Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

STUDY OF SPI FRAMEWORK FOR CMMI CONTINUOUS MODEL BASED ON QFD

[1,2]YONGHUI CAO

[1]School of Economics & Management, Henan Institute of Science and Technology
[2]School of Management, Zhejiang University,
E-mail: caoyonghui2000@126.com

ABSTRACT

Software Process Improvement (SPI) is the modification of current software process methods in many software development organizations. It is the key of the continuous process improvement. In this paper, we introduce Software Process Improvement (SPI) and Quality Function Deployment (QFD); the SPI framework based on CMMI contains two portions: 1) SPI framework for CMMI staged model based on QFD and 2) SPI framework for CMMI based on QFD continuous model. This paper focuses on the analysis of the later.

Keywords: *Software Process Improvement, CMMI, QFD*

1. INTRODUCTION

Software Process Improvement (SPI) is the modification of current software process methods in many software development organizations. Its aim is to improve the organization's ability to produce better software products (Humphrey, 1990)[1]. It becomes the key to the survival of many software development organizations. Humphrey (1990) identifies software process improvement in six steps : 1) Understand the current status of the development process; 2) Define a vision of the desired process; 3) Establish a list of required process improvement actions in priority order; 4) Produce a plan to accomplish the required actions; 5) Commit resources to execute the plan; 6) Start over at step 1.

At present, many international models or standards are developed for Software Process Improvement (SPI). For example, these standards have ISO standard, (Capability Maturity Model) CMM, Capability Maturity Model Integrated (CMMI). CMMI is a SPI models from the Software Engineering Institute. In terms of quality and process improvement, these models and standards share some common concerns. CMMI emphasizes continuous improvement while the ISO standard addresses the minimum criteria for a quality system. It is unfair to make a judgment on which one is better(Paulk, 1994) [2].

During process improvement, these standards and models should not be used independently from business and other requirements in an organization. However, considering the more detailed guidance and greater breadth provided by CMMI, it may be a better choice for some software development organizations (Francois Coallier,1994) [3]. Philosophically, the CMMI is a specific implementation of Total Quality Management (TQM). Drawing upon the works of Deming(1986)[4], the CMMI is a framework for improving and integrating systems and software engineering processes. Process improvements have been shown to increase productivity, quality, and cycle times, and result in organizations more accurately predicting schedules and budgets. CMMI is intended to cover both product and service throughout their life cycle of development, deployment, and maintenance, as well as being extensible to incorporate new bodies of knowledge (Chrissis et al., 2003)[5]. The current four bodies of knowledge supported in the current CMMI, also referred to as disciplines, are systems engineering, software engineering, integrated product and process development, and supplier sourcing (Chrissis et al., 2003)[5].

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

QFD has many Benefits. The QFD process provides a great deal of help in obtaining objective, measurable information that can be used for understanding the product to be developed and how it will meet the customers needs for the product. There is guidance in how to carry out the initial information gathering process and what types of information to gather.

QFD allows for a systematic evaluation of customer requirements for a product and features that will meet these requirements. There is a quantification of most of the information processed and this allows for an objective justification for decisions made as a result of that process. The information gathered can help in resolving design tradeoffs and in setting quality goals and measures for development. If two critical features conflict, the conflict will need to be resolved in the design; this information is not known ahead of time and discovered much further along the process when resolving it would be much more difficult (Boushi)[6].

QFD provides a way of tracing requirements from initial definition to completion. Because all steps are recorded and measured it is possible to revisit all decisions and filter changes to the appropriate parts of the project. This is often lost in conventional development processes as decision points are not clearly recorded and the reasoning behind the decisions are easily lost. QFD forces a focus on the customer needs. Any project, whether commercial or in-house, needs to meet the customer needs to be successful. By identifying and quantifying the customer requirements up front QFD ensures that the real requirements are not ignored and tractability helps ensure that they are still visible at the tail end of the project.

QFD can be an aid in shortening development time as it focuses on the essential needs for the product and the essential features to meet those needs. Once the initial costs for QFD are past the process can reduce costs. Trained personnel with appropriate tools can work quickly towards a good solution to a problem. The better, the initial solution the lower, the overall costs will be. QFD leads to a final system that meets the customer needs well and contains features that meet these needs. Features that do not contribute can be identified and excluded early on. In a commercial situation, the better, the solution meets the customer needs the more successful it will be in the market place.

2. QUALITY FUNCTION DEPLOYMENT

2.1 History of Quality Function Deployment

The first articles on QFD appeared in 1972 and in 1978 the process was published as a paperback entitled QFD by Dr. Yoji Akao and Dr.Shigeru Mizuno: An approach to Total Quality Control. The introduction of QFD to America and Europe began in 1983 when the American Society for Quality Control published Akao's work in Quality Progress and Cambridge Research (today Kaizen Institute) invited Akao to give a QFD seminar in Chicago. In 1984, Ford USA learned of the QFD process, and within one year a project was set up with Ford and its suppliers to implement a QFD program. In 1987 The Budd Company and Kelsey-Hayes, both Ford suppliers, developed the first case study on QFD outside Japan. These companies pioneered the development of QFD in the U.S. as an operating mechanism to transform customer expectations into specific design and manufacturing requirements. The first US automobile to reflect the application of QFD was the 1988 Lincoln Continental. Following Ford's lead, other U.S. companies started showing great interest in QFD. All three automakers and many of their suppliers have now adopted QFD as an important tool for listening to the customers' voice. QFD has moved outside of the automotive manufacturing sector and users include such diverse companies as AT&T, Polaroid, Dupont, Florida Power and Light, Ritz-Carlton, and Procter& Gamble among others.

2.2Methodology of Quality Function Deployment

QFD 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 (Akao.1990)[7]. Nowadays, QFD has been applied to virtually every industry and business, including software development t(Liu X, Inuganti P., Veera C. 2003)[8](Xiaoqing (Frank) Liu,2007)[9].

One important technique in QFD is the House of Quality (Figure 1). 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:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

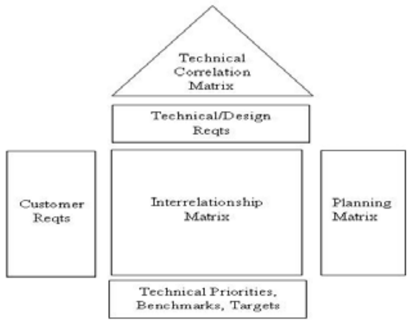

Figure 1.House of Quality in QFD

Table1 Six Major Components Of The House Of Quality

six major components:	description
1.Customer requirements (WHAT's)	A structured list of requirements derived from customer statements.
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 (Akao,1998)[10] (Zultner,1992)[11].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 worldwide. 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, IS014000, 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 [12].

3. SPI FRAMEWORK FOR CMMI CONTINUOUS MODEL BASED ON QFD

3.1 CMMI

The CMMI is the next generation of the SW-CMM process improvement model. Hundreds of organizations worldwide that may have been using the SW-CMM are now transitioning to the CMMI as an organization-wide solution to process improvement. The adoption of the CMMI is also more rapid than the adoption of the SW-CMM when it was first made available [CMMI 2004].

It embodies several different process models and spans multiple domains such as software engineering, systems engineering, software acquisition, workforce management and development, and integrated product and process development. The CMMI encompasses the disciplines of systems engineering, software engineering, Integrated Product and Process Development (IPPD), and supplier sourcing. The CMMI refers to these disciplines as Bodies of Knowledge.

The CMMI consists of 25 process areas, divided into the four categories of Process Management, Project Management, Engineering, and Support. Each of these process areas is described, assigned an identifier, and organized by category. Each process area consists of a set of specific goals, applicable to a single process area, and generic goals, which apply to all process areas as depicted in Figure 2.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

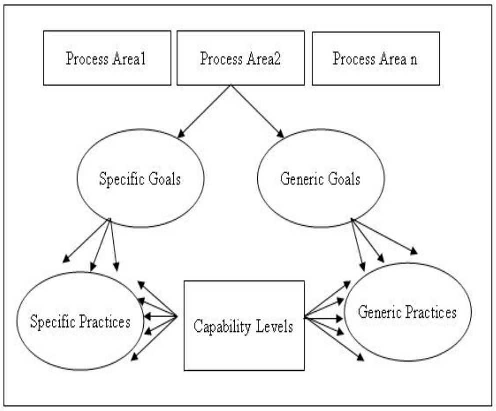

Figure 2. CMMI Model Components [CMMI Product Team 2002]

Each specific and generic goal contains a set of practices, and evidence that these practices are performed by an organization is required for the process area to be considered completely implemented.

There are two CMMI models that may be selected for implementation by an organization: the Staged Model and the Continuous Model. Like the original SW-CMM, the Staged Model has five maturity levels (see Table 2), and each level is associated with a set of process areas (see Table 3). For an organization to achieve and maintain a particular CMMI maturity level, all process areas at that level and all lower levels must be successfully implemented and maintained. A maturity level cannot be attained until all lower levels have been achieved.

Table 2. CMMI Maturity Levels Staged Mode

Level	Name	Description
1	Performed	The process accomplishes the work necessary to produce work products.
2	Managed	A Performed process that is planned and executed in accordance with policy.
3	Defined	A Managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines; has a maintained process description; and contributes work products, measures, and other process improvement information to the organizational process needs.
4	Quantitatively Managed	A Defined process that is described in more detail and performed more rigorously than a managed process
5	Optimizing	A Quantitatively managed process that is changed and adapted to meet relevant current and projected business objectives.

Table 3. CMMI Maturity and Process Area Mapping

Level	Name	Process areas
1	Performed	No process areas performed
2	Managed	Requirements Management(REQM) Project Planning(PP) Project Monitoring and Control (PMC) Supplier Agreement Management (SAM) Measurement and Analysis(MA) Process and Product Quality Assurance(PPQA) Configuration Management(CM)
3	Defined	Requirement Development(RD) Technical Solution(TS) Product Integration(PI) Verification (VER) Validation(VAL) Organizational Process Focus(OPF) Organizational Process Definition (OPD) Organizational Training(OT) Integrated Project Management (IPM) Risk Management(RSKM) Integrated Teaming(IT) Integrated Supplier Management (ISM) Decision Analysis and Resolution (DAR) Organizational Environment for Integration (OEI)
4	Quantitatively Managed	Organizational Process Performance (OPP) Quantitative Project Management (QPM)
5	Optimizing	Organizational Innovation and Development (OID) Causal Analysis and Resolution (CAR)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The second CMMI model, the Continuous Model, was not included in the original SW-CMM. The Continuous Model provides flexibility to organizations in their process improvement strategy. Each process area may be assessed at a capability level independent of the other process areas, which allows an organization to focus on specific process areas. For example, Requirements Development may be at a capability level of four, while Project Planning may be at a capability level of two. Similar to the Staged Model, a higher capability level for an individual process area cannot be attained until all lower levels have been achieved for that process area. This approach benefits organizations that are deficient in certain process areas or only wish to implement those process areas that are aligned with their business objectives. The organization may implement any of 25 process areas at any of 6 capability levels. The six CMMI capability levels for the Continuous Model are described in Table 4.

Table 4. CMMI Capability Levels (Continuous Model)

Level	Name	Description
0	Incomplete	The process is either not performed or partially performed.
1	Performed	The process satisfies the specific goals of the process area and supports and enables the word needed to produce work products.
2	Managed	A Performed process has the basic infrastructure in place to support the process
3	Defined	A Managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines and contributes work products, measures, and other process improvement information to the organizational process assets.
4	Quantitatively Managed	A Defined process that is controlled using statistical and other quantitative techniques.
5	Optimizing	A Quantitatively Managed process that is improved based on an understanding of the common causes of variation inherent in the process.

3.2 CMMI and QFD

Our SPI framework works with CMMI, which is gaining popularity in the industry. Again, QFD is used to help with the SPI based on CMMI. 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 Practice 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 Practiced 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. As a result, the priority value of each requirement is adjusted after the impacts from the other requirements are assessed.

3.3 SPI framework for CMMI continuous model using QFD

The techniques of correlation-based prioritization with the help of QFD are used in the SPI framework for CMMI continuous model. In this model, 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. At the next phase, generic practices for the generic goals, and specific practices for specific goals at various capability levels are prioritized. Because in CMMI continuous model, different PAs can have

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

different of capability levels, the priorities of Practices at different capability levels are determined by their correlations with the same set of process requirements. Different PAs can have different of capability levels. The prioritization of Practices should be done for individual PAs. Then, the Practices in each level of individual PAs are prioritized separately in this framework. 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.

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.

Table5 four phases in CMMI continuous model

Phase	Description
Phase1	Requirements Elicitation/Integration
Phase2	CMMI PAs Prioritization
Phase3	Practices Prioritization
Phase4	Action Plan Prioritization

In the phase 1, various perspectives are represented as P1 through Pn. At the same time, each perspective contains many requirements. For instance, the software process requirements in perspective 1 are represented as R1-1, R1-2, etc.

In Figure 3, 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. These integrated requirements are represented as Rl through Rm.

The prioritization ensures that requirements 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. 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).

The second phase 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.

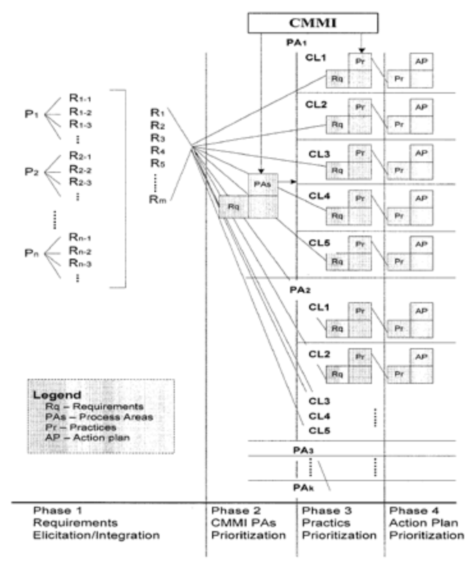

Figure 3. Software Process Improvement through CMMI Continuous Model Using QFD

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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 is "practice prioritization". It 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.

4. CONCLUSIONS

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, CMMI from the Software Engineering Institute (SEI) were selected as the basis of the of the proposed SPI approach. New SPI frameworks based on CMMI from SEI are developed in the study. This new framework discusses in detail how to prioritize and integrate requirements, how to map requirements to various components in CMMI, and how to prioritize action plans. The proposed framework has three objectives: 1) to develop a method, based on QFD, for the integration and prioritization of requirements from multiple perspectives (groups); 2) to map process requirements, including business requirements, to CMM or CMMI with the help of QFD; and 3) to be able to prioritize software process improvement actions based on process requirements.

REFRENCES:

[1] Humphrey, W. S. Managing the software process Reading, MA: Addison-Wesley ,1990.

[2] 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.

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

[4] Deming;W .E.(1986).Out of the crisis. Cambridge MA:MIT Center for Advanced Engineering.

[5] Chrissis, M. B., Konrad, M., Shrum, S. (2003). CMMI: guidelines for process integration and product improvement. Boston, MA: Addison-Wesley Publishing Company.,2003.

[6] Boushi, M. "Towards better object-oriented software with QFD," Transaction of 6th Symposium on QFD.

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

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

[9] 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.

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

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