Global Journal of Enterprise Information System
July 2010 -Dec 2010
Volume-2 Issue-2


# Essence of Software Process Reengineering in SME's: Towards Process Customization and Automation

**Ashima**
*Department of Computer Science and Engineering,*
*Thapar University, Patiala*
*ashima@thapar.edu*

**Himanshu Aggarwal**
Department of Computer Engineering,
Punjabi University, Patiala.
himanshu.pup@gmail.com

## ABSTRACT

Software Process Reengineering is the core kernel of software process improvement (SPI). Today, the very promising small scale software enterprises (SME's) are striving for standardization of their software development processes. They are carrying out improvement processes but not a proper set of processes. Lack of experienced process engineers and activities which lead them to have a good CMM level is a big issue to be resolved. There are no underlying problems for large scale software industries due to enough resources like available budget, trained as well as experienced and dedicated software professional's team for software process improvement programs, required set of efficient tools and technology for actual implementation. And top of the all required infrastructure, proper understanding and mindset for applying software reengineering initiative. Unfortunately, there is limited adoption, absorption, adaptation and assimilation of software process improvement models in SME's due to lack of know-how and available resources in terms of money, time, perceived benefits, quality focus. Unavailability of required automated tool sets, in-house software process assessment. In nutshell, there is limited adoption, absorption, adaptation and assimilation of software process improvement models in SME's due to lack of know-how and available resources in terms of money, time and perceived benefits. This paper explores as SPI has evolving nature, there is a need of Cost-effective framework which provide customization of tools and techniques w.r.t prevalent technologies encompassing agility, object orientation, component based modeling and reuse, architecture centric approaches, configuration and risk management, heterogeneous project types and sizes.

## KEYWORDS

| | |
|---|---|
| Software Process Reengineering | Software Process Automation |
| SPI | CMM |
| SME's | Software Process Customization |

## Preface

Software Process Reengineering is thought of as a vehicle which has the ability to carry the organizations achieves higher capability levels. The efficient and qualitative software project management totally finds its success in managing the triple constraint i.e. scope, time and cost. But, Software Process Reengineering adds some more dimensions to software project management at a higher level of abstraction i.e. at process level. It guides and directs the available set of processes to be reengineering e.g. adding more effective and efficient software processes, deleting ineffective set of processes, reordering the processes and activities in process set, configuring i.e. adapting and changing the processes according to the project requirements. Therefore SPI is really an indispensible cog in the gears of software process standardization and continuous software quality enhancement. This paper's objective is to find the state of art in SPI in Small Scale Enterprises SME's, sieve the essential parameters from existing approaches towards SPI which focus on SMEs. Further, these parameters can be studded into SPI which can be used to add a new vision to SPI. The explored factors will be able to characterize the relevant technology, tools, methods, software process automation according to the needs of the SME's. A further objective is that of discussing the significant issues related to this area of knowledge, and to propose different strategies from which innovative research activities can be thought of and planned. As small and medium-sized software enterprises (SMEs) because are not capable of bearing the cost of implementing these software process improvement programs. Implementation of software engineering techniques is difficult task for SMEs as they often operate on limited resources and with strict time constraints. There are number of methodologies to address these issues. In this paper, various SPI methodologies for SMEs are discussed. This will lead towards maturity of software process improvement in SMEs and also facilitates in development of automation tools for SPIs in future.

## Standard Approaches

The evolution of Software Process and Quality Frameworks described by (Sheard, 2001)[i] represented the multitude of frameworks and standards used to derive one from other shows the usage of one framework in developing another. For example, the Systems Engineering Capability Maturity Model (SE-CMM) of EPIC1 developed from the Capability Maturity Model (CMM) for Software, the International organization for standardization (ISO) Software Process Improvement Capability dEtermination (SPICE), MIL-STD-499B (draft), and the Institute of Electrical and Electronics Engineers standard for systems engineering [IEEE 1220]. The SE-CMM was subsequently used in creating the Integrated Product Development CMM [IPD-CMM], the Security Systems Engineering CMM (SSECMM), and a merged Systems Engineering Capability Model (SECM) that is currently being developed with facilitation from the Electronics Industries Association (EIA). Incoming Arrows in following figure 1 shows how one standard is derived from other different SPI standards.

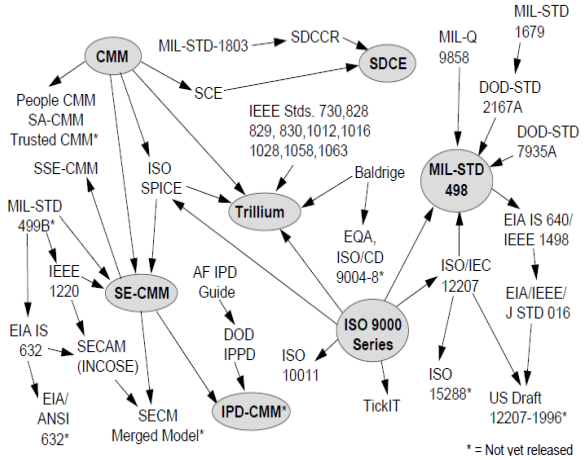


Figure 1: The Frameworks Quagmire (Sheard, 2001)[i]

**(a) Capability Maturity Model (CMM):** Capability Maturity Model (CMM) proposed by the US Software Engineering Institute (SEI) to measure a contractor's ability to develop quality software (Humphery W. , 1989) [ii]

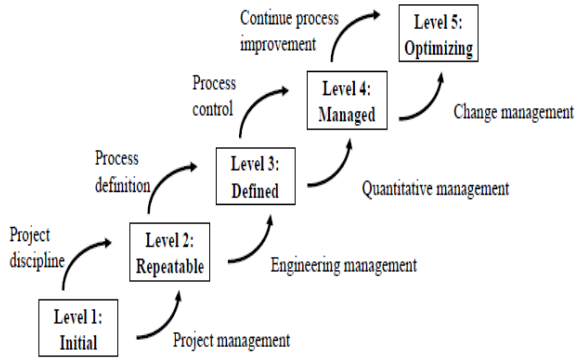


Figure 2:CMM (Adopted from (Humphery W. , 1989)[ii] )

The Capability Maturity Model (CMM) developed at the Software Engineering Institute is based on the premises that maturity indicates capability and to obtain continuous process improvement it is much better to take small evolutionary steps rather than revolutionary innovations. it aims at guiding software organizations in selecting process improvement strategies by first determining their current process maturity before identifying their organization's critical quality and process improvement issues. These five developmental stages are referred to as maturity levels, and at each level, the organization has a distinct process capability. By moving up these levels, the organization's capability is consistently improved.

**(b) Capability Maturity Model Integration (CMMI):** Capability Maturity Model Integration (CMMI) is a process improvement approach that provides organizations with the essential elements of effective processes. It is a model that consists of best practices for system and software development and maintenance. The model may also be used as a framework for appraising the process maturity of the organization. CMMI has features like Integration of software engineering and system engineering, Treating an each process very minutely, Focusing on continuous improvement (Crosby, 1979)[iii]

| Level | Focus | Process Areas |
|---|---|---|
| 5 Optimizing | Continuous process improvement and resolution | Organization innovation and development, Casual Analysis |
| 4 Quantitative managed | Quantitative management | Organization process performance Quantitative process management |
| 3 Defined | Process standardization | Requirements development, Technical solution, Product integration, Verification, Validation, Organization process focus, Organization process definition, Organization project management, Integrated supplier management, Risk management, Decision analysis and resolution, Organization Environment for integration, Integrated training |
| 2 Managed | Basic project management | Requirements management, project planning, Project Project monitoring and control, Supplier agreement management, Measurement and analysis, Process and product quality assurance, Configuration Mgmt |
| 1 Performance | None | |

Figure 3: CMMI (Adopted from (Crosby, 1979) [iii]).

**(c) Software Process Improvement and Capability dEtermination (SPICE):** SPICE stands for Software Process Improvement and Capability dEtermination. 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 (Dorling, 1993)[iv]. The objective is to assist the software industry to make significant gains in productivity and quality, while at the same time helping purchasers to get better value for money and reduce the risk associated with large software projects and purchases (Route and Terrence, 1995)[v]Model is depicted in Figure 4.

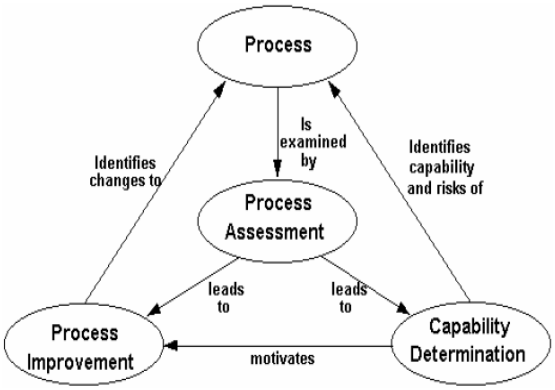


Figure 4:SPICE Adopted from (Route and Terrence, 1995)[v])

**(d) BOOTSTRAP:** BOOTSTRAP methodology can be applied to small and medium size software companies or software departments within a large organization. A new release (Release 3.0) of the BOOTSTRAP methodology has been developed to assure conformance with the emerging ISO standard for software process assessment and improvement (Hasse, 1994)[vi].The BOOTSTRAP methodology explored by (Kuvaja, 1994)[vii] provides support for the evaluation of process capability against a set of recognized software engineering best practices, include internationally recognized software engineering standards, identify organization's process strengths and weaknesses, support improvement planning with suitable and reliable results and also support the achievement of the organization's goals by planning improvement actions.

## Current Trends and Approaches to Software Process Customization

**(a) Software Product Line Practices:** The software engineering product line practices argued by (Jones, 2002)[viii] include those practices necessary to apply the appropriate technology to create and evolve both core assets and products as follows:

Architecture Definition, Architecture Evaluation, Component Development, COTS utilization, Mining Existing Assets, Requirements Engineering, Software System Integration , Testing, Understanding Relevant Domains. Technical management practice: Configuration Management, Data Collection, Metrics, and Tracking, Make/Buy/Mine/Commission Analysis, Process Definition, Scoping, Technical Planning, Technical Risk Management, Tool Support. Organizational Management Practices: Building a Business Case, Customer Interface Management, Developing an Acquisition Strategy, Funding, Launching and Institutionalizing, Market Analysis, Operations, Organizational Planning, Organizational Risk Management

**(b) Software Process Customization:** (McCormick, 2001)[ix] emphasizes "What's needed is not a single software methodology, but a rich toolkit of process patterns and 'methodology components' (deliverables, techniques, process flows, and so forth) along with guidelines for how to plug them together

to customize a methodology for any given project." In (Moitra, 2001)[x] opinion Quality in the Indian software industry is also reflected in the high maturity of Indian software companies. India has the highest number of CMM level 5 companies. More than half of all CMM level 5 companies in the world are located in India. In addition, (Asundi, 1999)[xi] finds most Indian software companies have achieved ISO 9001 certification. Other quality certifications such as COPC, SixSigma, and People-CMM have also been achieved by a large number of Indian software companies. It is noticed by (Bhatnagar, 1987)[xii]that India has many domestic challenges to overcome such as poverty and illiteracy, the Indian software industry has matured and is predicted to play a significant role in software services and product markets .The existing skilled human resource has remained one of the most important reasons for companies to outsource to India.

**(c) The PrIME Project:** The Process Improvement in Multimodel Environments (PrIME) project will span a breadth of topics that are needed for an organization to be successful with process improvement in multimodel environments explored by (SEI). The project will concentrate on several subsets of models and standards that are commonly used in industry, such as Six Sigma, CMMI, Lean, and Agile methods.

**(d) Intel's Idea of Customization: Accelerated Software Process:** *Intel's* Information Technology (IT) department (Brodnik, 2008)[xiii]explored the idea of ASP[1] Improvement. They found that small projects which are less than six months have small teams, limited scope, and low risk need not to be appraised like large projects. Small projects should have different set of simplified version software processes clubbed into their software improvement initiatives. So, variety of project sizes should not have a single set of processes. Software processes must be tailored and customized according to the perspective of key stakeholders, engineers, process coaches, and auditors

**(e) Wipro's Idea of Customization: veloci-Q:** veloci-Q (Subhramanyam, 2004)[xiv] enabled the delivery of quality products and services without slowing down the system i.e. in time and quick delivery. In figure 5, the Unified Quality System veloci-Q removed outdated information and duplications that had accumulated over time. Introduction of Six Sigma concepts increased the customer and business focus of project execution processes. veloci-Q complies with the coveted Capability Maturity Model Integration for Systems Engineering, Software Engineering, and Integrated Product and Process Development (CMMI-SE/SW/IPPD), V1.1 and ISO 9001:2000 frameworks

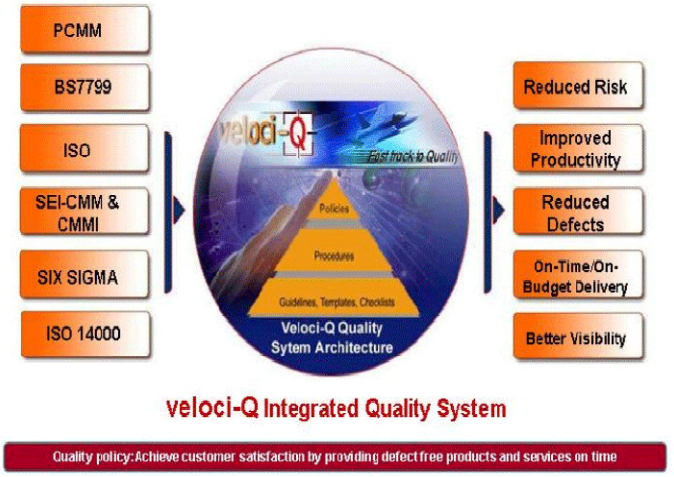

Figure 5: Unified Quality System (Subhramanyam, 2004)[xiv]

**(f) PRISMS Project:** Process Improvement for Small to Medium Software enterprises (PRISMS). The model (Commision of Software Standards, 2009)[xv] enables individual SMEs to tailor the software process improvement to the organization's business objectives. It works towards identifying key process areas and customizing and assessing these processes on ordinal scale according to their degree of conformance to the defined process. Figure 6 shows that assessment methods helps to aims at finding and prioritizing the essential set of processes based upon which the organization works towards maturity.

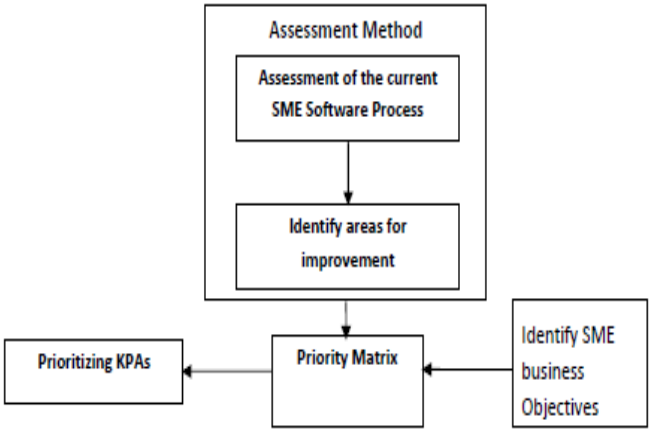


**Figure 6:** SMEs Area's for Improvement Prioritization Framework (Commision of Software Standards, 2009)[xv]

## Multidimensional Approaches to Software Process Reengineering:

| Thrust Area | Major Contribution |
|---|---|
| Software Process Improvement using Service Oriented Approach (Park, 2007)[xvi] | SIR-CM to store and manage heterogeneous assets which produced from software process improvement tools and adapted the Service Oriented Approach to construct the repository for connecting many different types of software process improvement tools. Configuration management method is also integrated to control and manage the assets. |
| A Gradual Approach for Software Process Improvement in Small and Medium enterprise (Alexandre, 2006)[xvii] | The approach proposes a gradual Software Process Improvement framework based on a series of gradual assessments: a micro-evaluation, an OWPL-evaluation and a SPICE or CMM assessment. It allows SMEs to start SPI in a much targeted manner, to quickly progress within a limited budget and, eventually, to reach an acceptable level according to SPI standard models such as CMM and SPICE. |
| Process and infrastructure oriented improvement | Both CMMI and MPS Model-based assessment |

| | |
|---|---|
| initiatives in Small Settings (Montoni, 2007)[xviii] | techniques indicates the TABA Workstation (Process-centered Software Engineering Environment) as a significant strength for the SPI implementations. The main objective of TABA Workstation is to provide an infrastructure to overcome inherent difficulties of SPI implementation initiatives like lack of financial resources. Moreover, the knowledge required for executing the improved processes is captured within the TABA workstation knowledge base. |
| Requirements Engineering Process Assessment and Improvement (Sommerville & Ransom, 2005)[xix] | To generate the greatest increase in RE maturity for the lowest cost without compromising organizational profitability or operations, following pragmatic proposals for maturity improvement:<br>(1) Focus improvements on areas of requirements engineering where the company is weak as suggested by the Area/Strength matrix. (2) Consolidate and standardize practices that are already in use in the company. (3) Only introduce new practices where the cost of introduction is low |
| Framework-Based Software Process Improvement (Jalote, 2002)[xx] | Lessons learned regarding managing the CMM initiative in an organization:<br>-Treat each SPI initiative as a project.<br>-Have a schedule of one year or less for SPI initiative<br>-Manage the risks to the SPI project |
| Return on Investment for implementing a SPI (Humphery W. S., 1991)[xxi] | Potential return on investment (ROI) for implementing a continuous software process improvement program can be as much as 8:1 within the first 2 years-$8 saved for every $1 invested-thus providing the needed stimulus and motivation. |
| Organizational-level software process improvement model (O-SPIM) (Xiaoguang, 2008)[xxii] | SQMSP Software Quality Management and Support Platform (SQMSP), which was used in some medium-sized enterprises in China. Organizational level software process improvement model is an object-oriented model, which mainly supports the software products development and business control from balancing, implementing and supporting the organizational level business. It covers the business processes for multi-product, multi-project comprehensive quality assurance |

| | |
|---|---|
| | features This model gives solutions for organizational-level requirements assigning, project balancing and resources balancing. |
| Best practices in implementation of software process improvement (Galinac, 2009)[xxiii] | The SPI implementation strategy consists of 14 best practices deriving from agile methods, exploiting ideas of incremental deliveries, short iterations with frequent reviews and close collaboration with customers, which are intended to be suitable for global software development (GSD) organizations. It also emphasizes that improvement teams implementing the strategy are more likely to have better progress and achieve better effectiveness in terms of improvement deployment within development teams. |
| Software Quality and IS project performance improvements (Girish, 2007)[xxiv] | IS implementation strategies – executive commitment and prototyping – have a significant impact on both software quality and project performance, training had a significant effect only on software quality and simplicity has a significant effect only on project performance. |
| Software Release Planning: an evolutionary and iterative approach (Greer & Ruhe, 2004)[xxv] | EVOLVE uses a genetic algorithm to derive potential release plans within redefined technical constraints to achieve higher flexibility and to better satisfy actual customer requirements. It takes as input a given set of requirements with their effort estimations and their categorizations into priorities by representative stakeholders. |
| Goal-Driven Agent-Oriented Software Processes (Cares, 2006)[xxvi] | i* framework represents the software process using an agent-oriented language to model it and a goal-driven procedure to design for improving the quality of software processes which require their explicit representation and management. |
| Measurement Practices at Maturity: Levels 3 and 4 (Alain, 2008)[xxvii] | Evaluation and continuous improvement of software maintenance are key contributors to improving software quality. The software maintenance function suffers from a scarcity of the management models that would facilitate these functions |
| IT Performance Improvement (Mallette, 2005)[xxviii] | Using COBIT with SEI CMM combines the best of both worlds to improve IT performance and drive the results to the business bottom line. The IT Governance. Institute's Control Objectives for Information and related Technology (COBIT) is |

generally accepted as the de facto for IT. Sustaining current performance while continuously reducing costs, decreasing exposure to risk and carving out resources to safely improve performance from a budget constantly targeted for cost reduction is the IT challenge.

## Conclusion

It has been that small and medium-scale software enterprises hire external Consultants for process appraisals i.e. no in-house software process appraisal .This means higher cost expended for getting appraised from outside consultants. As technology has already come a long way therefore adding process assets reusable libraries, reusable design and code components, reusable verified and validated requirements set and object orientated methods and tools is actually needed to enhance software process improvement small scale enterprises. Agility is another significant feature which can speed up the pace of software development. But to glue agility and reusability of components, design, architecture intact, a process reengineering framework must be developed which can enable process customization according to the project requirement , project scope, project type , size, cost, time and above all quality. Learning modules can be added to the framework to suggest customized set of processes. These learning modules should aim at appropriate process set which can lead the SME's to one higher level of capability in a customized way.

## REFERENCES

[i] Sheard, S. A. (2001). Evolution of the Framework's Quagmire. *Computer* , *34* (7), 96-98.
[ii] Humphrey, W. (1989). *Managing the Software Process.* Addison-Wesley.
[iii] Crosby, P. (1979). *Quality is Free: The Art of Making Quality Certain.* New York: Penguin.
[iv] Dorling, A. (1993). SPICE Software Process Improvement and Capability Determination. *Software Quality Journal* , *2* (4), 209-224.
[v] Rout & Terence, P. 1.-6. (1995). SPICE: A Framework for Software Process Assessment. *Software Process Improvement and Practice* , *1* (1), 57-66.
[vi] Haase, V. M. (1994). BOOTSTRAP: Fine tuning process assessment., *IEEE Software* , *11* (4), 25-37.
[vii] Kuvaja, P. (1994). *Software Process Assessment and Improvement: the BOOTSTRAP Approach.* UK: Blackwell Publishers, Oxford.
[viii] Jones, L. G. (2002). *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice.* US: SEI,Carnegie Mellon University.
[ix] McCormick, M. (2001). Programming Extremism. *Communications of ACM* , *44* (6), 110.
[x] Moitra, D. (2001). Indias Software Industry. *IEEE Software* , 77-80.
[xi] Asundi, A. a. (1999, July). Retrieved august 2010, from www.heinz.cmu.edu: http://www.heinz.cmu.edu/project/india/pubs/nber_iso_jul99.pdf
[xii] Bhatnagar, S. C. (1997). The Indian software industry: moving towards maturity 12. 277 -288. *Journal of Information Technology* , *12*, 277-288.
[xiii] Brodnik, M. P. (2008, Feb). Why Do I Need All That Process? I'm Only a Small Project. *CrossTalk* .
[xiv] Subramanyam, V. P. (2004). *An Integrated Approach to Software Process Improvement at Wipro Technologies:veloci-Q.* US: Software Engineering Institute.
[xv] Commission on Software Standards. (2009, December)*software_process_improvement_standards_specification.* Retrieved July 2010, from http://nic.gov.sd: http://nic.gov.sd/ar/pdf/software_process_improvement_standards_specification.pdf
[xvi] Park, E. J. (2007). Frameworks of Integration Repository for Software Process Improvement using SOA. *Proceedings of 6th IEEE/ACIS International Conference on Computer and Information Science* (pp. 200-206.). IEEE.
[xvii] Alexandre, S. R. (2006). OWPL: A Gradual Approach for Software Process Improvement In SMEs' . *Proceeding of 32nd EUROMICRO Conference on Software Engineering and Advanced Applications* , (pp. 328 – 335).
[xviii] Montoni, M. S. (2007). MPS Model and TABA Workstation: Implementing Software Process Improvement Initiatives in Small Settings'. *Fifth International Workshop on Software Quality, (WoSQ'07)* , (pp. 57-63).
[xix] Sommerville, I. &. (2005). An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement. *ACM Transactions on Software Engineering and Methodology* , *14* (1), 85-117.
[xx] Jalote, P. (2002). Lessons Learned in Framework-Based Software Process Improvement. *Proceedings of Ninth Asia-Pacific Software Engineering Conference* , 261-265.
[xxi] Humphrey, W. S. (1991). Software Process Improvement at Hughes Aircraft. *IEEE Software* , *8* (4), 11-23.
[xxii] Xiaoguang Yan, X. W. (2008). Research on Organizational-level Software Process Improvement Model and Its Implementation. *Proceedings of International Symposium on Computer Science and Computational Technology.*
[xxiii] Galinac, T. (2009). Empirical evaluation of selected best practices in implementation of software process improvement. *Information and Software Technology* , *51* (9), 1351–1364.

[xxiv] Girish, H. J. (2007). Software quality and IS project performance improvements from software development process maturity and IS implementation strategies. *Journal of Systems and Software* , *80* (4), 616–627.
[xxv] Greer, D. & Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. *Information and Software Technology* , *46* (4), 243–253.
[xxvi] Cares, C. F. (2006). Goal-Driven Agent-Oriented Software Processes. *Proceedings of 32nd EUROMICRO Conference on Software Engineering and Advanced Application*, (pp. 336-347).
[xxvii] Alain, A. (2008). A Software Maintenance Maturity Model (S3M): Measurement Practices at Maturity Levels 3 and 4' International Workshop on Software Quality and Maintainability. *Electronic Notes in Theoretical Computer Science* , *233*, 73-87.
[xxviii] Mallette, D. &. (2005). IT Performance improvement with COBIT and the SEI CMM. *Information Systems Control Journal* ,

http://www.karamsociety.org