BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
Vol. 1, Issue 2, Nov 2013, 13-20

EFFICACY OF ISO 9000 STANDARDS CONVERSELY CMM FOR A SOFTWARE ORGANIZATION

MONIKA YADAV[1] & KAUSHIK KUMAR[2]

[1]Research Scholar, Department of Management, Birla Institute of Technology, Mesra, Ranchi, India

[2]Associate Professor, Department of Mechanical Engineering, Birla Institute of Technology, Mesra, Ranchi, India

ABSTRACT

The two most common process models in use today for software engineering are the Capability Maturity Model (CMM), and the International Organization for Standards (ISO) ISO 9000 standard. To embezzle the classic "To be or not to be" phrase of the ancient thespian Shakespeare, "To CMM, or ISO--that is the question". Indeed, choosing a process improvement framework is a daunting prospect for the uninitiated. ISO 9000 series of standards, developed by the International Standards Organization, and the Capability Maturity Model for Software (CMM), developed by the Software Engineering Institute, share a common concern with quality and process management.

This paper has analyzed the evolution of software engineering into the complex and challenging discipline that it is today; determined the difference between a lifecycle model and a process model--and discovered how a process model benefits a software development organization. Two of the most common process models in use where briefly compared and contrasted--the CMM and the ISO.

KEYWORDS: Software Industry, ISO 9000 Family of Standards, Capability Maturity Model (CMM), Mapping of ISO 9000 and CMM

INTRODUCTION

Quality Management System

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. 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 signalling 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. Figure 1 depicts quality management system for those organization who are continuously striving for continual improvement.

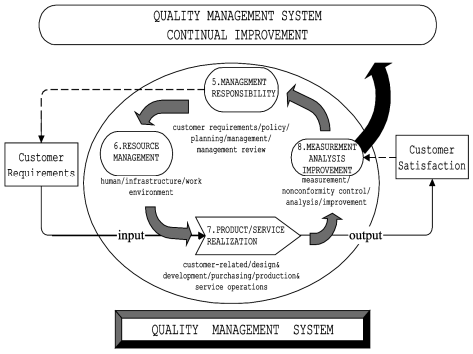

Figure 1: CMM Maturity Levels

Software Industry

Survival in a competitive environment has always been a challenge throughout the history of the world which is also true in the industrial scenario. 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. 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.

There are a huge number of software standards, methodologies, practices, models and guidelines imported in to the current era of software engineering. These standards usually tend to be tailor made and hence fit all approach that may be optimum for some projects but is often times ill-suited for others [1]. This is because they are continually improving which has become a complicated exercise for software industries, however, many companies developed internal standards based on the military standards, and then improve the process as their software development processes matured. The software development systems based on these internal, commercial standards, and improved over the years have proved to be good systems [2]. In this paper the two dominant process improvement models in use today namely, Capability Maturity Model and the ISO 9000 standard, are illustrated, contrasted, and analyzed for applicability to software development environments. The focus is on the ISO guidelines in areas that are most relevant to software engineering.

Some general conclusions are developed as to the applicability of each of the process model standards for different types of software development organizations and business environments.

ISO 9000 STANDARDS

The standards are published by ISO, the International Organization for Standardization concern for quality and process management and available through National standards Bodies. The ISO 9000 family of standards is related to quality management system 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 ISO 9000 deals with the fundamentals of Quality management systems, including the eight management principles on which the family of standards is based. The specific standard in the ISO 9000 series of concern to design and development organizations e.g. software organization is ISO 9001. ISO 9001 [3-5] deals with the requirements that organizations wishing to meet the standard have to fulfil. Third party certification bodies provide independent confirmation that organizations meet the requirements of ISO 9001. Over a million organizations worldwide are independently certified, making ISO 9001 one of the most widely used management tools in the world today. Despite widespread use, however, the ISO certification process has been

criticized as being wasteful and not being useful for all organizations.

CAPABILITY MATURITY MODEL (CMM)

The capability maturity model (CMM) was the first process methodology that tried to model software engineering process systems. CMM [6-12] 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 [7-8] and the SEI capability maturity model (CMM) for software [9-12]. 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 [9] with the supplement of a set of detailed key practices [10], and a questionnaire by Zubrow et al. [13].

A summary of differences between Version 1.0 and Version 1.1 was provided by Paulk et al. [11]. Humphrey and other researchers have also contributed towards the development CMM [6-8]. 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 [14-26]. Like any system or process CMM also has shortcomings which were studied by scholars [30-34]. 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 1.

Table 1: CMM Maturity Levels

Maturity Level	Description
1-Initial	The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics
2-Repeatable	Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
3-Defined	The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software
4-Managed	Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled
5-Optimized	Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies

MAPPING OF ISO 9000 AND CMM

The development of ISO was done in an era dominated by manufacturing industries and software organizations then were in budding stage. Hence the standard was intentionally written for Hard-goods manufacturing industries and not software industry. In contrast, the CMM framework was created from the ground-up to be specific to the software industry. So there is difference in the approach and outcome of the two. The basic essence of ISO 9000's concept is to follow a set

of standards to make success repeatable on the contrary CMM emphasizes a process of continuous improvement towards quality. Once an organization has met the criteria to be ISO certified by an independent auditing authority, the next step is only to maintain that level of certification where as the CMM is an on-going process of evaluation and improvement, moving from one level of achievement to the next. Even at the highest level of maturity in CMM, the focus is still on continuous improvement. In ISO 9001 there are 20 clauses through which one can compare CMM in order to evaluate its efficacy. There is a lot of analysis involved while doing this comparison, and there are differences in interpretation for both ISO 9001 and the CMM. A common challenge in ISO based certification and CMM based appraisal is reliability and consistency of assessment which is partially addressed by strict training prerequisites for CMM appraisers and others too. The basic similarity between them is to "say what you do, do what you say". Table 2 illustrates the mapping between the two standards and also gives a path for a software organization to switch from ISO to CMM with minimum alterations.

Table 2: Mapping Between ISO 9000 and CMM with Reference to 20 Clauses of ISO

Serial no	ISO 9000	CMM
1.Management Responsibility	In this a designated manager ensures that the quality Policy is implemented and maintained i.e defined, documented understood as per the given standard	In this they have responsibility for quality policy and verification activities which is primarily addressed in Software Quality Assurance.
2.Quality system	It requires that a documented quality system, including procedures and instructions, is to be established.	Quality system activities are primarily addressed in this in Software Quality assurance.
3.Contract review	ISO 9001 requires that contracts be reviewed to determine whether the requirements are adequately defined, agreed with the bid, and can be implemented.	Review of the customer requirements, as allocated to software, is described in the CMM in Requirements Management.
4.Design control	ISO 9001 requires that procedures to control and verify the design be established. purchaser's requirements specification	the life cycle activities of requirements analysis, design, code, and test are described under Software Product Engineering.
5.Document control	ISO 9001 requires that the distribution and modification of documents be controlled	The documentation usually done under Software Configuration Management
6.Purchasing	ISO 9001 requires that purchased products conform to their specified requirements	This is addressed in Software Subcontract Management
7.Purchaser supplier product	ISO 9001 requires that any purchaser-supplied material be verified and maintained.	They are not well addressed in this.
8.Product identification and traceability	ISO 9001 requires that the product be identified and traceable during all stages of Production, delivery, and installation.	The CMM covers this clause primarily in Software Configuration Management.
9.Process control	ISO 9001 requires that production processes be defined and planned	It is generally done under this Software Project Planning.
10.Inspection and testing	ISO 9001 requires that incoming materials be inspected or verified before use and that in process inspection and testing be performed.	It is carried out under the Software Product Engineering.
11.Inspection ,measuring, and test equipment	ISO 9001 requires that equipment used to demonstrate conformance be controlled, calibrated, and maintained. When test hardware or software is used, it is checked before use and rechecked at prescribed intervals.	It is carried out under the testing practices in Software Product Engineering.
12.Inspection and test status	ISO 9001 requires that the status of inspections and tests be maintained for items as they progress through various processing steps.	It is carried out under the Software Product Engineering.

Table 2: Contd.,

13.Control of non-conforming product	ISO 9001 requires that nonconforming product be controlled to prevent inadvertent use or installation.	Installation is not addressed in the CMM .rest is carried out i.e Design, implementation, testing, and validation are addressed in Software Product Engineering.
14.Corrective action	ISO 9001 requires that the causes of nonconforming product be identified so that corrective action to be taken.	This would be addressed in Software Quality Assurance in the CMM.
15.Handling, Storage, Packaging, and Delivery	ISO 9001 requires that procedures for handling, storage, packaging, and delivery be Established and maintained.	They are not well addressed in this.
16.Quality Records	ISO 9001 requires that quality records be collected, maintained, and dispositioned.	The practices defining the quality records to be maintained in the CMM are distributed Throughout the key process areas in the various Activities Performed practices.
17.Internal Quality Audits	ISO 9001 requires that audits be planned and performed. The results of audits are communicated to management, and any deficiencies found are corrected.	The auditing process is described in Software Quality Assurance.
18.Training	ISO 9001 requires that training needs be identified and that training be provided, since selected tasks may require qualified personnel. Records of training are maintained.	Specific training needs in the CMM are identified in the training and orientation practices in the Ability to Perform common feature.
19.Servicing	ISO 9001 requires that servicing activities be performed as specified.	CMM is intended to be applied in both the software development and maintenance environments, the practices in the CMM do not directly address the unique aspects that characterize the maintenance Environment.
20.Statistical Techniques	ISO 9001 states that, where appropriate, adequate statistical techniques are identified and used to verify the acceptability of process capability and product characteristics.	The practices describing measurement in the CMM are distributed throughout the key process areas.

CONCLUSIONS

From above it is quite clear that there is a strong correlation between ISO 900 and the CMM, even some issues in ISO 9001 are not covered in the CMM, and vice versa. The biggest difference, between these two documents is the emphasis of the CMM on continuous process improvement but ISO 9001 addresses the minimum criteria for an acceptable quality system. It can also be noted that the CMM focuses strictly on software, while ISO 9001 has a much broader scope: hardware, software, processed materials, and services [32]. Hence the Capability Maturity Model is better suited to software organizations that are currently using, or plan to implement, an iterative lifecycle. The organization get benefit in terms of customer relations and market status by becoming ISO 9001 certified. It will be better positioned to accommodate technology evolution by embracing the CMM.

REFERENCES

1. Software Quality Professional: VOL. 8, Issue Two/© March 2006, And ASQ publication: Page.28, “Are Standards the Answer?”
2. Handbook of Software Quality Assurance edited by G. Gordon Schulmeyer and James I. McManus. –

3rd ed-1998. : Page.104, “Standardization of Software Quality Assurance – Conclusion.

3. "ISO 9000-1: 1994", Quality management and quality assurance standards, Part 1: Guidelines for selection and use,(ISO,Geneva 1994).
4. "ISO 9001: 1994", Quality systems. Model for quality assurance in design, development, production, installation and servicing,(ISO,Geneva 1994)
5. "ISO 9002: 1994", Quality systems. Model for quality assurance in production, installation and servicing, (ISO,Geneva 1994).
6. 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.
7. Humphrey, W.S. (1988), Characterizing the Software Process: A Maturity Framework, IEEE Software, March, pp.73-79.
8. Humphrey, W.S. (1989), Managing the Software Process, Addison-Wesley Longman, Reading, MA.
9. 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.
10. 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.
11. 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.
12. 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.
13. Zubrow, D. (1997), the Software Community Process Maturity Profile, Software Engineering Institute, Pittsburgh.
14. Humphrey, W.S., Snyder, T.R. and Willis, R.R. (1991), Software Process Improvement at Hughes Aircraft, IEEE Software, July, pp.11-23.
15. 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.
16. Saiedian, H. and Kuzara, R. (1995), SEI Capability Maturity Model.s Impact on Contractors, IEEE Computer, Vol.28, No.1, pp.16-26.
17. Paulk, M.C. (1995), how ISO 9001 compares with the CMM, IEEE Software, January, pp.74-83.
18. 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.
19. Wang, Y. and G. King (2000), Software Engineering Processes: Principles and Applications, CRC Press, USA, April, ISBN: 0849323665, pp.1- 752.
20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.'

27. Bollinger, T.B. and McGrowan, C. (1991), A Critical Look at Software Capability Evaluations, IEEE Software, July, pp. 25-41.

28. Brodman J.G. and Johnson D.L. (1994), What Small Business and Small Organization Say about the CMM, Proceedings of the 16th International Conference on Software Engineering (ICSE16), pp.331-340.

29. Fayad, M.E. (1997), Software Development Process: the Necessary Evil? Communications of the ACM, Vol.40, No.9, Sept.

30. Fayad, M.E. and Laitinen, M. (1997), Process Assessment: Considered Wasteful, Communications of the ACM, Vol.40, No.11, Nov.

31. Humphrey, W.S. and Curtis, B. (1991), Comment on .a Critical Look., IEEESoftware, Vol.8, No.4, pp.42-47.

32. 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.