IDENTIFICATION OF EFFECTIVE KEY PROCESSES IN SOFTWARE PROCESS IMPROVEMENT MODELS FOR SMES

Ashima*

Dr. Himanshu Aggarwal**

ABSTRACT

Software Process Improvement (SPI) is recognized as a new era in modern software development or software engineering. SPI actually finds its actual place in software process re-engineering and giant software organizations are heading towards SPI in search of excellence. Only self sufficient and extensively matured organizations think to afford SPI initiatives. It seems miserable that these SPI initiatives are not a cup of tea for Small and Medium Scale Enterprises (SMEs) as Indian software industry composes a large chunk of SMEs contributing to nation's economy. Existing software process improvement standards were not written for SMEs. Scarcity of resources, budget and time hampers SMEs to implement expensive software process improvement initiatives. A large number of universities, industrial research centres, and associations have tried to find their own answers to the issue being faced by most SMEs, and are proposing dedicated software process models for SMEs. However, no one has been able to propose a one-size-fits-all SPI solution for SMEs. In this paper, more than a dozen SPI models are analysed. And the intent is to find a customized solution for SPI in SMEs. Here, we try to identify important key processes which directly or indirectly have impact on software process improvement in SMEs.

*Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala.

**Associate Professor, Faculty of Computer Engineering, Punjabi University, Patiala.

1. INTRODUCTION

The crux of improvement and attaining excellence lies in how we follow best terminology and execute correct methodology. Similarly, SMEs require exact set of software processes in their SPI program that can improve software processes continuously and thereby can act as catalyst in organizational maturity. SMEs in indian software industry are developing large amount of different kind of software. But they don't exercise software process optimization in order to improve their software processes. Here, a number of different software process models are surveyed and discussed to find out the inherent characteristics of common key processes. The common key processes are further drilled in order to identify effective set of software processes for SMEs.

2. SOFTWARE PROCESS IMPROVEMENT MODELS

2.1. Capability Maturity Model (CMM): Mother of all SPI Models, The Capability Maturity Model (CMM) is proposed by the US Software Engineering Institute (SEI) in 1986. CMM focuses on the improvement of the software development process, and uses effective, organized management and plans to continually improve and enhance the quality of software products. There exist 18 key process areas (KPAs) in CMM consisting of 5 maturity levels [12].

Maturity Levels	Key Process Areas
5.Optimizing	Process Change Management(PSP,TSP) Technology Change Management(PSP,TSP) Defect Prevention(PSP,TSP)
4.Managed	Software Quality Management(PSP,TSP) Quantitative Process Management(PSP,TSP)
3.Defined	Peer Reviews(PSP,TSP) Intergroup Coordination Software Product Engineering(PSP,TSP) Integrated Software Management(PSP,TSP) Training Program Organization Process Definition(PSP,TSP) Organization Process Focus(PSP,TSP)
2.Repeatable	Software Configuration Management(TSP) Software Quality Assurance(TSP) Software Subcontract Management

	Software Project Tracking and Oversight(PSP,TSP) Software Project Planning(PSP,TSP) Requirement Management(TSP)
1.Initial	

Table 1: Key Process Areas and Maturity Levels of CMM [12]

2.2. Personal Software Process (PSP): The Personal Software Process (PSP) is a software process developed at the Software Engineering Institute (SEI).The PSP was released in the year 1995. The PSP is a technique that engineers can use to improve the predictability, quality, and productivity of their work.PSP addresses the need for process improvement in small organizations and small project teams.PSP follows the concepts of the CMM. There exist 12 key process areas (KPAs) in PSP. The key process areas are described in Fig 1. [22].

2.3. Team Software Process (TSP): The initial version of the TSP was developed and piloted by Watts Humphrey. It was released in the year 1999.The goal of TSP is to help engineering teams develop and deliver high-quality software-intensive systems within planned cost and schedule commitments. There exist 17 key process areas (KPAs) in TSP[23].

2.4. Capability Maturity Model Integration (CMMI): CMMI was proposed by Software Engineering Institute (SEI) and was introduced in 2002It integrates capability models for software development, systems engineering, and integrated product and process development[10]. CMMI representation can be Staged or Continuous. Staged representation supports the easy migration from CMM to CMMI. The continuous representation is designed to allow the user to focus on the specific processes that are considered important for the organization's immediate business objectives or those to which the organization assigns a high degree of risk. There exist 22 key process areas (KPAs) in CMMI [10].

2.5. Six-Sigma (SS): Six-Sigma is represented as "6σ".Its main objective is elimination of defects, waste, or quality control problems. Six-Sigma was originated at Motorola in the early 1980's. The name Six-Sigma refers to the capability of the process to deliver units within the set limits .An important part of Six-Sigma is the DMAIC and DMADV procedure. There exist 14key process areas (KPAs) in Six-Sigma which is staged representation of CMMI[7].

2.6. P-CMM: P-CMM stands for People Capability Maturity Model. P-CMM was developed by SEI and was introduced in 1995.The main aim of P-CMM is to enhance organization capabilities and effectiveness in human resource activities of software organizations to attract, develop, motivate, organize, and retain the talent needed to continuously improve software development capability [4]. The activities that make up the P-CMM framework start with focusing on individual/ unit level and proceed to include group/ organization alignment only after the processes for individual/ unit level development have been carried out. There exist 20 key process areas (KPAs) in P-CMM [4].

2.7. Malcolm Balridge National Quality Award: The Malcolm Balridge National Quality Award was established by the Malcolm Baldrige National Quality Improvement Act of 1987.It recognizes U.S. organization s in the business, health care, education, and non-profit sectors for performance excellence. The award promotes awareness of performance excellence as an increasingly important element in competiveness. The Malcolm Baldrige Award provides complementary perspective for executives to break the code of predictability of cost, schedule, performance, and customer satisfaction [16].

The following seven areas are evaluated for the prestigious award:

- Leadership,
- Information and Analysis,
- Strategic Planning,
- Human Resource Development and Management,
- Process Management,
- Business Results,
- Customer Focus and Satisfaction.

2.8. SE-CMM: SE-CMM stands for System Engineering Capability Maturity Model. The Systems Engineering CMM effort began in August 1993 in response to industry requests for assistance in coordinating and publishing a model for the systems engineering community analogous to the CMM for Software [19].

The main goal of SE-CMM is to improve system or product engineering process. The SE-CMM is becoming a reference model for assessing current practices, planning, implementing, and measuring process improvement efforts and determining the business results of such efforts. There exist 17 process areas (PAs) in SE-CMM divided into three categories. They are Engineering, Project and Organizational process areas [19].

2.9. SSE-CMM: SSE-CMM stands for System Security Engineering Capability Maturity Model. SSE-CMM v1.0 was published in 1996.In 2002, SSE-CMM was approved as ISO/IEC 21827.[5]. It is focussed upon the requirements for implementing security in a system or series of related systems that are the Information Technology Security (ITS) domain. However, experience with the Model has demonstrated its utility and applicability to other security domains other than the IT domain. There exist 22 process areas (PAs) in SSE-

CMM. The process areas are divided into two categories. They are Security Engineering and Project and Organizational process areas [5] as shown in Table 2.

Security Engineering Process Areas	Project and Organizational Process Areas
Administer Security Needs, Assess Impact, Assess Security Risk, Assess Threat, Assess Vulnerability, Build Assurance Argument, Coordinate Security, Monitor Security Posture, Provide Security Input, Specify Security Needs, Verify and Validate Security.	Ensure Quality, Manage Configuration, Manage Project Risk, Monitor and Control Technical Effort, Plan Technical Effort, Define Organization's Security Engineering Process, Improve Organization's Security Engineering Process, Manage Product Line Evaluation, Manage System Engineering Support Environment, Provide Ongoing Skills and Knowledge, Coordinate with Suppliers.

Table 2: Process Areas of SSE-CMM[5]

2.10. ISO 9000: The ISO 9000 family of standards had been developed to assist organizations, of all types and sizes, to implement and operate effective quality management systems. Most of the practices required by ISO 9000 have been in use in industries of various kinds for decades [20].ISO 9000 contains 35 process areas (PAs).It contains the process areas of CMM and SE-CMM [6].

2.11. Trillium Model: Trillium has been developed by a consortium of telecommunications companies headed by Bell Canada. Trillium has also been designed to be applied to embedded software systems such as telecommunications systems. It combines the requirements from the ISO 9000 series, the CMM and the Malcolm Balridge National Quality Award criteria [20].

The goal with the Trillium model is to help organizations to start and conduct a process improvement program on an existing process or lifecycle. Trillium model is based on roadmaps rather than process areas. There exist 28 roadmaps in trillium model[20].

2.12. SPICE: SPICE stands for Software Process Improvement Capability dEtermination.ISO/IEC 15504 was the result of the SPICE project. SPICE was developed in the year 1995. Process improvement has the objective of changing or optimizing processes for greater effectiveness to achieve gains in product quality and productivity. Capability determination however is concerned with assessing an organization or project in order to determine risks to the successful outcome of a contract, development or service delivery. There exist 20 key process areas (KPAs) in SPICE [1].

2.13. ISO/IEC 12207: ISO/IEC 12207 was published on 1 August,1995 .Its goal is to provide a comprehensive set of life cycle processes, activities and tasks for software that is part of a larger system, and for standalone software products or services[18]. In ISO/IEC 12207 the project provides the context for describing processes concerned with planning, assessment and control. There exist 17 process areas (PAs) in this method. The process areas are divided into three categories Primary, Supporting and Organizational [18].

2.14. BOOTSTRAP:BOOTSTRAP methodology was initially developed in an European industry. It was specially developed to speed up the application of software engineering technology in the European software industry In the BOOTSTRAP methodology the starting point is to use the methodology for software process improvement [17]. The objective of the BOOTSTRAP methodology is to provide support to the evaluation of process capability against a set of recognised software engineering best practices. There exist 37 process areas (PAs) in BOOTSTRAP. It integrates process areas from CMM, ISO 9000, ISO/IEC 12207 and ISO/IEC 15504 [17].

2.15. SECAM: SECAM stands for Systems Engineering Capability Assessment Model. The most recent Version 1.5 of the Systems Engineering Capability Assessment Model was released in July of 1996 [9].

SECAM is focussed to assess an organizations systems engineering capability for integrated systems and integrated product and process development (IPPD) teams and determine areas for improvement. SECAM consist of 19 process areas (PAs).The process areas of SECAM are divided into three categories: Management, Organization and Systems Engineering [9] as shown in Table 3.

1. Management	**2. Organization**	**3.Systems Engineering**
1.1Planning 1.2.Tracking and Oversight 1.3.Sub-Contract Management	2.1.Process Management and Improvement 2.2.Competency	3.1.System Concept Development 3.2.Requirement and

1.4.Inter-group Coordination 1.5.Configuration Management 1.6.Quality Management 1.7.Risk Management 1.8.Data Management	Development 2.3.Technology Management 2.4.Environment and Tool Support	Functional Analysis 3.3.System Design 3.4.Integrated Engineering Analysis 3.5.System Integration 3.6.System Verification 3.7.System Validation

Table 3: Process Areas of SECAM [9]

2.16. SDCE: SDCE stands for Software Development Capability Evaluation. The SDCE was initiated in August 1992 and was developed by a team of Air Force, industry, and Federally Funded Research and Development Center participants [20]. Its primary objective is to reduce development risk by selecting contractors who are capable of successfully developing embedded software to meet the program lifecycle requirements .Another SDCE objective is to address software engineering and directly related systems engineering and development disciplines that are critical elements of successful software-intensive systems development [20].

2.17. MIL-STD-498, ANSI/EIA 632, and IEEE 1220:MIL-STD-498 stands for Military-Standard-498.It was a United States military standard. It was released in Nov, 1994. The main objective is to establish uniform requirements for software development and documentation. There exist 26 process areas (PAs) in MIL-STD-498 [15].

MIL-STD-498A was released after MIL-STD-498.Then MIL-STD-498B was developed which was never released. From MIL-STD-498B, two new models were developed and released. They are ANSI/EIA 632 and IEEE 1220 [15]. ANSI/EIA 632 was intended to provide a framework for developing and supporting a universal SE discipline for both defence and commercial environments. There exist 13 process areas (PAs) in ANSI/EIA 632 [11].IEEE 1220 was established to provide a standard for managing a system. There exist 14 process areas (PAs) in IEEE 1220 [8].

2.18. ISO/IEC 15288: ISO/IEC 15288 was established in the year 2004 by the combination of ANSI/EIA 632 and IEEE 1220 and was released in 2008. The purpose of this International Standard is to provide a defined set of processes to facilitate communication among acquirers, suppliers and other stakeholders in the life cycle of a system [20]. ISO/IEC 15288concerns those systems that are man-made and may be configured with one or more of the following: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials and naturally occurring entities. There exist 22 process areas in ISO/IEC 15288 [20].

2.19. TickIT: TickIT was introduced in early 1990's. The TickIT certification scheme was developed by the Department of Trade and Industry (DTI) and the British Computer Society (BCS) with strong support of UK software suppliers and is published in1993 [21]. TickIT is based on three important concepts. They are accredited certification in the IT/software sector i.e. certification bodies are specifically accredited to practice in the IT sector, use of specially registered auditors i.e. the auditors have to pass a special training course prove that they have worked in a software development environment and attend a three-person interview panel as a final check on their competence and the use authoritative guidance material i.e. the TickIT guide which includes the European IT Quality System Auditor Guide. There exist 33 process areas (PAs) in TickIT [21].

2.20. TRISO : TRISO stands for TRidimensional Integrated Software. The TRISO-Model, presenting a 3-D integrated software engineering methodology, was first proposed in the SPW 2005 held in Beijing, China (Li 2005). It expands a possible direction of Integrated Software Engineering. The TRISO-Model classifies the essential elements of software development processes into three dimensions: SE (Software Engineering) Technology, SE Process and SE Human [14].

3. IDENTIFICATION OF EFFECTIVE KEY PROCESSES IN SPI MODELS FOR SMES

There is an acute need of identifying a set of effective software processes that can be included in dedicated Software Process Models for SMEs. Twenty different software process improvement models are surveyed on 157 process areas and 43 Key process areas identified as effective process area for SPI in SMEs. Effectiveness is based upon the higher number of occurrence of a software process in different SPI models. Table 4 also shows key processes which are common to more than one improvement model.

SNo	Key Processes	Software Improvement Models containing the specified Process Area
1.	Requirement Management	CMM, CMMI, TRISO, TSP, SDCE, IEEE 1220, Six-Sigma and ISO 9000.
2.	Software Project Planning	CMM, PSP, TSP and ISO 9000.
3.	Software Project	CMM, PSP, TSP and ISO 9000.

	Tracking and Oversight	
4.	Software Subcontract Management	CMM and ISO 9000.
5.	Software Quality Assurance	CMM, TSP, SDCE, MIL-STD-498 and ISO 9000.
6.	Software Configuration Management	CMM, TSP, SDCE, MIL-STD-498 and ISO 9000.
7.	Organization Process Focus	CMM, CMMI, TRISO, PSP, TSP, Six-Sigma and ISO 9000.
8.	Organization Process Definition	CMM, CMMI, PSP, TSP, Six-Sigma, ISO 9000 and TRISO.
9.	Training Program	CMM, TSP and ISO 9000.
10.	Integrated Software Management	CMM, PSP, TSP and ISO 9000.
11.	Software Product Engineering	CMM, PSP, TSP and ISO 9000.
12.	Intergroup Coordination	CMM, TSP, SECAM, SDCE and ISO 9000.
13.	Peer Reviews	CMM, PSP, TSP, SDCE and ISO 9000.
14.	Quantitative Process Management	CMM, PSP, TSP and ISO 9000.
15.	Software Quality Management	CMM,PSP,TSP,SDCE and ISO 9000
16.	Defect Prevention	CMM,PSP,TSP and ISO 9000
17.	Technology Change Management	CMM,PSP,TSP and ISO 9000
18.	Process Change Management	CMM,PSP,TSP and ISO 9000
19.	Project Monitoring and Control	CMMI, TRISO and Six-Sigma

20.	Supplier Agreement Management	CMMI, TRISO and Six-Sigma
21.	Measurement and Analysis	CMMI, TRISO IEEE 1220, Six-Sigma and TickIT.
22.	Process and Product Quality Assurance	CMMI, TRISO and Six-Sigma.
23.	Requirement Development	CMMI, TRISO, IEEE 1220 and Six-Sigma.
24.	Technical Solution	CMMI, TRISO, IEEE 1220 and Six-Sigma.
25.	Product Integration	CMMI, TRISO, IEEE 1220 and Six-Sigma.
26.	Verification	CMMI, TRISO, ISO/IEC 12207, IEEE 1220, Six-Sigma, SPICE, BOOTSTRAP and TickIT.
27.	Validation	CMMI, TRISO, ISO/IEC 12207, IEEE 1220, Six Sigma, SPICE, BOOTSTRAP and TickIT.
28.	Organizational Training	CMMI, TRISO and Six-Sigma.
29.	Integrated Project Management	CMMI, TRISO, IEEE 1220 and Six-Sigma.
30.	Risk Management	CMMI, TRISO, MIL-STD-498, IEEE 1220,ISO/IEC 15288, SPICE, BOOTSTRAP and Six-Sigma.
31.	Integrated Teaming	CMMI, TRISO and IEEE 1220.
32.	Integrated Supplier Management	CMMI, TRISO and Six-Sigma
33.	Decision Analysis and Resolution	CMMI, TRISO, IEEE 1220 and Six-Sigma.
34.	Organizational Process Performance	CMMI, TRISO and Six-Sigma.
35.	Quantitative Project Management	CMMI, TRISO and Six-Sigma.
36.	Organizational Innovation and Deployment	CMMI, TRISO and Six-Sigma.
37.	Causal Analysis and	CMMI, TRISO and Six-Sigma.

	Resolution	
38.	Training	P-CMM, SDCE and ISO/IEC 12207.
39.	Competency Development	P-CMM and SECAM
40.	Technology Management	SECAM and Trillium Model
41.	Configuration Management	CMMI, TRISO Trillium Model, ISO/IEC 12207,IEEE 1220,SPICE and Six-Sigma
42.	Reuse	Trillium Model, SPICE and SDCE
43.	Sub-Contract Management	Trillium Model and SDCE

Table 4: Effective Software Key Process

4. CONCLUSION

The effectiveness of SPI processes can be contributed to commonality of the specific process in different Software Process Models. Software processes or software process areas common in many software process models means that software organization cannot afford to discard that particular key process from their SPI initiative. As there is a need to tailor an efficient as well as effective SPI model, identification of effective key processes will contribute as a building block to the required SPI model for SMEs. Further, these identified software processes need to be studied for their cost and time consumption, so that a low priced edition of SPI model can be proposed.

5. REFERENCES

1. A. Dorling, "SPICE: Software process improvement and capability determination," *Software Quality Journal*, vol. 2, no. 4, pp. 209–224, Dec, 1993.
2. Ashima and Himanshu Aggarwal, "Essence of Software Process Reengineering in SME's: Towards Process Customization and Automation*," Global Journal of Enterprise Information System*,vol-2 issue-2,pp.95-105, July 2010.
3. A. Sweeney and D.W.Bustard," Software process improvement: Making it happen in practice",*SoftwareQuality Journal*, vol.**6**, No.4, pp. 265–273, May 1997.
4. B. Curtis, W. Hefley and S. Miller, *People Capability Maturity Model*. Addison-Wesley: Boston, MA, USA, 2001.
5. Cui Bao-Ling ,"System Security Engineering Capability Maturity Model",*Journal of the Harbin Institute of Technology* , vol. 35,no. 3,pp.293-297,2002.

6. D. Stelzer, W. Mellis and G. Herzwurm, "A critical look at ISO 9000 for software quality management," *Software Quality Journal*, vol. 6, no. 2, pp. 65–79, 1997.
7. F.C Herold III,P. Carden and K. Stephens ,'Six Sigma', *Six Sigma and Related Studies in the Quality Disciplines. The Best on Quality Book Series of the International Academy for Quality*, Vol. 14, Chapter 2, pp.15–34, 2003.
8. IEEE 1220-1994. *IEEE Trial-Use Standard for Applicationand Management of the Systems EngineeringProcess, IEEE Computer Society, Institute of Electricaland Electronics Engineers*, February 28, 1995.
9. INCOSE Capability Assessment Working Group, *Systems Engineering Capability Assessment Model*, Version 1.50, June 1996.
10. L.G. Jones ,*Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice,* US: SEI,Carnegie Mellon University.2002
11. Martin and N. James*, "Overview of the EIA 632 Standard :Process for Engineering a System," Eighth Annual International Symposium of INCOSE,* Vancouver, British Columbia, July 26-30, 1998.
12. M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, "Capability Maturity Model, Version 1.1," *IEEE Software*, vol. 10, no. 4, pp. 18-27, July 1993.
13. M. Chrissis, M. Konrad and S. Shrum.*CMMI: Guidelines for Process Integration and Product Improvement*.MA: AddisonWesley, Boston, 2003 (ISBN 0321154967).
14. MingshuLi,"TRISO-Model: A New Approach to Integrated Software Process Assessment and Improvement", *Software Process Improvement and Practice,* vol.12, issue.5, pp.387-398, 2007.
15. *MIL-STD-498, Military Standard Software Development and Documentation,* Department of Defense, December 5, 1994.
16. MO.Tingey ,*Comparing ISO 9000, Malcolm Baldridge, and the SEI CMM for Software*. Prentice Hall PTR, NJ, 1996
17. P. Kuvaja, J. Simila¨, L. Krzanik, A. Bicego, G. Koch and S. Saukkonen. *Software Process Assessment and Improvement:The BOOTSTRAP Approach*. Blackwell Publishers, 1994.
18. Raghu Singh, "International Standard ISO/IEC 12207 Software Life Cycle Processes",*Software Process Improvement and Practice*, vol. 2, pp.35-50, 1996.
19. Roger Bate et. al., *A Systems Engineering Capability Maturity Model,* Version 1.1, SoftwareEngineering Institute, Carnegie Mellon University, November 1995.

20. Sarah A. Sheard, “The Frameworks Quagmire,” *Crosstalk: The Journal of Defense SoftwareEngineering*, vol. 10, no. 9, September 1997.
21. T.J. Hall,” The Quality Systems Manual:The Definitive Guide to the ISO 9000 Familiy and TickIT”,*Journal of Operational Research Society*,vol.48,no.1,pp.105,January,1997.
22. Watts S. Humphrey, “The Personal Software Process: Status and Trends”, *IEEE Software*, vol.17, no.6, pp.71-75, Dec 2005.
23. Watts S. Humphrey, *Introduction to the Team Software Process*. Reading, MA: Addison-Wesley Publishing Company, Inc., 2000.