value for the PA “Project Planning” in Figure 7.4, to obtain the global importance values (GPr).

7.3.4. 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.

An example of the AP-HoQ matrix based on CMMI continuous model is shown in Figure 7.6. 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 in Table 5.1 is used in this matrix.
4. Calculate the weighted importance of actions using Equation (15):

$$FA_i = \sum_{j=1}^{z} G\,\mathrm{Pr}_j * IR(A_i, \mathrm{Pr}_j) \qquad (15)$$

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

The same set of Practices from capability level 1 in PA “Project Planning” in Figure 7.5 are entered in the rows of Figure 7.6, 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 Equation (15). For instance, the weighted importance of A1 can be calculated as:

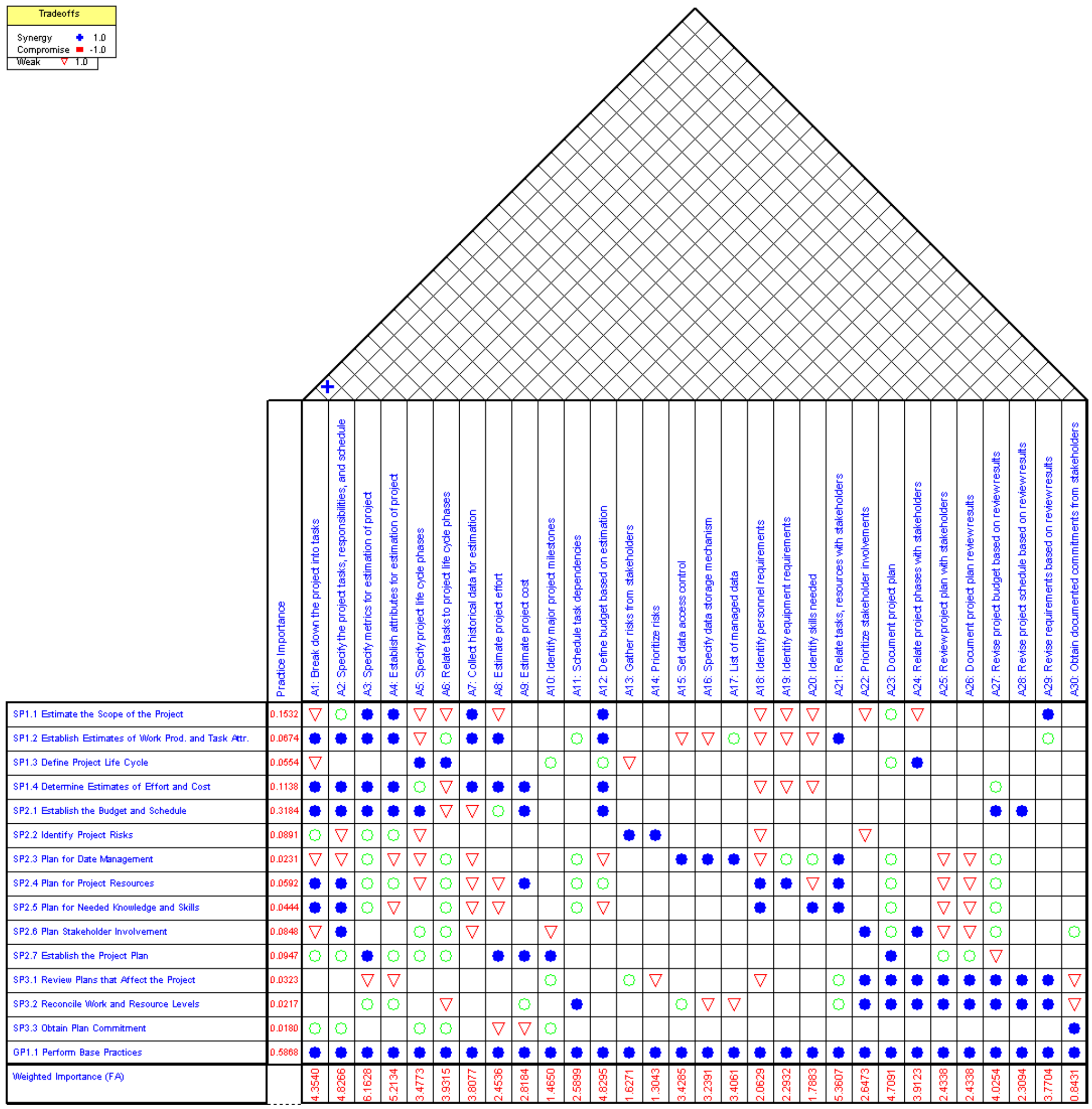


Figure 7.6. Action Plan House of Quality Matrix (AP-HoQ Matrix) for PA “Project Planning” in CMMI Continuous Model

$$FA_1 = \sum_{j=1}^{3} NKP_j * IR(A_1, KP_j)$$

= 0.1532*1 + 0.0674*9 + 0.0554*1 + 0.1138*9 + 0.3184*9 + 0.0891*3 + 0.0231*1 + 0.0592*9 + 0.0444*9 + 0.0848*1 + 0.0947*3 + 0.0180*3 + 0.5868*9 = 4.3540

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.

The same calculations are applied to another PA, "Requirements Management." This PA deals with the management of product requirements in the software development process to be improved. The same set of process requirements in Figure 7.4 – Figure 7.6 are used. These requirements has been integrated and prioritized in Phase 1 of the framework and the PAs are prioritized based on their correlations with these requirements (shown in Figure 7.4).

If "Requirements Management" PA is also aiming to reach capability level 1, Figure 7.7 shows the RPr Matrix between the set of prioritized process requirements and all Practices in "Requirements Management" PA. 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 as shown in Table 5.1 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 Equation (14).

For instance, in Figure 7.7, the Weighted Importance value for the Practice SP1.5 is calculated as:

$$F\,\mathrm{Pr}_{1.5} = \sum_{j=1}^{18} AP_j * IR(\mathrm{Pr}_{1.5}, R_j)$$

= 0.0401*3 + 0.1493*9 + 0.0418*9 + 0.0233*9 + 0.0585*9 = 3.2367

Standard 9-3-1

Strong ● 9.0
Moderate ○ 3.0
Weak ▽ 1.0

	Requirements Importance	SP1.1 Obtain an understanding of requirements	SP1.2 Obtain commitment to requirements	SP1.3 Manage reuqirement changes	SP1.4 maintain bidirectional traceability of requirements	SP1.5 Identify inconsistencies between project work and requirements	GP1.1 Perform Base Practices
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 (FPr)		5.8528	2.0853	3.8921	4.2461	3.2367	6.8824
Normalized Importance (NPr)		0.2234	0.0796	0.1486	0.1621	0.1236	0.2627
Global Importance (GPr)		0.4321	0.1539	0.2873	0.3135	0.2390	0.5081

Figure 7.7. Requirements-Practices Impact (RPr) Matrix for PA "Requirements Management" in CMMI Continuous Model

After all Weighted Importance values in the matrix are calculated, they are normalized and then multiplied by the Normalized Importance value of "Requirements Management" PA, which is 1.9339, to obtain the final Global Importance values. The Global Importance value of SP1.5 in this example is 0.2390.

Following the prioritization of the Practices in "Requirements Management" PA, actions are developed from the Practices and they are prioritized using the AP-HoQ as shown in Figure 7.8. The Practices in "Requirements Management" PA and their Global Importance values from Figure 7.7 are entered in the rows of Figure 7.8. The derived actions entered in the columns. The correlation between each Practice-action pair is determined based on the 9-3-1 values from Table 5.1 and entered in the correlation section in Figure 7.8. Based on the importance values of Practices and the correlations, the Weighted Importance values of all actions are calculated using Equation (15).

For instance, the Weighted Importance of A2 is calculated as:

$$WA_2 = \sum_{j=1}^{6} NKP_j * IR(A_2, KP_j)$$

$$= 0.1539*3 + 0.2873*1 + 0.5081*9$$

$$= 5.3219$$

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 the whole SPI project can reach a higher level of process requirements satisfaction.

When there are more than one PA in an SPI project, actions can be prioritized across PAs. In the examples above, with the actions prioritization individually for the two PAs, Project Planning and Requirements Management, they can be put together and prioritized. Table 7.1 below shows the ranked list of actions from these two PAs.

Standard 9-3-1		
Strong	●	9.0
Moderate	○	3.0
Weak	▽	1.0

	Practice Importance	A1: Set requirements documentation guidelines	A2: Collect requirements from stakeholders	A3: Assess impact of requirements on existing commitments	A4: Document changed commitments	A5: Record requirement changes	A6: Assess impact from requirement changes	A7: Generate requirement traceability matrix
SP1.1 Obtain an understanding of requirements	0.4321	●		○		○	●	▽
SP1.2 Obtain commitment to requirements	0.1539		○	●	●			▽
SP1.3 Manage reuqirement changes	0.2873		▽	○	●	●	●	▽
SP1.4 maintain bidirectional traceability of requirements	0.3135						○	●
SP1.5 Identify inconsistencies between project work and requirements	0.2390	▽						●
GP1.1 Perform Base Practices	0.5081	●	●	●	●	●	●	●
Weighted Importance (FA)		8.7008	5.3219	8.1162	8.5437	8.4549	11.9880	10.4187

Figure 7.8. Action Plan House of Quality Matrix (AP-HoQ Matrix) for PA “Requirements Management” in CMMI Continuous Model

Table 7.1. Prioritized Actions from “Project Planning” and “Requirements Management”

Actions	**Weighted Importance**
A6: Assess impact from requirement changes	11.988
A7: Generate requirement traceability matrix	10.4187
A1: Set requirements documentation guidelines	8.7008
A4: Document changed commitments	8.5437
A5: Record requirement changes	8.4549
A3: Assess impact of requirements on existing commitments	8.1162
A3: Specify metrics for estimation of project	6.1628
A21: Relate tasks, resources with stakeholders	5.3607
A2: Collect requirements from stakeholders	5.3219
A4: Establish attributes for estimation of project	5.2134
A12: Define budget based on estimation	4.8295
A2: Specify the project tasks, responsibilities, and schedule	4.8266
A23: Document project plan	4.7091
A1: Break down the project into tasks	4.354
A27: Revise project budget based on review results	4.0254
A6: Relate tasks to project life cycle phases	3.9315
A24: Relate project phases with stakeholders	3.9123
A7: Collect historical data for estimation	3.8077
A29: Revise requirements based on review results	3.7704
A5: Specify project life cycle phases	3.4773
A15: Set data access control	3.4285
A17: List of managed data	3.4061
A16: Specify data storage mechanism	3.2391
A9: Estimate project cost	2.8184
A22: Prioritize stakeholder involvements	2.6473
A11: Schedule task dependencies	2.5899
A8: Estimate project effort	2.4536

Table 7.1. Prioritized Actions from "Project Planning" and "Requirements Management" (cont.)

A26: Document project plan review results	2.4338
A25: Review project plan with stakeholders	2.4338
A28: Revise project schedule based on review results	2.3094
A19: Identify equipment requirements	2.2932
A18: Identify personnel requirements	2.0629
A20: Identify skills needed	1.7883
A13: Gather risks from stakeholders	1.6271
A10: Identify major project milestones	1.465
A14: Prioritize risks	1.3043
A30: Obtain documented commitments from stakeholders	0.8431

8. APPLICATION EXAMPLES

In this section, application examples are used to illustrate the framework for SPI using CMM as introduced in Section 6 and the framework for SPI using CMMI continuous model as introduced in Section 7. These examples use the same set of requirements obtained and prioritized below.

8.1. REQUIREMENTS PRIORITIZATION

A software development organization is considering improve its software process. Three perspectives of requirements are collected from various levels and branches of the organization.

Perspective 1: Business Requirements

These requirements are from the executive level of the business. They primarily deal with the large scope objectives from the organization point of view. These requirements include:

- Increase profit
- Lead in competition
- Reduce cost of development
- Reduce time to develop
- Reduce marketing time
- Improve quality

Perspective 2: Management Requirements

These requirements are from managers of each software development department. They primarily deal with the objectives toward the production of software. Their scope is smaller than that of the business requirements. These requirements include:

- Within budget
- On schedule
- High customer satisfaction
- Increase productivity

- Manage project aggressively
- High conformance to software engineering standard

Perspective 3: Quality Requirements

These requirements are from either the quality assurance team if there is one in the organization or from quality specialists integrated into development teams. They primarily deal with the quality issues of software products. These requirements include:

- Low failure rate
- Low defect rate
- High reliability
- High requirement satisfaction
- High maintainability
- High usability

With these three perspectives of requirements available, local priorities and perspective priorities were assigned to them. The initial global priorities are then calculated based on the local priorities and perspective priorities. In Table 8.1, symbol "P" represents perspective, and symbol "R" represents requirement. For example, P_1 represents the first perspective which is the business perspective and R_2^1 represents the second requirement in the first perspective, which is the "lead in competition" requirement. Local and perspective priorities are given in the column next to the perspective names and requirement names. The calculated initial global priorities are listed in the right-most column.

For example, LW_1^1, which is the local priority of requirement R_1^1, is 7; the perspective priority PW_1 for perspective P_1, which is business perspective, is 3. Therefore, the global priority W_1^1 for requirement R_1^1 is calculated using Equation (5) as following:

$$W_1^1 = PW_1 * LW_1^1 = 7 * 3 = 21$$

The raw initial global priority of 21 is shown in Table 8.1. After normalization it becomes 0.1615.

The same calculation is applied to other requirements. The local priority LW_3^2 of requirement R_3^2 has a value of 4. The perspective priority PW_2 for P_2 is 2. Therefore, the initial global priority W_3^2 for requirement R_3^2 is calculated as:

$$W_3^2 = PW_2 * LW_3^2 = 4 * 2 = 8$$

After normalization the raw initial global priority becomes 0.0615. In this example, all raw initial priorities are set to four decimal places precision.

Table 8.1. Calculation of Initial Priorities

Perspectives and Requirements	**Reqt Local Priority (LW)**	**Persp Priority (PW)**	**W=LW*PW**	**Raw Initial Global Priority**
P_1: Business Requirements		3		
R_1^1: Increase profit	7		21	0.1615
R_2^1: Lead in competition	5		15	0.1154
R_3^1: Reduce cost of development	5		15	0.1154
R_4^1: Reduce time to develop	2		6	0.0462
R_5^1: Reduce marketing time	2		6	0.0462
R_6^1: Improve quality	1		3	0.0231
P_2: Management Requirements		2		

Table 8.1 Calculation of Initial Priorities (cont.)

R_1^2: Within budget	7		14	0.1077
R_2^2: On schedule	6		12	0.0923
R_3^2: High customer satisfaction	4		8	0.0615
R_4^2: Increase productivity	2		4	0.0308
R_5^2: Manage project aggressively	2		4	0.0308
R_6^2: High conformance to software engineering standard	1		2	0.0154
P_3: Quality Requirements		1		
R_1^3: Low failure rate	6		6	0.0462
R_2^3: Low defect rate	4		4	0.0308
R_3^3: High reliability	4		4	0.0308
R_4^3: High requirement satisfaction	3		3	0.0231
R_5^3: High maintainability	2		2	0.0154
R_6^3: High usability	1		1	0.0077

8.2. APPLICATION EXAMPLE OF SPI BASED ON CMM USING QFD

CMM Level 2

Based on the raw initial global priorities of the 18 requirements as shown in Table 8.1, these requirements can be integrated in stage 1 of the framework using RI Matrices as introduced in Section 6.2.1 and the method to integrate more than 2 perspectives as introduced in Section 5.3.

Considering that there are three perspectives, 3 RI Matrices are needed to capture the correlations between all pairs of requirements from different perspectives. Figure 6.2 has shown the RI Matrix between the Business Perspective and the Quality Perspective.

Figure 8.1 below illustrates the RI Matrix between the Business Perspective and the Management Perspective.

	Initial Global Priorities	Within Budget	On schedule	High customer satisfaction	Increase productivity	Manage project aggressive	High conformance 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 8.1. Requirements Impact (RI) Matrix between Business and Management Perspectives

Similarly, Figure 8.2 shows the RI Matrix between the Management Perspective and the Quality Perspective.

	Initial Global Priorities	Low failure rate	Low defect rate	High reliability	High requirement satisfaction	High maintainability	High usability	Weighted Priorities (P2-3)
Initial Global Priorities		0.0462	0.0308	0.0308	0.0231	0.0154	0.0077	
Within budget	0.1077	○	○	▽		●	▽	0.0440
On schedule	0.0923		●			○		0.0298
High customer satisfaction	0.0615	●		○	○		●	0.0398
Increase productivity	0.0308				○	○		0.0036
Manage project aggressively	0.0308		●	○	○	▽		0.0140
High conformance to software engineering standard	0.0154	●	●	○	○	●	○	0.0157
Weighted Priorities (P3-2)		0.0469	0.0483	0.0133	0.0096	0.0232	0.0054	

Figure 8.2. Requirements Impact (RI) Matrix between Management and Quality Perspectives

After all three RI Matrices are completed, Equation (9) is used to calculated the final priority of each of the 18 process requirements. For instance, the final priority FP_1^1 of the requirement of "increase profit" is calculated as:

$$FP_1^1 = WP_1^2 + WP_1^3 = 0.3404 + 0.1119 = 0.4523$$

The same equation is applied to all requirements to calculated final priorities. These final priorities are then normalized using Equation (7) before they are adjusted by Equation (8) using their initial global priorities and the α value as introduced in Section 5.2. In this application, the correlations is considered as important as the initial global priorities. Thus, an α value of 1 is used in calculating the adjusted priorities of these process requirements.

For example, the final priority value of the process requirement of “increase profit”, which is 0.4523 as shown above, is normalized to 0.1902. This normalized value is adjusted using Equation (8) as follows:

Adjusted Priority = Initial global priority + α * Normalized priority
= 0.1615 + 1 * 0.1902 = 0.3517

In the above calculation, 0.1615 is the raw initial global priority as shown in Table 8.1. The adjusted priority of 0.3517 reflects both the perspective local weight and the requirement local weight. In addition, it reflects the correlation between this requirement and all other requirements from different perspectives. This value will be used in Stage 2 of the SPI framework, which is CMM Goal Prioritization.

There are six key process areas in Level 2 of CMM. As an example, all seventeen goals from level 2 KPAs are selected. The eighteen requirements obtained from requirements integration phase are mapped to the twenty goals as shown in Figure 8.3. Impact of each requirement on each goal is determined and represented using 9-3-1 standard values in Table 5.1. Entries are left blank where there is no impact. Weighted importance (FG) of all goals is determined. Then they are normalized to maintain consistency and prevent loss of precision.

	Requirements Importance	Requirement Management (KPA 1)	G1: Requirements controlled to establish baseline	G2: Software plans, products, activities are kept consistent	Software project planning (KPA 2)	G3: Software estimates are documented	G4: Software projects activities are planned and documented	G5: Affected groups agree to commitment	Software Project Tracking and Oversight (KPA 3)	G6: Actual results and performances are tracked against plans	G7: Corrective actions are taken and managed to closure when deviation occurs	G8: Changes to software commitments are agreed to by affected groups and individuals	Software subcontract management (KPA 4)	G9: Prime contractor selects qualified software subcontractors	G10: Prime contractors and subcontractor agree to their commitments	G11: Prime contractor and subcontractor maintains ongoing communications	G12: Prime contractor tracks subcontractor's results and performance against commitments	Software Quality Assurance (KPA 5)	G13: SQA activities are planned	G14: Adherence of software products and activities to standards, procedures, requirements is verified	G15: Affected groups and individuals are informed of SQA activies and results	G16: Noncompliance issues that cannot be resolved withing software project are addressed by senior management	Software Configuration Management (KPA 6)	G17: SCM activities are planned	G18: Selected software work products are identified, controlled and available	G19: Changes to identified software work products are controlled	G20: Affected groups and individuals are informed of the status and content of software baselines
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	0.1936			●		▽	●	▽		●	●					○	●								▽		
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	0.0996		○	▽															○					▽			
R14: Low defect rate	0.0646																		○					▽			
R15: High reliability	0.0498																		○					●		○	
R16: High requirement satisfaction	0.0581										▽				○	○	▽		▽	●							
R17: High maintainability	0.0373																		▽					●		○	
R18: High usability	0.0138																		▽								
Weighted Importance (FG)			2.2731	6.3341		3.8063	2.3196	1.4415		2.5779	2.6360	0.6134		1.1596	0.7877	1.5792	2.5162		1.3522	1.3008	0.1254	0.3762		1.5651	0.5698	0.7868	0.1813
Normalized Importance (NG)			0.1405	0.3916		0.2353	0.1434	0.0891		0.1594	0.1630	0.0379		0.0717	0.0487	0.0976	0.1556		0.0836	0.0804	0.0078	0.0233		0.0968	0.0352	0.0486	0.0112

Figure 8.3. Goal Prioritization Using RG Matrix (Level 2)

For example, goal G_1 (requirements controlled to establish baseline) of KPA 1 (Requirements Management) is mapped against all 18 requirements and the relationships are represented using 9-3-1 symbol sets. In this case, requirement R_5 (reduce marketing time) and requirement R_6 (improve quality) have strong impacts on goal G_1. Requirement

R_2 (lead in competition), requirement R_{11} (manage software development aggressively), and requirement R_{13} (low failure rate) have moderate impacts on goal G_1. Similarly, requirement R_7 (within budget) has a weak impact on goal G_1.

Hence, weighted priority FG_1 of goal G_1 is calculated using Equation (10) as following:

$$FG_1 = \sum_{j=1}^{18} AP_j * IR(G_1, R_j)$$

$$= AP_2 * IR(G_1, R_2) + AP_5 * IR(G_1, R_5) + AP_6 * IR(G_1, R_6)$$

$$+ AP_7 * IR(G_1, R_7) + AP_{11} * IR(G_1, R_{11}) + AP_{13} * IR(G_1, R_{13})$$

$$= 0.2401*3 + 0.0556*9 + 0.0401*9 + 0.2673*1 + 0.0418*3 + 0.0996*3 = 2.2731$$

The adjusted priorities AP_2, AP_5, AP_6, AP_7, AP_{11}, and AP_{13} for requirements R_2, R_5, R_6, R_7, R_{11}, and R_{13} are *0.2401*, *0.0556*, *0.0401*, *0.2673*, *0.0418*, and *0.0996*, respectively; 9, 3, and 1 are the correlation values corresponding to the symbols in the matrix. The requirements with no impact on goal G_1 are not included in the calculation.

After the goals are prioritized, key practices are also priorities using GP matrix in Figure 8.4. All twenty goals used in RG matrix are used in this example. Also, as an example, “Activities Performed” is considered as the representative common feature of key practices in CMM Level 2 KPAs. Eight activities from level 2 KPAs are selected. The twenty CMM goals, prioritized using RG matrix, are mapped to these nine activities. Impact of each goal on each activity is determined and represented using 9-3-1 standard values. Entries are left blank where there is no impact. Weighted importance (FKP) of all eight activities is determined. Then they are normalized to maintain consistency and prevent loss of precision.

For example, activity KP_1 (SE group uses requirements as the basis for plans, work products, activities) is mapped onto 20 goals. This activity has a strong correlation with goal G_1 and G_2, and weak correlations with goal G_3, G_4, G_5, and G_6. It does not have correlation with the other goals. The weighted priority FKP_1 for activity KP_1 is calculated using Equation (11) as following:

	Goals Importance	KP1: SE group uses requirements as the basis for plans, work products, activities	KP2: changes to the allocated requirements are reviewed and incorporated into project	KP3: sw project planning is initiated in the early stages of overall project planning	KP4: SE group participates with other affected grps throughout the project's life	KP5: Project's schedule is tracked and corrective actions are taken	KP6: Peirodic technical reviews and interchanges are held with subcontractor	KP7: SQA group's activities are performed in accordance with SQA plan	KP8: Changes to baselines are controlled according to a documented procedure
Requirement Management (KPA 1)									
G1: Requirements controlled to establish baseline	0.1405	●	○	▽					○
G2: Software plans, products, activities are kept consistent	0.3916	●	●	○	▽				
Software project planning (KPA 2)									
G3: Software estimates are documented	0.2353	▽							
G4: Software projects activities are planned and documented	0.1434	▽	▽	●					
G5: Affected groups agree to commitment	0.0891	▽			●				
Software Project Tracking and Oversight (KPA 3)									
G6: Actual results and performances are tracked against plans	0.1594	▽				●			
G7: Corrective actions are taken and managed to closure when deviation occurs	0.1630					●			○
G8: Changes to software commitments are agreed to by affected groups and individuals	0.0379		▽		○	▽			
Software subcontract management (KPA 4)									
G9: Prime contractor selects qualified software subcontractors	0.0717								
G10: Prime contractors and subcontractor agree to their commitments	0.0487					▽			
G11: Prime contractor and subcontractor maintains ongoing communications	0.0976						●		
G12: Prime contractor tracks subcontractor's results and performance against commitments	0.1556						○		
Software Quality Assurance (KPA 5)									
G13: SQA activities are planned	0.0836			○				▽	
G14: Adherence of software products and activities to standards, procedures, requirements is verified	0.0804							●	
G15: Affected groups and individuals are informed of SQA activies and results	0.0078				●				
G16: Noncompliance issues that cannot be resolved withing software project are addressed by senior management	0.0233								
Software Configuration Management (KPA 6)									
G17: SCM activities are planned	0.0968			○					
G18: Selected software work products are identified, controlled and available	0.0352								▽
G19: Changes to identified software work products are controlled	0.0486								●
G20: Affected groups and individuals are informed of the status and content of software baselines	0.0112				○				▽
Weighted Importance (FKP)		5.4161	4.1272	3.1471	1.4110	2.9882	1.3452	0.8072	1.3943
Normalized Importance (NKP)		0.2625	0.2000	0.1525	0.0684	0.1448	0.0652	0.0391	0.0676

Figure 8.4. Activity Prioritization Using GP Matrix (Level 2)

$$FKP_1 = \sum_{j=1}^{20} NG_j * IR(KP_1, G_j)$$

$= NG_1 * IR(KP_1, G_1) + NG_2 * IR(KP_1, G_2) + NG_3 * IR(KP_1, G_3) + NG_4 * IR(KP_1, G_4) + NG_5 * IR(KP_1, G_5) + NG_6 * IR(KP_1, G_6)$

$= 0.1405*9 + 0.3916*9 + 0.2353*1 + 0.1434*1 + 0.0891*1 + 0.1594*1 = 5.4161$

The normalized priorities NG_1, NG_2, NG_3, NG_4, NG_5, and NG_6 for goals G_1, G_2, G_3, G_4, G_5, and G_6 are 0.1405, 0.3916, 0.2353, 0.1434, 0.0891, and 0.1594, respectively; 9 and 1 are the correlation values corresponding to the symbols in the matrix. The goals with no correlation with activity KP_1 are not included in the calculation. All FKPs are calculated similarly using Equation (11). They are then normalized and the normalized priorities are used as inputs for the prioritization of actions in the next phase of the framework.

The eight CMM activities prioritized from GP matrix are mapped to fourteen actions as shown in Figure 8.5. Impact between each activity and action is determined and represented using 9-3-1 standard values. Weighted and normalized importance of all fourteen actions is determined along with roof values.

For example, action A_1 (hold meetings with stakeholders for requirements review) has moderate correlations with activities KP_2 and KP_8, and a weak correlation with activity KP_4.

The weighted priority FA_1 of action A_1 is calculated using Equation (12) as following:

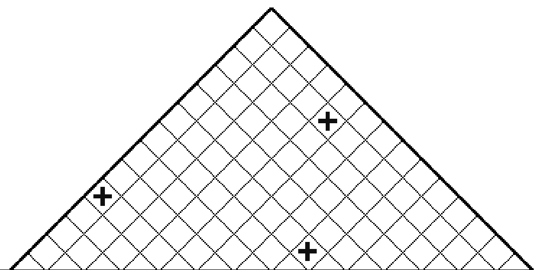
$$FA_1 = \sum_{j=1}^{8} NKP_j * IR(A_1, KP_j)$$

$= NKP_2 * IR(A_1, KP_2) + NKP_4 * IR(A_1, KP_4) + NKP_8 * IR(A_1, KP_8)$

$= 0.2000*3 + 0.0684*1 + 0.0676*3 = 0.8712$

The normalized priorities NKP_2, NKP_4, and NKP_8 for activities KP_2, KP_4, and KP_8 are *0.2000, 0.0684*, and *0.0676*, respectively; 3 and 1 are the correlation values corresponding to the symbols in the matrix. The activities with no correlation with action

A_1 are not included in the calculation. After all weighted priorities are calculated, they are normalized into NA.



	Importance of Activities	A1: Hold meetings with stakeholders for requirements review	A2: Develop project tasks plan	A3: Develp project schedule	A4: Assign tasks to different teams based on plan	A5: Hold review meeting at milestones	A6: Check progress against schedule	A7: Identify tasks behind schedule and find solutions	A8: hold meetings with subcontractor on a regular base	A9: maintain ongoing communication with subcontractors	A10: Develop SQA plan	A11: Assess and improve software development quality	A12: Check SQA activity results against plan	A13: Evaluate importance of changes to be made	A14: Generate sequence of approved changes
KP1: SE group uses requirements as the basis for plans, work products, activities	0.2625		●	●	▽	○					○				
KP2: changes to the allocated requirements are reviewed and incorporated into project	0.2000	○												●	○
KP3: sw project planning is initiated in the early stages of overall project planning	0.1525		▽	○			▽								
KP4: SE group participates with other affected grps throughout the project's life	0.0684	▽			▽				○	○		▽		○	
KP5: Project's schedule is tracked and corrective actions are taken	0.1448		▽	●		○	●	●	○	○			○		○
KP6: Peirodic technical reviews and interchanges are held with subcontractor	0.0652					●			●	●				○	
KP7: SQA group's activities are performed in accordance with SQA plan	0.0391		○			○					○	●	●		
KP8: Changes to baselines are controlled according to a documented procedure	0.0676	○				▽						▽		●	●
Weighted Importantce (FA)		0.8712	2.7771	4.1232	0.3309	1.9936	1.4557	1.3032	1.2264	1.2264	0.9048	0.4879	0.7863	2.8092	1.6428
Normalized Importance (NA)		0.0397	0.1266	0.1879	0.0151	0.0909	0.0664	0.0594	0.0559	0.0559	0.0412	0.0222	0.0358	0.1280	0.0749

Figure 8.5. Action Plan Development Using AP-HoQ Matrix (Level 2)

CMM Level 3

The same 18 process requirements are used here in the application example for CMM Level 3. Figure 6.2, Figure 8.1, and Figure 8.2 shows the RI Matrices. Equations (9), (7), and (8) are used to combine the final priorities of the process requirements, normalize them, and calculate the adjusted priorities. These adjusted priorities become the input of the CMM Goal Prioritization phase.

There are seven key process areas in level 3 of CMM. As an example, all seventeen goals from level 3 KPAs are selected. In Figure 8.6, the eighteen requirements obtained from requirements integration phase are mapped to the seventeen goals. Impact of each requirement on each goal is determined and represented using 9-3-1 standard values. Entries are left blank where there is no impact. Weighted importance of all goals is determined. Then they are normalized to maintain consistency and prevent loss of precision.

For example, goal G_2 (strengths and weaknesses of software process are identified) is mapped against all 18 requirements and the relationships are represented using 9-3-1 symbol sets. In this case, goal G_2 has strong correlation with requirement R_2 (lead in competition) and requirement R_6 (improve quality), moderate correlations with requirement R_1 (increase profit), requirement R_8 (on schedule), requirement R_{10} (increase productivity), and requirement R_{12} (high conformance to software engineering standard). The weighted priority FG_2 of goal G_2 is calculated using Equation (10) as following:

$$FG_2 = \sum_{j=1}^{18} AP_j * IR(G_2, R_j)$$

$$= AP_1 * IR(G_2, R_1) + AP_2 * IR(G_2, R_2) + AP_6 * IR(G_2, R_6)$$

$$+ AP_8 * IR(G_2, R_8) + AP_{10} * IR(G_2, R_{10}) + AP_{12} * IR(G_2, R_{12})$$

$$= 0.3517*3 + 0.2401*9 + 0.0401*9 + 0.1936*3 + 0.0559*3 + 0.0233*3 = 4.3953$$

The adjusted priorities AP_1, AP_2, AP_6, AP_8, AP_{10}, and Ap_{12} for requirements R_1, R_2, R_6, R_8, R_{10}, and R_{12} are 0.3517, 0.2401, 0.0401, 0.1936, 0.0559, and 0.0233, respectively; 9 and 3 are the correlation values corresponding to the symbols in the matrix. The requirements with no impact on goal G_2 are not included in the calculation.

The weighted priorities of all goals in Figure 8.6 are calculated using Equation (10). These weighted priorities are then normalized and used in the KP prioritization phase.

All seventeen goals used in RG matrix are used to prioritize KPs in Phase 3. Also, as an example, "Activities Performed" are considered as the representative common feature of key practices in key process areas. Nine activities from level 3 KPAs are selected. The seventeen CMM goals, prioritized using RG matrix, are mapped to these nine activities as shown in Figure 8.7. Impact of each goal on each activity is determined and represented using 9-3-1 standard values. Entries are left blank where there is no impact. Weighted importance of all nine activities is determined. Then they are normalized to maintain consistency and prevent loss of precision.

For example, activity KP_1 (software process is assessed periodically and action plan is developed to address assessment findings) is mapped onto 17 goals. This activity has a moderate correlation with goal G_1 and G_{10}, and strong correlations with goals G_2 and G_3, and a weak correlation with goal G_{16}. It does not have any impacts on other goals. The weighted priority FKP_1 for activity KP_1 is calculated using Equation (11) as following:

$$FKP_1 = \sum_{j=1}^{17} NG_j * IR(KP_1, G_j)$$

$$= NG_1 * IR(KP_1, G_1) + NG_2 * IR(KP_1, G_2) + NG_3 * IR(KP_1, G_3) + NG_{10} * IR(KP_1, G_{10}) + NG_{16} * IR(KP_1, G_{16})$$

$$= 0.1751*3 + 0.1433*9 + 0.1620*9 + 0.0720*3 + 0.0145*1 = 3.5035$$

The normalized priorities NG_1, NG_2, NG_3, NG_{10}, and NG_{16} for goals G_1, G_2, G_3, G_{10}, and G_{16} are 0.1751, 0.1433, 0.1620, 0.0720, and 0.0145, respectively; 9, 3, and 1 are the correlation values corresponding to the symbols in the matrix. The goals with no correlation with activity KP_1 are not included in the calculation.

After the normalized priorities of KPs are calculated, the nine CMM activities prioritized from GP matrix are mapped to fourteen actions in Figure 8.8. These actions are derived from these activities. These actions represent the steps to perform the activities. Impact between each activity and action is determined and represented using 9-

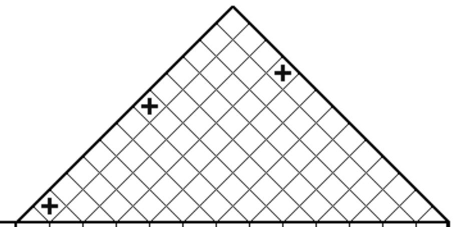
3-1 standard values. Weighted and normalized importance of all seventeen actions is determined along with roof values.

	Requirements Importance	Organization Process Focus (KPA 1)	G1: Process development and improvement activities are coordinated	G2: Strengths and weaknesses of software processes are identified	G3: Organization-level process development and improvement activities are planned	Organization Process Definition (KPA 2)	G4: Standard software process for organization is developed and maintained	G5: Information related to the use of standard software process by software is collected, reviewed and made available	Training Program (KPA 3)	G6: Training activities are planned	G7: Training for developing the skills and knowledge needed to perform software management and technical roles is provided	G8: Individuals in software engineering group and softwa related groups receive necessary training	Integrated Software Management (KPA 4)	G9: Project's defined software process is tailored from organization's standard software process	G10: Project is planned and managed according to its defined software process	Software Product Engineering(KPA 5)	G11: Tasks are defined, integrated, and consistently performed	G12: Software work products are kept consistent	Intergroup Coordination (KPA 6)	G13: Customer's requirements are agreed to by all affected groups	G14: Commitments between engineering groups are agreed to by affected groups	G15: Engineering groups identify, track, and resolve intergroup issues	Peer Reviews (KPA 7)	G16: Peer review activities are planned	G17: Defects in software work products are identified and removed
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	0.1936		●	○	○		○							○	○		●	○							
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	0.0996		▽																						●
R14: Low defect rate	0.0646		▽															▽							●
R15: High reliability	0.0498																							○	●
R16: High requirement satisfaction	0.0581														▽					●				○	
R17: High maintainability	0.0373									●	●	●					○	○							
R18: Usability	0.0138																			▽					
Weighted Importance (FG)			5.3722	4.3953	4.9705		2.6741	0.3762		0.7491	0.7491	0.7491		1.6203	2.2088		3.1198	3.6917		3.0727	1.0395	0.3762		0.4440	3.8707
Normalized Importance (NG)			0.1751	0.1433	0.1620		0.0872	0.0123		0.0244	0.0244	0.0244		0.0528	0.0720		0.1017	0.1203		0.1002	0.0339	0.0123		0.0145	0.1262

Figure 8.6. Goal Prioritization Using RG Matrix (Level 3)

	Goals Importance	KP1: Process is assessed periodically, action plans developed to address assessment findings	KP2: Use of organization's software process database is coordinated at organization level	KP3: Software engineering methods and tools are integrated into defined software process	KP4: Software requirements are developed, maintained, documented, verified according to defined software process	KP5: Organization level training courses are developed and maintained according to organization standards	KP6: Project's defined software process is tailored from organization's standard process	KP7: Data on defects identified in peer reviews and testing are collected and analyzed	KP8: Representatives of project engineering groups conduct periodic technical reviews and interchanges	KP9: Data on conduct and results of peer reviews are recorded
Organization Process Focus (KPA 1)										
G1: Process development and improvement activities are coordinated	0.1751	○	○				●			
G2: Strengths and weaknesses of software processes are identified	0.1433	●	▽				▽			
G3: Organization-level process development and improvement activities are planned	0.1620	●	○	○						
Organization Process Definition (KPA 2)										
G4: Standard software process for organization is developed and maintained	0.0872			●	○		○			
G5: Information related to the use of standard software process by software is collected, reviewed and made available	0.0123				●		○			
Training Program (KPA 3)										
G6: Training activities are planned	0.0244					▽				
G7: Training for developing the skills and knowledge needed to perform software management and technical roles is provided	0.0244					●				
G8: Individuals in software engineering group and softwa related groups receive necessary training	0.0244					○				
Integrated Software Management (KPA 4)										
G9: Project's defined software process is tailored from organization's standard software process	0.0528		▽				●			
G10: Project is planned and managed according to its defined software process	0.0720	○					▽			
Software Product Engineering(KPA 5)										
G11: Tasks are defined, integrated, and consistently performed	0.1017							○		
G12: Software work products are kept consistent	0.1203							○		
Intergroup Coordination (KPA 6)										
G13: Customer's requirements are agreed to by all affected groups	0.1002				○					
G14: Commitments between engineering groiups are agreed to by affected groups	0.0339								○	
G15: Engineering groups identify, track, and resolve intergroup issues	0.0123								●	
Peer Reviews (KPA 7)										
G16: Peer review activities are planned	0.0145	▽						▽	▽	▽
G17: Defects in software work products are identified and removed	0.1262							●		●
Weighted Importance (FKP)		3.5035	1.2074	1.2708	0.6729	0.3172	2.5649	1.8163	0.2269	1.1503
Normalized Importance (NKP)		0.2752	0.0948	0.0998	0.0529	0.0249	0.2015	0.1427	0.0178	0.0904

Figure 8.7. Activity Prioritization Using GP Matrix (Level 3)



	Importance of Activities	A1: Meet quarterly to assess software development processes	A2: Identify areas of improvement in software development processes	A3: Integrate selected methods and tools into standardized process	A4: Hold meetings with stakeholders for requriement analysis according to standardized process	A5: Provide training courses for members involved in software development	A6: Maintain and update content of training according to organizational standards	A7: Evaluate suitability of organization's standard process for current project	A8: Make necessary modification to organization's standard process	A9: Hold peer reviews sessions to identify and collect data on inconsistencies	A10: Record peer review results	A11: Prepare and execute test cases	A12: Analyze defects identified from test cases	A13: Hold technical review meetings on a regular base
KP1: Process is assessed periodically, action plans developed to address assessment findings	0.2752	●	○					○						
KP2: Use of organization's software process database is coordinated at organization level	0.0948	○	▽											
KP3: Software engineering methods and tools are integrated into defined software process	0.0998			●					●					
KP4: Software requirements are developed, maintained, documented, verified according to defined software process	0.0529				●									
KP5: Organization level training courses are developed and maintained according to organization standards	0.0249					●	●							
KP6: Project's defined software process is tailored from organization's standard process	0.2015		○	●				●	●					
KP7: Data on defects identified in peer reviews and testing are collected and analyzed	0.1427	○								●	▽	●	●	
KP8: Representatives of project engineering groups conduct periodic technical reviews and interchanges	0.0178													●
KP9: Data on conduct and results of peer reviews are recorded	0.0904										●			
Weighted Importantce (FA)		3.1893	1.5249	2.7117	0.4761	0.2241	0.2241	2.6391	2.7117	1.2843	0.9563	1.2843	1.2843	0.1602
Normalized Importance (NA)		0.1708	0.0817	0.1452	0.0255	0.0120	0.0120	0.1414	0.1452	0.0688	0.0512	0.0688	0.0688	0.0086

Figure 8.8. Action Plan Development Using AP-HoQ Matrix (Level 3)

For example, action A_1 (Hold meetings with stakeholders for requirements review) has a strong correlation with activity $KP_{1,}$ and moderate correlations with activities KP_2 and KP7. The weighted priority FA_1 for action A_1 is calculated using Equation (12) as following:

$$FA_1 = \sum_{j=1}^{9} NKP_j * IR(A_1, KP_j)$$

$$= NKP_1 * IR(A_1, KP_1) + NKP_2 * IR(A_1, KP_2) + NKP_7 * IR(A_1, KP_7)$$

$$= 0.2752*9 + 0.0948*3 + 0.1427*3 = 3.1893$$

The normalized priorities NKP_1, NKP_2, and NKP_7 for activities KP_1, KP_2, and KP_7 are 0.2752, 0.0948, and 0.1427, respectively; 9 and 3 are the correlation values corresponding to the symbols in the matrix. The activities with no correlation with action A_1 are not included in the calculation.

8.3. APPLICATION EXAMPLE OF SPI BASED ON CMMI CONTINUOUS MODEL USING QFD

In this section, the SPI framework based on CMMI continuous model using QFD is illustrated. This framework starts from the same set of 18 process requirements introduced in Section 8.1. Figure 6.2, Figure 8.1, and Figure 8.2 shows the RI Matrices. Equations (9), (7), and (8) are used to combine the final priorities of the process requirements, normalize them, and calculate the adjusted priorities. These adjusted priorities become the input of the CMMI PA Prioritization phase and the Practices Prioritization phase.

Figure 7.4 shows the mapping between the 18 process requirements and eight Process Areas. The weighted importance values (FPA) and the normalized importance values (NPA) are calculated from the adjusted requirements priorities as well as the correlations between the process requirements and the eight PAs.

Because in CMMI continuous model, various PAs can have different capability levels, two examples showing the SPI of two PAs at two different capability levels are used to illustrate the framework. Figure 7.5 shows the Practices prioritization in the PA of “project planning,” aiming at capability level 1. 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 Figure 7.6.

In this application example, the PA of "project monitoring and control" is used to illustrate the SPI framework based on CMMI continuous model. The aim in this application example is capability level 2. Following the RPA Matrix in Figure 7.4, ten (10) Specific Practices and ten (10) Generic Practices in the PA are mapped to the 18 process requirements, as shown in Figure 8.9.

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

Figure 8.9. Requirements-Practices Impact (RPr) Matrix for PA "Project Monitoring and Control" in CMMI Continuous Model

Correlation between each Practices and each process requirement is determined and represented using 9-3-1 standard values. Entries are left blank where there is no impact. Weighted importance values of all Practices are determined using Equation (14). For example, the Specific Practice (SP1.5), "Monitor Stakeholder Involvement" in Figure 8.9, 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 Equation (14) as following:

$$F\,\mathrm{Pr}_{1.5} = \sum_{j=1}^{18} AP_j * IR(\mathrm{Pr}_{1.5}, R_j)$$

$$= 0.0740*3 + 0.1935*3 + 0.1493*9 + 0.0418*3 + 0.0233*1 + 0.0585*9$$

$$= 2.8214$$

After all weighted importance values in Figure 8.9 are calculated, they are normalized to maintain consistency and prevent loss of precision. These normalized importance values (NPr) are also shown in the figure. These normalized weights are then multiplied by the normalized importance value of this PA they belong to, in this example, the NPA value of the PA "Project Monitoring and Control," which is 2.0289 in Figure 7.4. The resultant Global Importance values (GPr) reflects both the correlation between process requirements and individual Practices and the importance of the PA. These GP values are used at the next phase of the framework as inputs to the AP-HoQ Matrix.

In the last phase of the SPI framework based on CMMI continuous model, the same set of Practices from capability level 2 in PA "Project Monitoring and Control" as shown in Figure 8.9 are entered in the rows of Figure 8.10, and a number of actions derived from these Practices are entered in the columns. The correlation between each Practice-action pair is determined and represented using 9-3-1 standard values as shown in Table 5.1. Based on these correlations and the Global Importance values of Practices, the weighted importance can be calculated for each action using Equation (15).

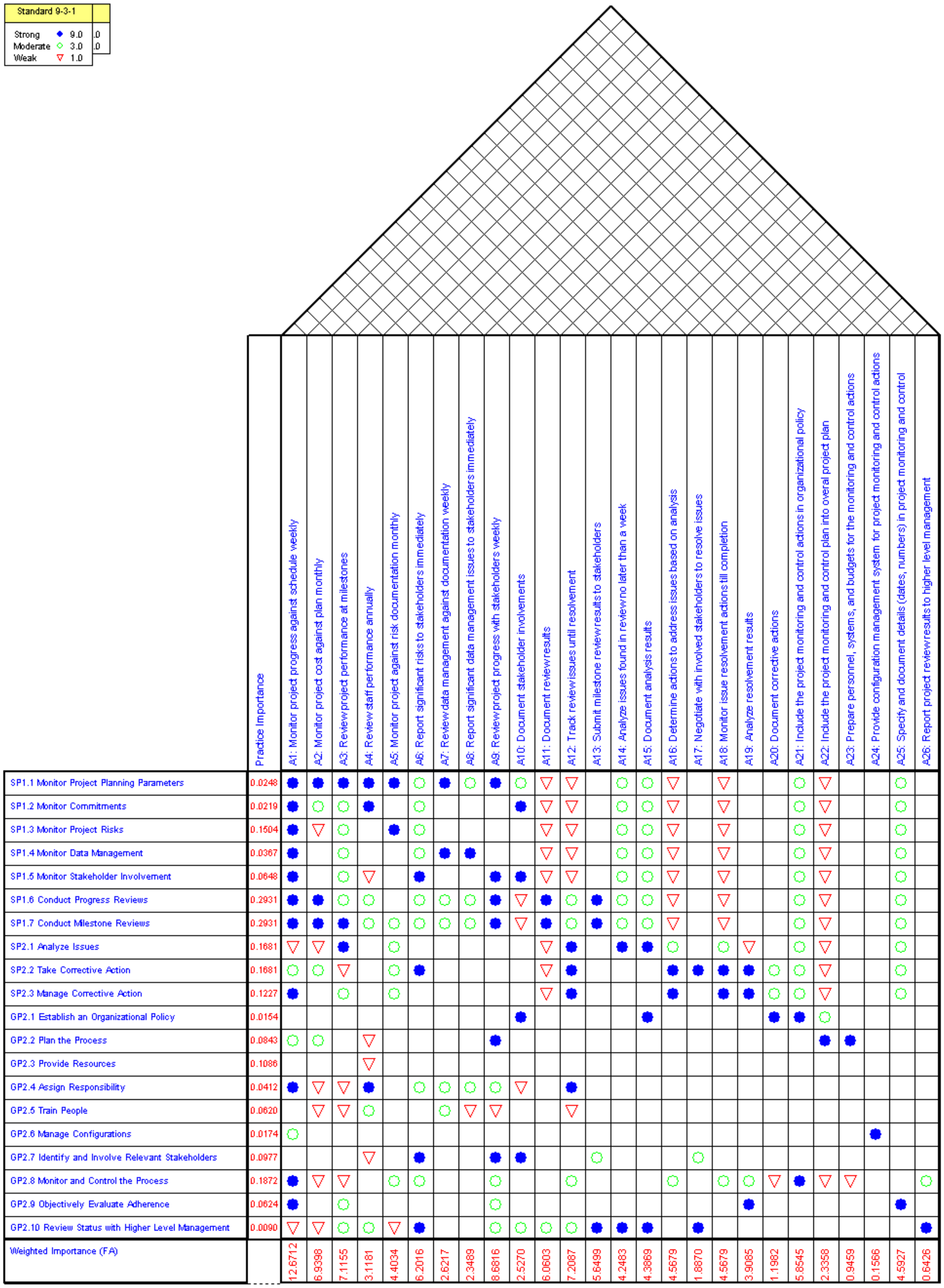


Figure 8.10. Action Plan House of Quality (AP-HoQ) Matrix in for PA "Project Monitoring and Control" in CMMI Continuous Model

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 A1 can be calculated using Equation (15) as following:

$$FA_8 = \sum_{j=1}^{20} NKP_j * IR(A_8, KP_j)$$

= 0.0417*3 + 0.0617*9 + 0.4924*3 + 0.4924*3 + 0.0692*3 + 0.1042*1

= 3.9466

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.

The same calculations are applied to another PA, “Risk Management.” The same set of process requirements in Section 8.1 is used. These requirements has been integrated and prioritized in Phase 1 of the framework and the PAs are prioritized based on their correlations with these requirements (shown in Figure 7.4).

If “Risk Management” PA is also aiming to reach capability level 3, Figure 8.11 shows the RPr Matrix between the set of prioritized process requirements and all Practices in “Risk Management” PA.

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 as shown in Table 5.1 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 Equation (14). For instance, in Figure 8.11, the Weighted Importance value for the Practice SP1.1 is calculated as:

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 Collect 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	0.0585									
R17: High maintainability	0.0373									
R18: High usability	0.0133									
Weighted Importance (FPr)		2.3296	0.3762	0.8044	0.6841	2.3031	2.1726	2.1726	5.5016	5.5016
Normalized Importance (NPr)		0.1066	0.0172	0.0368	0.0313	0.1054	0.0995	0.0995	0.2518	0.2518
Global Importance (GPr)		0.1646	0.0266	0.0568	0.0483	0.1627	0.1535	0.1535	0.3886	0.3886

Figure 8.11. Requirements-Practices Impact (RPr) Matrix for PA “Risk Management” in CMMI Continuous Model

$$F\,Pr_{1.1} = \sum_{j=1}^{18} AP_j * IR(Pr_{1.1}, R_j)$$

$$= 0.0401*9 + 0.0418*1 + 0.0978*9 + 0.0664*9 + 0.0499*9$$

$$= 2.3296$$

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 Figure 8.12. The Practices in "Riks Management" PA and their Global Importance values from Figure 8.11 are entered in the rows of Figure 8.12. The derived actions entered in the columns. The correlation between each Practice-action pair is determined based on the 9-3-1 values from Table 5.1 are entered in the correlation section in Figure 8.12. Based on the importance values of Practices and the correlations, the Weighted Importance values of all actions are calculated using Equation (15).

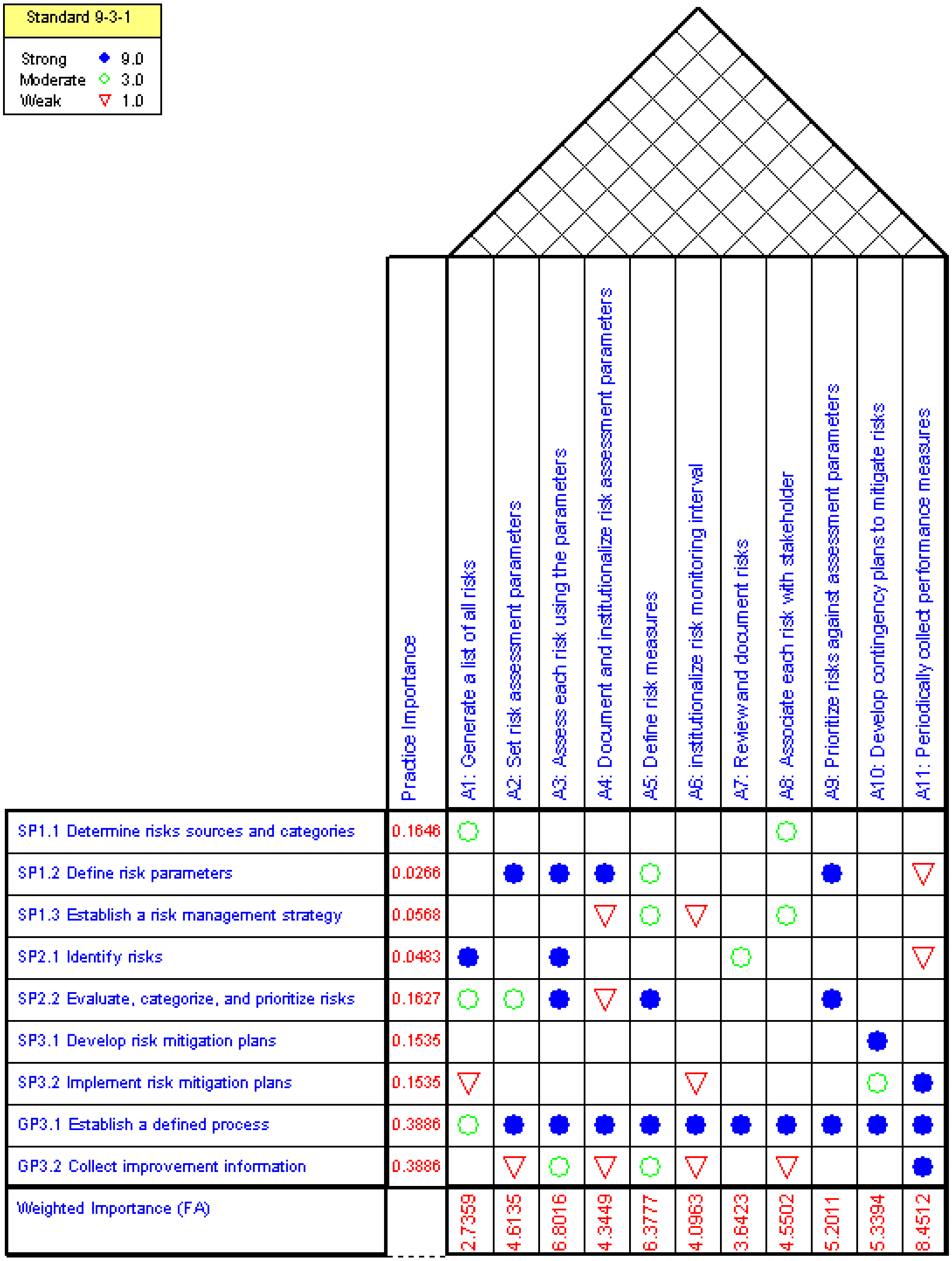
For instance, the Weighted Importance of A1 is calculated as:

$$FA_1 = \sum_{j=1}^{6} NKP_j * IR(A_1, KP_j)$$

$$= 0.1646*3 + 0.0483*9 + 0.1627*3 + 0.1535*1 + 0.3886*3$$

$$= 2.7359$$

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 the whole SPI project can reach a higher level of process requirements satisfaction.



	Practice Importance	A1: Generate a list of all risks	A2: Set risk assessment parameters	A3: Assess each risk using the parameters	A4: Document and institutionalize risk assessment parameters	A5: Define risk measures	A6: institutionalize risk monitoring interval	A7: Review and document risks	A8: Associate each risk with stakeholder	A9: Prioritize risks against assessment parameters	A10: Develop contingency plans to mitigate risks	A11: Periodically collect performance measures
SP1.1 Determine risks sources and categories	0.1646	○							○			
SP1.2 Define risk parameters	0.0266		●	●	●	○				●		▽
SP1.3 Establish a risk management strategy	0.0568				▽	○	▽		○			
SP2.1 Identify risks	0.0483	●		●				○				▽
SP2.2 Evaluate, categorize, and prioritize risks	0.1627	○	○	●	▽	●				●		
SP3.1 Develop risk mitigation plans	0.1535										●	
SP3.2 Implement risk mitigation plans	0.1535	▽					▽				○	●
GP3.1 Establish a defined process	0.3886	○	●	●	●	●	●	●	●	●	●	●
GP3.2 Collect improvement information	0.3886		▽	○	▽	○	▽		▽			●
Weighted Importance (FA)		2.7359	4.6135	6.8016	4.3449	6.3777	4.0963	3.6423	4.5502	5.2011	5.3394	8.4512

Figure 8.12. Action Plan House of Quality Matrix (AP-HoQ Matrix) for PA "Risk Management" in CMMI Continuous Model

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. Because the priority values reflect the weighted importance of PAs, they can be compared across PAs. Even though they are at different capability levels, using the Weighted Importance values calculated from Equation (15), the actions can be prioritized across PAs. Table 8.2 below shows the ranked list of actions from these two PAs.

Table 8.2. 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 resolvement	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

Table 8.2. Prioritized Actions from "Project Monitoring and Control" and "Risk Management" (cont.)

A16: Determine actions to address issues based on analysis	4.5679
A18: Monitor issue resolvement 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 resolvement 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 overal 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

9. CONCLUSION

In today's software development industry, Software Process Improvement typically relies on existing models and standards. Some popular ones include ISO 9000 series of standards, ISO 15504, CMM, CMMI, etc. Some common limitations of these models and standards are the specification of "what to do" but not "how to do it." While making these models and standards widely applicable to many different software development organizations, such limitations also leave many software development organizations in the situation of generating detailed actions in order to comply with these models and standards. In addition, the specifications in the models and standards are not directly related to business goals and other requirements from the organizations.

This study addressed this issue by using QFD as a tool to connect requirements within an organization to the action plans for its process improvement. After careful review of several SPI approaches, CMM and CMMI from SEI were selected as the basis of the of the proposed SPI approach. New SPI frameworks based on both CMM and CMMI from SEI are developed in the study. These new frameworks discuss in detail how to prioritize and integrate requirements, how to map requirements to various components in CMM and CMMI, and how to prioritize action plans.

The proposed frameworks have three objectives: 1) to map process requirements, including business requirements, to CMM or 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.

As introduced in Section 6, the second phase of the SPI framework based on CMM links the process requirements with CMM goals. Similarly, in the SPI framework based on CMMI continuous model, the process requirements are also linked to CMMI PAs and Practices. Through these links, software development organizations can see the direct benefit to the business by reaching a higher level in CMM or CMMI.

In the proposed frameworks, 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.

In this study, a method to integrate requirements from multiples groups in an organization is also proposed in the first phase in the proposed SPI frameworks, which is "Requirement Elicitation/Integration." This method produces outputs that reflect a) the local importance of each process requirement within a perspective, b) the importance of the perspective that a process requirement belongs to, and c) the correlations between a process requirement and all process requirements in other perspectives. Requirements with more and stronger correlations with other requirements from multiple stakeholders are identified. Satisfying these requirements will also satisfy other requirements to some extent. Therefore, they receive higher priority values in this framework. The final importance values serves as good criteria in prioritizing the other components in the SPI frameworks.

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 in Section 8, the actions are related with process requirements through CMM or 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 CMMI Continuous model. As shown in the example along with the SPI model based on CMMI Continuous model, the two PAs are both trying to reach capability level 1. In the application example in Chapter 8, the two PAs in CMMI Continuous model aim at different capability levels—Level 2 and Level 3. In both cases, the SPI framework for CMMI Continuous model works as expected.

To validate the frameworks proposed in this study, a domain expert from Toshiba verified and validated the application examples for the proposed SPI model based on CMM. The integration and prioritization of requirements from various perspectives were evaluated, as well as the impact relationships between requirements and KPA goals, between KPA Goals and KPs, and between KPs and action plans. The evaluation results were positive. Following the same line, the application examples for the SPI framework based on CMMI were developed. The same three objectives as mentioned above were achieved.

BIBLIOGRAPHY

[1] 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.

[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] 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.

[4] 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.

[5] Capability Maturity Model Integration (CMMI), Version 1.2, CMMI-SE/SW/IPPD/SS, V1.2), http://www.sei.cmu.edu/cmmi/models/

[6] Khaled El Emam and Dennis R. Goldenson, “An Empirical Review of Software Process Assessments,” NRC/ERB-1065. November 1999. http://iit-iti.nrc-cnrc.gc.ca/publications/nrc-43610_e.html

[7] Francois Coallier, “How ISO 9001 Fits Into the Software World,” *IEEE Software,* Vol. 11, No. 1, January 1994, pp. 98-100.

[8] Paulk, Mark C., Charles V. Weber, and Mary Beth Chrissis, “The Capability Maturity Model for Software.” In K. El Emam and N. H. Madhavji (eds.), *Elements of Software Assessment and Improvement*, IEEE CS Press, 1999.

[9] Jennifer Gremba, Chuck Myers, “The IDEAL Model: A Practical Guide for Improvement”, Bridge, Issue 3, 1997.

[10] Bamberger J. 1997. Essence of the Capability Maturity Model. *Computer* 30(6): pp.112-114.

[11] Paulk M., Weber C., Curtis B, Chrissis M. Eds. 1995. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA, Addison-Wesley.

[12] CMMI Product Team. 2002. Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation. *Technical Report, CMU/SEI-2002-TR-012, ESC-TR-2002-012*.

[13] CMMI Product Team. 2002. Capability Maturity Model Integration (CMMI), Version 1.1, Continuous Representation. *Technical Report, CMU/SEI-2002-TR-011, ESC-TR-2002-011*.

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

[15] Liu X, Inuganti P., Veera C. 2003. An Integration Methodology for Software Quality Function deployment. *Final Project Report to the Toshiba Corporation*.

[16] Xiaoqing (Frank) Liu, Yan Sun, Praveen Inuganti, Chandra Sekhar Veera, and Yuji Kyoya. "A Methodology for the Tracing of Requirements in Object-Oriented Software Design Process Using Quality Function Deployment," *Software Quality Professional Journal*, September 2007, Volume 9, Issue 4.

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

[18] Zultner, R.E. "Quality Function Deployment (QFD) for Software." American Programmer, 1992.

[19] 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

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

[21] Ita Richardson, Eamonn Murphy, KevinRyan, "Development of Generic Quality Function Deployment Matrix", Quality Management Journal, Vol. 9, No. 2, APRIL 2002, pp. 25-43

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

[23] 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

[24] Saaty, T.L. 1990. *Multicriteria Decision Making: The Analytic Hierarchy Process*. AHP Series, Volume 1. RWS Publications.

[25] Xiaoqing (Frank) Liu, Yan Sun, Yuji Kyoya and Kunio Noguchi, “QFD Application in Software Process Management and Improvement Based on CMM,” *Proceedings of the third workshop on Software quality*, St. Louis, Missouri, 2005.

[26] Xiaoqing (Frank) Liu, Yan Sun, Gautum Kane, Yuji Kyoya, Kunio Noguchi. Business-Oriented Software Process Improvement Based on CMM Using QFD. *Software Process: Improvement and Practice*, November/December 2006, Vol. 11, No. 6, pp. 573-589.

[27] Yamamura, G. and Wigle, G.B., "SEI CMM Level 5: For the Right Reasons", *CrossTalk*, Volume 10 No. 8, Aug 1997.

[28] Leon G. Oldham, David B. Putman, Mark Peterson, Bruce Rudd, and Keven Tjoland, “Benefits Realized from Climbing the CMM Ladder,” *CrossTalk*, Vol. 12, No. 5, May 1999.

[29] Jeff King and Michael Diaz, “How CMM Impacts Quality, Productivity, Rework, and the Bottom Line,” *CrossTalk*, Vol. 15, No. 3, March 2002.

[30] James Herbsleb, Anita Carleton, James Rozum, Jane Siegel, and David Zubrow, “Benefits of CMM-Based Software Process Improvement: Initial Results,” *Technical Report, CMU/SEI-94-tr-013, ESC/SEI-tr-94-013*, August 1994.

[31] Hyde K. and Wilson D. 2004. Intangible Benefits of CMM-based Software Process Improvement. *Software Process Improvement and Practice* 9(4): pp.217-228.

[32] Diane L. Gibson, Dennis R. Goldenson, and Keith Kost, “Performance Results of CMMI-Based Process Improvement,” *Technical Report, CMU/SEI-2006-tr-004*, August 2006.

[33] Xiaoqing (Frank) Liu and Yan Sun, Chandra Sekhar Veera, Yuji Kyoya and Kunio Noguchi, “Priority Assessment of Software Process Requirements from Multiple Perspectives,” *Journal of Systems and Software*, Vol. 79, no. 11, Pages 1649-1660, 2006.

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

[35] Zultner R. E., “Project QFD – Managing Software Development Better with Blitz QFD,” *9th Symposium on QFD*, 1997, pp. 15-26.

VITA

Yan Sun was born on July 27, 1977. He studied English and received a B.A. at Beijing Foreign Studies University in Beijing, China between 1995 and 1999. He received an M.S. in Sociology from Illinois State University in August 2001, an M.S. in Applied Computer Science from Illinois State University in August 2003, and a Ph.D. in Computer Science from the University of Missouri-Rolla in May, 2008.