

## Quality standards for a software industry –A review

Monika Yadav<sup>1</sup>, Sanjeev Kumar<sup>2</sup> and Kaushik Kumar<sup>3\*</sup>

<sup>1</sup>(Department of management, Birla institute of technology, Mesra ,India)

<sup>2</sup>(Department of Computer Sc.& engineering, Laxmipati Institute of sc.& tech., RGPV University, Bhopal,India)

<sup>3</sup>(Department of mechanical engineering, Birla institute of technology, Mesra ,India)

---

**Abstract:** In the present era of globalization QUALITY has immerged as the major factor for survival and the organizations are issuing various standards in order to achieve the same. But for any organization pursuing the quality and the subsequently continuously improving the same, the challenge is to select a particular standard. For a software organization the task has become more difficult as majority of the globally existing standards have been set considering the manufacturing sector – the dominant industry in 19<sup>th</sup> century. The main objective of this paper is to evaluate the available quality standards for the software industry with an intention to describe the strengths and weaknesses of each of them. A particular software organization depending on their specific requirement can select the best possible option amongst these for successful implementation and in the end survival.

**Keywords:** Capability maturity model, Capability Maturity Model Integration, ISO 9001, ISO 9000-3.

---

### I. INTRODUCTION

The software industry, one of the fastest growing businesses over the last two decades, is also facing the problem of global competition. It has suffered due to the recent recession in the global economy and the path of revival is dictated by the quality of the output. The ‘safety-critical nature of software’ also has accentuated the necessity for ‘quality’ of software. There is no unique definition for ‘software quality’, and its domain is hazy. Consequently, there is no agreement on how to actually achieve it. In spite of all this, nobody has denied its significance. The consensus is that there is a greater need to foresee and respond to the customer needs, and to ensure quality of the ‘processes’ in complex software development projects.

In this context the selection of standard would be a crucial decision affecting the future of the organization in particular and industry in general. The situation has aggravated as the market has turned from seller to buyer dominance. Some of the customers have also started demanding for quality in accordance with some global standard. This has made the situation much more difficult for a software organization as the majority of the existing standards were framed during World War I & II when the prominent industry was manufacturing and software was nonexistent.

In this work effort has been made to list out the various standards suitable for a software organization and the evaluation of each of them.

### II. QUALITY STANDARDS

Every industry is striving for quality in both perspective either in customer point of view or it can be business point of view for which they are continuously improving their working culture in order to provide customer satisfaction up to maximum level. The aim of Software Quality Management (SQM) is to manage the quality of software and of its development process. Then (QMS) help them in every aspect which is mainly focused on the organizational structure, procedures, processes and resources needed to implement in order quality management. Early systems emphasized predictable outcomes of an industrial product production line, using simple statistics and random sampling. By the 20th century, labour inputs were typically the most costly inputs in most industrialized societies, so focus shifted to team cooperation and dynamics, especially the early signaling of problems via a continuous improvement cycle. In the 21st century, QMS has tended to converge with sustainability and transparency initiatives, as both investor and customer satisfaction and perceived quality is increasingly tied to these factors. Here the emphasis is on four standards with their some pros and cons with respect to software industry.

#### 1.1. ISO 9001

The ISO 9000 family of standards [1-7] is related to quality management systems and designed to help organizations ensure that they meet the needs of customers and other stakeholders while meeting statutory and regulatory requirements related to the product. The standards are published by ISO, the International Organization for Standardization, and available through National standards bodies. ISO 9000 deals with the fundamentals of quality management systems, including the eight management principles on which the family

of standards is based. Since they first appeared in 1987, the ISO 9000 series of standards have generated much interest in industry and governments worldwide. It is the key standard in the ISO family which Quality Systems Model for Quality Assurance in Design, Development, Production, Installation and Servicing.

ISO 9001 deals with the requirements that organizations wishing to meet the standard have to fulfill. The global adoption of ISO 9001 may be attributable to a number of factors. A number of major purchasers require their suppliers to hold ISO 9001 certification. In addition to several stakeholders' benefits, a number of studies have identified significant financial benefits for organizations certified to ISO 9001, with a 2011 survey from the British Assessment Bureau showing 44% of their certified clients had won new business. Corbett et al. [8] showed that certified organizations achieved superior return on assets compared to otherwise similar organizations without certification. Heras et al. [9] found similarly superior performance and demonstrated that this was statistically significant and not a function of organization size. Naveha and Marcus [10] claimed that implementing ISO 9001 led to superior operational performance in the US motor carrier industry. Sharma [11] identified similar improvements in operating performance and linked this to superior financial performance. Chow-Chua et al. [12] showed better overall financial performance was achieved for companies in Denmark. Rajan and Tamimi [13] showed that ISO 9001 certification resulted in superior stock market performance and suggested that shareholders were richly rewarded for the investment in an ISO 9001 system. As the standard was primarily made for manufacturing industry hence there are some pros and cons when it is used for the software organization as depicted in table 1

Table 1 Pros and Cons of ISO 9001

Sl.No	Pros	Cons
1	Improvement in "bottom line" profit through: Better efficiency, Continual improvement, Less waste.	Costly to obtain and maintain
2	Managed control of key processes	Lengthy time-scale to develop and obtain certification
3	Promotion and standardization of good working Practices	Difficult to implement
4	Greater marketing appeal and improved public Relations	Organizational resistance to change
5	Meeting the requirements for inclusion on some tender lists.	Staff resistance to change
6	A system for managing employee training	Hard to maintain enthusiasm for the system
7	The effective management of risk	More documentation
8	A vehicle for introducing a culture for opportunity	

There are so many shortcomings occurring while implementing it in to software organization so a modification was done and ISO 9000-3:1997 quality standard came into existence in order to help companies to apply the ISO 9001:1994 standard to computer software. It has been used to develop, supply, install, and maintain computer software. And it is really an expanded version of the old ISO 9001:1994 standard. ISO has simply copied the old text from ISO 9001 and pasted it into the new version of ISO 9000-3, and then added some new text that refers only to software.

### 1.2. ISO 9000-3

ISO9000-3 is typically considered as an international standard for quality. It is the defining standard accepted by almost all European and North American countries. Software development companies are often required to conform to ISO standards in order to be considered for contract awards. ISO-90003 provides special guidelines for implementing ISO-9000, and was created specifically for the software development industry. It is used for system development which is contributed in “guidelines for the development, supply and maintenance of software. In this life cycle of model is taken as the process to be controlled. The software development guidance document for ISO 9001 remains as a very general description of the procedures and guidelines for quality management. The main criticisms of the use of ISO 9000-3, for system development, are:

- The generic nature of standard ,
- The fact that is specified for quality system development organization and that title attention is given to quality measures on the level of specific development projects.
- The ISO 9000 recognize only one certifiable level of quality, while in practice different quality level may be advisable for different situation.
- The emphasis’ on strict procedure and their documentation in manual leads to bureaucratic type of behavior instead growing awareness of the importance of quality for the organization searching for new insight methodologies in system development may be hampered. Boehm’s spiral is the good example of it which copes with uncertainty in system development but clashes with the rules of ISO 9000-3

Though through-put time may be important in some cases than superb quality and a somewhat quick and dirty approach might then be advisable, quality standards leave very little room for the idea of end-user and rapid application development.

The above resulted in the development of standards which can take care of the shortcoming of ISO system, especially for a software industry. ISO 9000-3 serves as a guide for interpreting ISO 9001 standard for "the development, supply, and maintenance of software. Hence for a software producer, ISO 9001 must be interpreted according to 9000-3 guidelines, so the only meaningful certification is ISO 9001/9000-3. In this they primarily observe that to measure the quality should address the organization's capabilities. Software producers must meet demanding requirements. They must deliver a product or enhancement that: meets user needs, has minimal defects that do not affect the user's ability to use the product; is delivered when required; and has the lowest life-cycle costs

"Product" in this context is what the user receives, uses and perceives. For embedded software, this means the software, hardware, user documentation, and training and support services. Such a capability is the result of not only a good quality system but also of assets like leadership, human resources, marketing, processes, and technology and capital. ISO 9001/9000-3 partially supports the first four categories:

- Leadership (in terms of strong managers and sound development strategies): ISO 9001 covers quality leadership issues such as commitment to quality and ensuring that policy is understood, implemented, and maintained at all levels in the organization. However, it covers only a fraction of that covered by the Baldrige Award examination criteria, which explicitly require (among other things) that senior executives demonstrate and reinforce the organization's customer orientation.
- Human resources: This includes intellectual assets, because engineering is intellectually intensive. ISO 9001 covers this by mandating that procedures be established and maintained for identifying the training needs of all personnel who perform activities that affect quality. It also mandates that personnel who perform specific tasks be qualified on the basis of education, training, and experience, as required it does not go into any detail on either of these requirements. It is up to the auditor and certifying body to interpret this. ISO 9001 also does not address total-quality-management issues, like morale and motivation.
- Marketing: The knowledge of the right product for the right market at right time, ISO 9001 weakly addresses this. Coverage is limited to contract review, although ISO 9000-3 extends to customer obligations and joint customer-supplier review. It does not cover market research and how to use customer-satisfaction data, for example.
- Processes: Processes must be in place and mature - under control and under continuous improvement. An organization that develops, markets, and supports software must have many processes in place, from human resources development, marketing, and purchasing to reuse, life-cycle-cost modeling, and technology, reliability and configuration management. The key strength of ISO 9001 is in its quality system processes.

As no standard can be foolproof hence ISO 9000-3 also has pros and cons.

#### Pros

- It is recognized all over the world
- The ISO standards prescribe one fixed level of quality management.
- There are some general guidelines on the measurement of product and process

#### Cons

- Outwardly focused, ISO 9000-3 is not specially aimed at software development,
- ISO 9000-3 Guidelines is not much easier to implement or building quality system in software development.
- It models the development process not in a detailed way than like CMM
- Quality characteristics of the clauses in ISO 9000-3 are not described in detail than the quality characteristics of CMM.
- The structuring of the quality characteristics in ISO 9000-3 is not makes it easier to discuss quality matters.

One of the major drawbacks of the ISO family of standards is that they do not address the procedure for continuous improvement. This has given rise to an altogether new standard, which is neither an extension nor a modification of ISO, focused on continuous improvement in software quality.

### **1.3. CAPABILITY MATURITY MODEL**

The capability maturity model (CMM) was the first process methodology that tried to model software engineering process systems. CMM [14-20] was initially developed in the Software Engineering Institute (SEI) at Carnegie-Mellon University in 1987. The current version of CMM (Version 1.1) was released in 1993. The requirements for distinguishing and selecting matured software providers led to the development of the SEI method for assessing software project contractors [15-16] and the SEI capability maturity model (CMM) for software [17-20]. A set of important concepts and successful experience, such as process, quality, and management techniques, have been introduced into software engineering from management sciences. However,

in addition to the initial goals of CMM for software engineering management capability modeling and software organization maturity measurement, researchers and the software industry soon realized that the concept of software process introduced in CMM is a universal model for organizing software engineering. This led to studies in a new approach to process-based software engineering and the development of a number of new software process models and standards. CMM (V.1.1) was developed by Paulk and his colleagues [17] with the supplement of a set of detailed key practices [10], and a questionnaire by Zubrow et al. [21]. A summary of differences between Version 1.0 and Version 1.1 was provided by Paulk et al. [19]. Humphrey and other researchers have also contributed towards the development CMM [22-24]. People have reported the applications and case studies of CMM which has made the system acceptable to the software organizations. For the proper evaluation, the relationships of CMM with other process models were studied and published [22-34]. Like any system or process CMM also has shortcomings which were studied by scholars [35-37].

In a nut shell the Capability Maturity Model for Software describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes. For software CMM is organized into five maturity levels as depicted in Table 2.

Table 2. Capability maturity model levels

S.NO	Level	Description
	Level 0: Incomplete Processes	Absence of processes or general failure to achieve process objectives. No process to product realization.
1.	Level 1 (The Initial Level): Performed Processes	The first and the lowest level in CMM is the <i>Initial</i> level. At this point organizations have few or no processes. Successes are mainly due to individual initiative and effort and processes that may exist are given a go-bye in crisis. The outcome of a project is therefore unpredictable.
2.	Level 2 (The Repeatable Level): Managed Processes.	At repeatable level, the processes are followed at the project level for various software project management functions and their performance is planned and tracked through a documented process. At this level, since the project management processes are in place, the organization is ‘disciplined’ and processes are expected to repeat successful practices as done in similar projects.
3.	Level 3 (The Defined Level): Established Processes	At this level, the organization defines processes for software engineering and management is standardized across the organization. Tailoring guidelines are developed to create project defined software processes and activities become stable and repeatable for implementing them organization-wide.
4.	Level 4 (The Managed Level): Predictable Processes.	It is reached when the organization uses quantitative goals for managing. Quantitative goals are set for software products and processes, using an organization-wide measurement program. The level involves a quantitative understanding of process capability and using this to manage processes. Variation in process performance is tracked and risks are identified and managed.
5.	Level 5 (The Optimizing Level): Optimized Processes	It is the highest maturity level of the CMM. At this level, the organization improves continuously, setting new goals and responding to new technologies and challenges. Processes are cost-effective and are improved over time to meet the organization needs. At this highest level, the process performance is measured for continuous process improvement to verify whether the changes in the processes are providing the expected benefits.

CMM has some merits and demerits in comparison to the ISO standards which can be summed up as depicted below:

Merits:

- CMM is a de facto standard in the USA,
- CMM is specially aimed at software development.
- CMM Guidelines is much easier for building quality systems in software development
- It models the development process in a much more detailed way than ISO 9000.
- Quality characteristics of the levels in CMM are described in much more detail than the quality characteristics of ISO 9000-3.
- The structuring of the quality characteristics in CMM makes it easier to discuss quality matters

Demerits:

- Inwardly focused.
- CMM points out that different level are possible but do not prescribe which level a systems developer has to choose.

The biggest difference is the emphasis in CMM on continuous process improvement where as ISO only addresses minimum criteria for an acceptable quality system. So in connection to a software industry it is observed that for certain things both play a genuine role but if any standard is to be completely adhered to in respect of a software industry, CMM is the better approach as compared to ISO 9000-3.

Hence CMMI came in to picture due to some shortcoming occurs in CMM while implementing it in to software industry and the Implementation of CMM raised many challenges that led to development of CMMI as

an improvement. However it does not replace CMM and the effectiveness depends on the specific area of application.

The Capability Maturity Model does come with some inherent drawbacks like:

- One of which is that when organizations use CMM, they look at each level as a target. They make their goal to reach the next level up. This can be a dangerous thought because if you become fixated on reaching the next level, you begin to lose perspective and forget that the real goal is to actually improve the processes.
- Similarly the CMM does not specify a particular way of achieving those goals. In order to achieve them one needs to think in a flexible way. The goals will only be achieved if the organizations processes are taken into account, as each organization is different so the steps needed for process improvement will be different. Just because one organization follows the rules set by the CMM it does not guarantee that it will be successful as there are other factors involved.
- CMM only helps if it is put into place early in the software development process. For example, if there is a process that is in a crisis then CMM will not help overnight. It can't be used as an emergency method of recovering from a difficult position.
- CMM is concerned with the improvement of management related activities. Whilst this is a big issue in the software development process it is not necessarily the most important thing to look at. Improved quality of code may be a vital issue in the context of software
- CMM model is superseded by CMMI, it is a reference model of matured practices in a specified discipline like Systems Engineering CMM, Software CMM, People CMM, Software Acquisition CMM etc. But they were difficult to integrate as and when needed.
- The CMM model proved useful to many organizations, but its application in software development has sometimes been problematic.
- CMM Key Process Area (KPA) concentrates on the completion of specific tasks or processes and does not motivate the organization to focus on process architecture.

#### 1.4. CMMI : CAPABILITY MATURITY MODEL INTEGRATION

Currently, there are several maturity models, standards, methodologies, and guidelines that can help an organization improve the way it does business. However, most available improvement approaches focus on a specific part of the business and do not take a systemic approach to the problems that most organizations are facing. For example, maturity models such as the Software Engineering Institute's (SEI's) Capability Maturity Model for Software (SW-CMM), which focuses on improving software, and the Electronic Industries Alliance's (EIA's) Systems Engineering Capability Model (SECM), which focuses on systems engineering are available. By focusing on improving only one area of a business, these models have unfortunately perpetuated the barriers that exist in organizations [38]. Capability Maturity Model Integration (CMMI) provides an opportunity to avoid or eliminate these barriers through integrated models that transcend disciplines. CMMI consists of best practices in software companies. It addresses practices that cover the product's life cycle from conception through delivery and maintenance. There is an emphasis on both systems engineering and software Engineering and the integration necessary to build and maintain the total product [39]. It is an integrated model of many CMMs intended to achieve process improvement. CMMI has two representations Staged representation, Continuous representation [40-41].

##### 2.4.1. Staged Representation

CMMI pushes to increase the maturity of the processes, focuses improvement on the process capability an organization can expect to attain; however, this expected capability or ability to function in a mature manner is contained within maturity levels or stages. This representation provides a roadmap for sequencing the implementation of groups of process areas. There are five maturity levels, ranges from level 1 to 5 as shown in figure 1, with each level providing the foundation for further improvements.



Fig 1: Staged representation of maturity levels in CMMI

### 2.4.2. Continuous Representation

It has the same basic information as the staged representation, which arranged differently, provides maximum flexibility. In this each process capability level ranges from 0 to 5 which are depicted in figure 2. The continuous representation provides flexibility for selecting processes fit for achieving business goal of the organization [42].

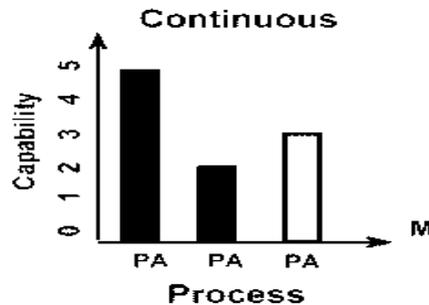


Fig 2: Continuous representation in CMMI

CMMI provides 25 process areas means a cluster of related practices in these areas which are implemented collectively, satisfies a set of goals considered important for making significant improvement [43]. The CMMI model is a process that focuses on what to do, not how to do it or who does it. Gist of CMMI is to provide guidance for improving organization's processes and ability to manage the development, acquisition, and maintenance of products or services.

CMM measures the maturity level of an organization by determining if an organization completes the specific activities listed in the Key Performance Areas (KPA), oblivious to whether the completion of such activity leads to the desired result. CMMI is also an activity based approach but the major difference is that CMMI takes a more result-oriented approach when defining and measuring Key Performance Areas.

CMM KPA concentrates on the completion of specific tasks or processes and does not motivate the organization to focus on process architecture. CMMI, on the other hand has an iterative lifecycle that integrates the latest best practices from the industry and attacks risks in process architecture at an early stage.

CMMI supersedes CMM in software development processes, but CMM is still relevant and appropriate for sequential, activity-based management paradigm.

There are numerous benefits and shortcomings of implementing CMMI in an IT / Software Development Organization, some of them can be summed up as below:

Benefits:

- CMMI is the successor of the CMM and evolved as a more matured set of guidelines and was built combining the best components of individual disciplines of CMM (Software CMM, People CMM etc). It can be applied to product manufacturing, People management, Software development etc.
- The Capability Maturity Model Integration (CMMI) project was formed to sort out the problem of using multiple CMMs.
- CMMI is also an activity based approach but the major difference is that CMMI takes a more result-oriented approach when defining and measuring Key Performance Areas.
- CMMI, on the other hand has an iterative lifecycle that integrates the latest best practices from the industry and attacks risks in process architecture at an early stage
- CMMI supersedes CMM in software development processes.
- CMMI describes integrated processes and disciplines as it applies both to software and systems engineering.
- CMMI continuous grants freedom in improving only those process areas which are critical for organization to improve and mitigate the risk
- CMMI Continuous enables increased foresight into process area capability improvement.
- CMMI allows improvement in different processes at different rates and gives cost and time flexibility to organization.
- Culture for maintaining Quality in projects starts in the mind of the junior programmers to the senior programmers and project managers
- Centralised QMS for implementation in projects to ensure uniformity in the documentation which means less learning cycle for new resources, better management of project status and health
- Incorporation of Software Engineering Best Practices in the Organizations as described in CMMI Model
- Cost saving in terms of lesser effort due to less defects and less rework

- This also results in increased Productivity
- On-Time Deliveries
- Increased Customer Satisfaction
- Overall increased Return on Investment
- Decreased Costs
- Improved Productivity

#### Shortcomings

- CMMI-DEV is may not be suitable for every organization.
- It may add overhead in terms of documentation.
- May require additional resources and knowledge required in smaller organizations to initiate CMMI-based process improvement.
- May require a considerable amount of time and effort for implementation.
- Require a major shift in organizational culture and attitude.

### III. CONCLUSION

In this present scenario there is a conflict between the desires for better quality systems resulting in measurable improvements of the systems development process and the way in which this is implemented by means of standards. Therefore Quality certificates can create a wrong impression of the real capabilities of an organization. This should not, in any sense, lead to a conclusion as abolition of standards. The majority of organizations can improve themselves by logical use of the existing standards. But there is also a need for improving the present standards. The quality of systems can be improved, in the shortest possible span, by an approach that is specially constructed for assessing systems development environments. This would support measurable improvement of the development process and would also advocate for the choice and implementation of actions. Although not absolutely perfect, but CMM and CMMI still offers more possibilities in this respect than does ISO 9000-3. In the longer term standards and certificates will have to be attuned the diversity that exists in the real world. It is felt that this transformation process will take some time and that therefore the organizations have to carefully assess the value of the quality certificates in different situations. Finally it can be stated that the CMM methodology provides for more possibilities for the construction of an appropriate software quality system than does ISO family standards. Moreover it can be summarized that a software company can use ISO 9000-3 for the certification and streamlining of the documentation for the processes involved or responsible for the betterment of the quality systems and then can directly switch over to CMM and work towards continuous improvement by achieving the various levels. From the above it is quite clear that the ISO 9000 family of standards in the form can be implemented in the software organization but the effect can be noticed in the working of the organization and not on the developmental activities. It can also be inferred that although the levels of both CMM and CMMI are similar but the difference lies in perception of these. CMM creates a standard model where as CMMI integrates different CMM models and creates an integrated CMM customized for the specific organization. Hence the CMMI has provided significant value for many organizations that have used it as a guide for improving the way they do their engineering work. It has helped them to gain control over their processes Management, Engineering, and supporting processes to assure that those processes serve the needs of the organization. Since CMMI is a customized standard hence by following the same more consistent success in the engineering projects can be achieved by putting the organization on the road to more effective processes. Most software organizations can start with CMM then add CMMI. When a software organization aims to achieve software process improvement, it is necessary to address more aspects, likes CMMI.

### REFERENCES

- [1] "ISO 9000-1: 1994", *Quality management and quality assurance standards*, (Part 1: Guidelines for selection and use, 1994).
- [2] "ISO 9001: 1994", *Quality systems. Model for quality assurance in design, development, production, installation and servicing*, 1994.
- [3] "ISO 9002: 1994", *Quality systems. Model for quality assurance in production, installation and servicing*.
- [4] "ISO 9003: 1994", *Quality systems. Model for quality assurance in final inspection and test*.
- [5] "ISO 9004-1: 1994", *Quality management and quality system elements*. (Part 1: Guidelines, 1994).
- [6] "ISO 9004-4: 1993", *Quality management and quality system elements*. (Part 4: Guidelines for quality improvement, 1993 )
- [7] "ISO 9000-3: 1992", *Quality management and quality assurance standards*. (Part 3: Guidelines for the application of ISO 9001 to the development, supply and management).
- [8] Corbett, Charles J, Montes Sancho, María J, Kirsch, David A Song, CX Wu, H Dai, Q Irier, H Upadhyay, AK ,The Financial Impact of ISO 9000 Certification in the United States An Empirical Analysis Management Science, Vol.51,2005,1046-1059.
- [9] Heras, Iñaki, Dick, Gavin P.M, Casadesús, Martí, ISO 9000 registration's impact on sales and profitability: A longitudinal analysis of performance before and after accreditation, *International Journal of Quality & Reliability Management*, Vol.19, 2002, 774-791.

- [10] Naveh, Eitan, Marcus, Alfred, Financial performance, ISO 9000 standard and safe driving practices effects on accident rate in the U.S. Motor carrier industry", *Accident Analysis & Prevention*, Vol. 39 ,2007,pp. 731-742.
- [11] Sharma and Divesh S "The association between ISO 9000 certification and financial performance", *The International Journal of Accounting*, Vol.40, 2005, pp.151-172.
- [12] Chow-Chua, Clare; Goh, Mark; Wan, Tan Boon, Does ISO 9000 certification improve business performance?, *International Journal of Quality & Reliability Management*, Vol.20, 2003, 936-953.
- [13] Alan C. Gillies, *Software Quality: Theory and Management*, Chapman & Hall, Ltd., London, UK, 1992.
- [14] Humphrey, W.S. and Sweet, W.L. (1987), *Method for Assessing the Software Engineering Capability of Contractors*, Technical Report CMU/SEI-87-TR-23, Software Engineering Institute, Pittsburgh, PA.
- [15] Humphrey, W.S. (1988), *Characterizing the Software Process: A Maturity Framework*, IEEE Software, March, pp.73-79.
- [16] Humphrey, W.S. (1989), *Managing the Software Process*, Addison-Wesley Longman, Reading, MA.
- [17] Paulk, M.C., Curtis, B., Chrissis, M.B. and Weber, C.V. (1993), *Capability Maturity Model for Software*, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR- 24, February.
- [18] Paulk, M.C., Weber, C.V., Garcia, S., Chrissis, M.B. and Bush, M. (1993), *Key Practices of the Capacity Maturity Model, Version 1.1*, Technical Report CMU/SEI- 93-TR-25, Software Engineering Institute, Pittsburgh, PA.
- [19] Paulk, M.C., Curtis, B., Chrissis, M.B. and Weber, C.V. (1993), *Capability Maturity Model, Version 1.1*, IEEE Software, Vol.10, No.4, July, pp.18-27.
- [20] Paulk, M.C., Weber, C.V. and Curtis, B. (1995), *The Capability Maturity Model: Guidelines for Improving the Software Process*, SEI Series in Software Engineering, Addison-Wesley.
- [21] Zubrow, D. (1997), *the Software Community Process Maturity Profile*, Software Engineering Institute, Pittsburgh.
- [22] Humphrey, W.S., Snyder, T.R. and Willis, R.R. (1991), *Software Process Improvement at Hughes Aircraft*, IEEE Software, July, pp.11-23.
- [23] Kitson D.H. and Masters, S. (1992), *An Analysis of SEI Software Process Assessment Results: 1987-1991*, Technical Report CMU/SEI-92-TR-24, Software Engineering Institute, Pittsburgh.
- [24] Saiedian, H. and Kuzara, R. (1995), *SEI Capability Maturity Model's Impact on Contractors*, IEEE Computer, Vol.28, No.1, pp.16-26.
- [25] Paulk, M.C. (1995), *how ISO 9001 compares with the CMM*, IEEE Software, January, pp.74-83.
- [26] Kitson, D.H. (1996), *Relating the SPICE Framework and SEI Approach to Software Process Assessment*, Proceedings of International Conference on Software Quality Management (SQM.96), MEP Press, London, pp. 37-49.
- [27] Wang, Y. and G. King (2000), *Software Engineering Processes: Principles and Applications*, CRC Press, USA, April, ISBN: 0849323665, pp.1- 752.
- [28] Wang Y., King, G., Doling, A. and Wickberg, H. (1999), *A Unified Framework of the Software Engineering Process System Standards and Models*, Proceedings of 4th IEEE International Software Engineering Standards Symposium (IEEE ISESS.99), IEEE CS Press, Brazil, May, pp.132-141.
- [29] Wang Y., Court, I., Ross, M., Staples, G., King, G. and Dorling, A. (1999), *Towards Software Process Excellence: A Survey Report on the Best Practices in the Software Industry*, *ASQ Journal of Software Quality Professional*, Vol.2, No.1, Dec., pp. 34-43.
- [30] Wang, Y., Dorling, A., Brodman, J. and Johnson, D. (1999), *Conformance Analysis of the Tailored CMM with ISO/IEC 15504*, Proceedings of International Conference on Product Focused Software Process Improvement (PROFES.99), Oulu, Finland, June, pp. 237-259.
- [31] Wang Y., King, G., Dorling, A., Patel, D., Court, I., Staples, G. and Ross, M. (1998), *A Worldwide Survey on Software Engineering Process Excellence*, Proceedings of IEEE 20th International Conference on Software Engineering (ICSE.98), Kyoto, April, IEEE Press, pp.439-442.
- [32] Wang, Y., I. Court, M. Ross, G. Staples, G. King and A. Dorling (1997), *Quantitative Analysis of Compatibility and Correlation of the Current SPA Models*, Proceedings of The IEEE International Symposium on Software Engineering Standards (IEEE ISESS.97), USA, June, pp. 36-56.
- [33] Wang, Y., I. Court, M. Ross, G. Staples, G. King and A. Dorling (1997), *Quantitative Evaluation of the SPICE, CMM, ISO9000 and BOOTSTRAP*, Proceedings of the IEEE International Symposium on Software Engineering Standards (IEEE ISESS.97), USA, June, pp. 57-68.
- [34] Wang, Y., I. Court, M. Ross and G. Staples (1996), *Towards a Software Process Reference Model (SPRM)*, Proceedings of International Conference on Software Process Improvement (SPI.96), Brighton, UK, November, pp.145-166.
- [35] Fayad, M.E. and Laitinen, M. (1997), *Process Assessment: Considered Wasteful*, Communications of the ACM, Vol.40, No.11, Nov.
- [36] Humphrey, W.S. and Curtis, B. (1991), *Comment on .a Critical Look.*, IEEE Software, Vol.8, No.4, pp.42-47.
- [37] Marquardt91 Donald Marquardt, et al. "Vision 2000: The Strategy for the ISO 9000 Series Standards in the '90s." ASQC Quality Progress, Vol. 24, No. 5, May 1991, pp. 25-31.
- [38] Royce ,walker vice president and general manager of strategic services of rational software corporation ,the rational edge, CMM vs CMMI from conventional to modern software management Feb. 2002.
- [39] Mary Beth Chrissis, Mike Konrad, Sandy Shrum. "CMMI: Guidelines for Process Integration and Product Improvement". 2003. Addison Wesley Professional
- [40] Software Engineering Institute (SEI). "CMMI version 1.1 CMU/SEI-2002-TR- 012". 2002.
- [41] Richard Basque. "CMMI un itin'eraire fl'ech'e vers le Capability Maturity Model Integration". 2004. Dunod.
- [42] B. Mutafelija and H. Stromberg, *Systematic Process Improvement Using ISO 9001:2000 and CMMI*, Artech H.
- [43] M. B. Chrissis, M. Konrad and S. Shrum, *CMMI Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2003.