The Knowledge Based Software Process Improvement Program: A Rational Analysis

Alagarsamy K
*Computer Center, Madurai Kamarj University,
Madurai, TN, India.
alagar_samy54@rediffmail.com*

Justus S
*Member IEEE
Dept of Microprocessor & Computer Science,
School of Physics, MKU.
justus.mku@gmail.com*

Iyakutti K
*Dept of Microprocessor & Computer Science,
School of Physics, MKU,
Madurai, TN, India.
iyakutti@gmail.com*

Abstract

Knowledge management is the key area of focus in the present information technology scenario. It forms a basis to derive standards and models and steers organizations through an enjoyable journey, an improved endeavor to reach the destination. Software process improvement program is a crucial venture for organizations functioning under a framework model and aspiring for higher maturity levels. While earlier works for software process improvement have been considering wider range of initiatives, we deem knowledge management to be a contemporary approach for refining software process improvement activities.

This paper is committed to a rational analysis into the knowledge-based guidance for implementing a software process improvement program. The work is directed by four research questions that focus on the knowledge based SPI initiative. The role of knowledge components and a knowledge driven model (KDM) are assessed by a measurement model. The impact of KDM on the end-product and its real effect on SPI is measured by quantifying the productivity of the projects, eventually the organization. An implementation of the knowledge driven software process improvement (SPI) program is explained with a suitable case study, an organization working towards attaining CMM level. Future issues pertaining to knowledge based process improvement forms the concluding note of this work.

1. Introduction

Most business or software success stories often advise that a business or a software process improvement journey begins by first applying a standard or model. They use standards and models not as bibles but as planning tools and as checklists to be considered when determining completeness of their planned efforts [3]. Using the so called standards and models is like using a roadmap when taking on a journey. A roadmap is a great initial planning tool and a good guide for a new and never-yet-traveled route to destination. It indicates what is possible and preferable and checks potential rewards in the journey. It does not dictate how to start, where to stop, nor advice on the required resources, costs, schedule and expected return-on-investment. Hence standards are only guide for the intended journey of an organization.

But improvement is felt only when the journey needs to be enjoyable and rewarding by consuming less resources. Though software process improvement (SPI) is implemented by several means [5], [7], knowledge based SPI is preferred to be admitted as effective and efficient way for improvement. Software processes have evolved from data driven processes to information driven processes to, today's, knowledge driven processes. And knowledge rules the dominant society in organizations and processes that govern their operational functions [2]. While literature and research works still speak on knowledge management, knowledge engineering and knowledge based models, we endeavor to propose a rational analysis of a knowledge based approach for software process improvement.

To rationalize the work we have framed four research questions, in the next section. In the third section we explain the basic working of a software process improvement program. The analysis of the knowledge based SPI program is discussed in section 4. The influence of the components of the knowledge based SPI over software processes is discussed and a functional model, Knowledge driven model, is derived from the existing IDEAL model [8]. The real impact of knowledge in SPI is at the organization's productivity, which is measured using the two proposed approaches. In section 5, we have presented a case study on implementing KSPI in a software development environment. The final section

International Conference on Software Engineering Advances(ICSEA 2007)
0-7695-2937-2/07 $25.00 © 2007 IEEE

concludes with future issues pertaining to this area of study, research and practice.

2. Research Questions

Many theories exist for knowledge creation and management. Also process improvement for software development is available in literature and is practiced in every software development activity. After an extensive study on the SPI literature, four investigative questions were framed, which would address where we intend to move and throw light on the impact of knowledge management in the vital areas of software process improvement. The four key research questions are

i. What methodologies are available for managing knowledge that is internally generated and externally shared?
ii. How software processes are around the knowledge managing components?
iii. What impact do the knowledge based processes have on the to-be delivered software products?
iv. Does the impact of knowledge really improve the software processes?

The first question addresses the theory of knowledge management and its practicing platforms. This thought on managing knowledge is widely discussed in our earlier work [2]. The second question covers how the role of knowledge components commands the software processes. The next question is on the impact of the knowledge based processes on the end-product. The final question focuses on the real improvement of software process that is knowledge driven. A practice oriented perspective is ventured to find answers to these research questions. The investigative directions based on these notions found way to the formulation of a knowledge driven model (KDM) for software improvement program and its operation. A study is conducted on the operational status of the SPI program based on the acquired knowledge, which could guide the organizations to compete for higher maturity levels.

3. Software Process Improvement Approach

The literature concerning software process improvement pilots the organization to operate the processes that could influence, positively, the final end-product to be delivered [12]. The SPI phases are incessantly executed in any development environment [1], [5], [8].

Phase 1- Learning: Learning is the required starting point of the overall process improvement program [2], [11], [15]. Without this baseline of the products, process and the environment, no basis for change and improvement can exist. Knowledge based organizational learning plays a vital role in knowledge generation, externalization and utilization [11], and leads to a careful understanding of the process models and their improvement strategies; however, this phase is often ignored.
Phase 2 – Evaluating: Evaluation of the objectives and goals of improvement is carried out with certain basic requirements: 1) Identification of baseline against which the evaluation can be done; 2) Changes are prioritized and the ones which will be focused are selected; 3) The evaluated goals and objectives must be measurable. Evaluation cannot be overemphasized because it causes multiple impact on the factors being examined; positive impact, no impact and negative impact [10], [21].
Phase 3 – Packaging: The changes, identified during the previous phase, in the subsisting processes and those having satisfactory results are integrated into the core process stream of the organization. In this phase, based on what has been learned and experienced, new models, standards and training materials are developed. These training tools are packaged into the knowledge repositories which are then fed back to individuals requiring information for their next projects and experiments [14], [17]. Every element of the standards needs to be assessed, prior to packaging.
Phase 4 – Operating: The knowledge dumped in the repositories, however, becomes unwarranted collections unless they find better operating environment [4], [11]. In global-competent, CMM or other model-based organizations, these knowledge banks are well managed using knowledge management tools. KMT focus on the management of documents and better utilization of the generated, packaged, externalized knowledge are available with these tools, and sometimes or often they enable knowledge reutilization [15], [17].

As organizations may find different routes to reach their derived and set objectives they are forced to follow certain of these activities to have an improved process to attain their goal. While the phases discussed above may be demanding high cost, dedication, they guarantee a measurable improvement in the process of software development.

4. The Knowledge based SPI Program

Knowledge management is the set of systematic and disciplined actions that an organization can take to obtain the greatest value from the knowledge available to it [4]. “Knowledge” in this context includes both the experience and understanding of the people in the organization and the information artifacts, such as documents and reports, available within the organization and in the world outside [4],

[17]. The operations of knowledge management fall into one of the two categories of school of SPI – the model based approach and pragmatic approach [5]. The latter approach focuses on activities that function but also have the improvement potential and the former activities that do not work but need replacement or considerable improvement. These approaches of Software Process Improvement (SPI), in its current form, emerge from work at the Software Engineering Institute (SEI) at the Carnegie-Mellon University in the 80's [7], [8].

In practice, SPI is operational based on these two approaches; processes that could be improved and those that require improvement, to remain active and dynamic. Moreover, the process of SPI itself is a continuous and evolutionary process with repeating cycles through the following steps [8]: assess current software practices, identify and prioritize improvements, implement and institutionalize improvements, evaluate improvement experiences.

The process of SPI is assessed based on the so-called Maturity Models. The most widely used model for software development is the Capability Maturity Model (CMM) [7]. The maturity levels are described as a number of (generic) software development and management practices, and an organization whose software practices correspond to the description of a level is said to be at that level [3]. Maturity models are often seemed to embody knowledge about "good" software practices, and on where to focus improvement initiatives given the current status of an organization. Though SPI could be approached and employed through several ways and means, we rationalize a contemporary approach, knowledge based model for executing SPI to be appreciable. The following sub-sections discuss the practicability of the knowledge centered SPI program.

4.1. Components of Knowledge based SPI

Knowledge management (KM) and software process improvement (SPI) are two different worlds housing committed, challenging and enthusiastic set of activities. To facilitate a better journey that is knowledge driven, an excelled insight into the knowledge entity is demanded. However, the understanding of the software processes threw light on the essential components that involve in the knowledge driven software process improvement program. The three primary components are designated as follows:

- *Processed information,* information obtained through acquisition of data is processed using mining techniques, codified and stored in the warehouses, which is termed as knowledge repositories. Processing of the stored information, results in new knowledge creation which serves as the source for deriving norms and guiding principles for the progress of the organization's improvement programs.
- *Knowledge workers,* consisting of administrative, technical and support staff are highly concerned about the generated knowledge and are responsible for the management of the knowledge. They package the knowledge in the form of policies, standards and learning materials, to generate explicit knowledge, and, in general, propose models of the product and of the processes, both formal and informal.
- *Knowledge users*, the top-level decision making executives, are the focal point in the entire improvement program, and they are the key users of knowledge, packaged by the technicians. They translate explicit knowledge to tacit knowledge, which guides decision making. The sole activity of this component of people is the implementation and the operation of the packaged knowledge which serves as the primary reactants of the SPI program.

The knowledge workers and users are directly involved in the knowledge based improvement program whereas the acquired data (or information, Knowledge) serve as spine of the entire process of knowledge based software process improvement. On the contrary, SPI itself is not a process that can be automated, installed and executed when desired; rather, it comprises a set of activities that involve technical, managerial and even human behaviour aspects. Hence, SPI is an intricate task to proceed and needs help from wider spectrum of disciplines [9]. The activity chart defined for SPI program may vary and gets refined from time-to –time and when necessary, from project to project. The degree of change in the chart is proportional to the organization's contribution and its dedication for the improvement program [1]. Table 1 states the set of activities that remain unchanged (static) and ones that often requires change (dynamic).

Most or even all of the dynamic activities, mentioned in the table, are specific and centered on knowledge and its associated components. While Static activities are derived from CMMI or other existing conventional standards, dynamic activities are based on interactions of knowledge workers and knowledge users, and knowledge activities like codification and populating the knowledge repositories, which in later phases help the management in institutionalizing and implementing the newly found hybrid knowledge for planning and controlling the process improvement program. Integrating the components across the nature of data repositories, cultural, continental and psychological differences of the knowledge workers and users remains a great challenge of the SPI program [2], [10], [11]. The functional effort of the integrated

Table 1. KSPI Components and their activities

Components in the Organization	Static Activities	Dynamic Activities
Development and Maintenance Team	Finish Software Products on time and within budget	Interact with analysts for goal setting and feedback
Software Processes	Stick on to processes defined for development and maintenance	Interact with analysts and knowledge workers to use the packaged and tailor-made processes
Production Processes	Develop documentation Document experiences	Codify data to knowledge repositories Provide feedback to knowledge workers
Management	Plan and control Use decision support tools	Interact with development project leaders and analysts Create new hybrid knowledge

components I_c can be estimated using process metrics that supports the management, understanding and prediction of the software development processes [9].

Conceptually and more precisely, higher the degree of interaction among the knowledge components more the dynamic processes (activities) are involved in action, causing the entire set of processes to improve. Thus our second research question is answered.

4.2 The Knowledge driven model for SPI

In the words of Schneider & von Hunnius *"Without a learning attitude and some appreciation for continuous process improvement, even best repositories will not make experiences fly"* [14] the importance and impact of knowledge, attained through learning, is felt in process improvement programs. Since the improvement programs are continuous, the knowledge repositories are to be continuously updated and reengineered based on the baseline requirement of the project environment. This unremitting activity of updation is coupled, sometimes loosely in small software enterprises and often tightly in large organizations, with the consistent, continuous and committed process of learning in organizations [2]. Learning leads to acquisition of tacit knowledge which when externalized becomes explicit knowledge [16]. This two-way transformation of acquired knowledge also forms the base for the improvement programs. In this section the research questions, framed earlier, on the impact of KM on SPI are answered.

The operation and the implementation of software process improvement, a demanding and complex undertaking, is supported by the so-called IDEAL model, a framework proposed by the Software Engineering Institute (SEI) [9]. All of the five phases in the model: Initiating, Diagnosing, Establishing, Acting and Leveraging, provide the path of actions that constitutes the SPI program.

From an extensive study, over a period of two years, on the SPI literature [2], [3], [4], [7], [10], [12], [15], [16] and analysis of the improvement strategies practiced in the software development enterprises we aspire to derive a new knowledge driven model (KDM) for SPI program based on the existing IDEAL model. In table 2, the comparison of the IDEAL model with the proposed KDM that drives SPI is pictured.
The KDM activities are summarized as follows:

- The Initiating phase instigates the improvement program by inspiring the feel to understand the need for it. Budgeting plans are sketched out along with the overall working plan for the improvement program.
- In the Diagnosing phase a wider study on the literature (documents, reports and memos from earlier similar projects) is performed for the acquisition of tacit knowledge. Conversion of explicit knowledge to tacit knowledge is through internalization [16]
- In the Establishing phase, the operation of the acquired knowledge is through packaging. The real maneuver of knowledge driven SPI is by implementing knowledge management tools that is based on knowledge engineering and data mining techniques.
- The Acting phase integrates and executes the knowledge derived from the previous phase and facilitates the SPI program into action. Ultimately the derived knowledge characterizes attributes for individual processes and the resultant process metrics aids the monitoring of the SPI program.

The IDEAL model, framed for SPI program, has led to the derivation of a knowledge driven model which focuses on the acquisition of explicit knowledge through externalization and its conversion to tacit knowledge through internalization and continuous learning [3], [17]. The operation of the KSPI is solely based on the tacit knowledge, acquired through internalization, of the executers. However, as long as the knowledge holders remain in the organization, the tacit knowledge could be transformed to explicit knowledge, in the form of documents, reports and memos. This process of converting tacit to explicit knowledge can be defined

IEEE COMPUTER SOCIETY

Table 2. The KDM activities for SPI

Phases	IDEAL activities	KDM activities
Initiating	Stimulate improvement	Understand the need for improvement
	Establish Sponsorship	Establish sponsorship
	Establish improvement infrastructure	Portray improvement infrastructure
Diagnosing	Characterize current practices	Collect existing literature
	Develop recommendations & Document Phase Results	Acquire tacit knowledge
Establishing	Set Strategy & priorities	Package knowledge for operation
	Plan actions Establish process action teams	Implement knowledge engineering techniques
		Operate KM tools for DSS
Acting	Plan, Execute & Track Installation	Derive required knowledge for SPI planning & execution
	Plan Execute Pilots	
	Define processes and measures	Characterize attributes for individual processes
Leveraging	Document & Analyze Lessons	Populate repositories & Analyze information
		Acquire explicit knowledge
	Revise Organizational approach	Derive hybrid knowledge through combination

as recording of experiences. Hence, migration of people in and out of the organization does not affect the knowledge banks. Moreover, the acquired knowledge, either tacit or explicit, only when shared, is consequential. The key challenge a knowledge based SPI program faces is knowledge sharing [2], [11], [14].

With this discussion on the proposed KDM and the consequent steps necessary to execute KSPI, the third and fourth research questions are promptly addressed. Any improvement program strikes at the productivity of the process or the quality of the end-product [19] and other several personnel related issues [21]. Hence, measuring and monitoring the knowledge based improvement in the productivity of the end-product, and ultimately the organization, assures that the knowledge based SPI program is very effective.

4.3. Measuring the knowledge based SPI

The impact of the Knowledge based SPI program is assessed by measuring the productivity of the projects, before and after applying the KDM. These measurement strategies are chosen based on the validity of the work done by Joost [22].

4.3.1. Conventional method Most studies define productivity as size divided by effort. According to Conte [18], productivity is defined as *the number of lines of source code produced per programmer-month (person-month) of effort*. Thus the following formula

$$P = \frac{1}{n}\sum_{i=0}^{n} p_i = \frac{1}{n}\sum_{i=0}^{n} \frac{s_i}{e_i} \qquad (1)$$

Where P is organizational productivity, p_i is the productivity of the ith project, s_i is the size of the software, e_i is the effort of the software, n is the number of projects in an organization. The size measure of the project can be functional points, object points or any other similar metrics and effort can be expressed in other measures without loss of generality. Productivity of the organization can be estimated by averaging the individual project's productivity. The effect of KSPI on the productivity of the organization before and after applying KDM is determined by using a t-test, which is as follows

$$\text{Effect}_0\text{: } P^{(!spi)} = P^{(spi)}$$

$$\text{Effect}_1\text{: } P^{(!spi)} \neq P^{(spi)}$$

This is a simple conventional method to estimate the effect of the KDM for SPI in an organization.

4.3.2. Regression Model method In works related to productivity, better comparisons of productivity changes among projects in an organization are required. But the conventional method satisfies this hypothesis to a lesser extent, showing insignificant level of variance. Hence we adopt a linear regression model where productivity is described by a regression equation that models effort as function of size and regression parameters [19]. The equation is a follows:

$$e_i = \beta_o + \beta_1 \cdot S_{fpi} + r_i \qquad (2)$$

Where ei is effort of project in hours, S_{fpi} is size of effort in functional points, r_i is residual variance. To ascertain the impact of KDM for SPI, the knowledge based process maturity (K) of the organization is factored into equation (2).

$$e_i = \beta_o + \beta_1 \cdot S_{fpi} + \beta_2 \cdot K_i + \beta_3 \cdot S_{fpi} \cdot K_i + r_i \quad (3)$$

Using ANOVA test [20], the regression parameters β_2 and β_3 can be calculated. By analyzing these values the estimate of the scaled effort, in hours, is obtained.

$$E_{hr}0: \beta_2 = 0 \text{ and } \beta_3 = 0$$

$$E_{hr}1: \beta_2 \neq 0 \text{ and } \beta_3 \neq 0$$

The advantage of the regression model approach is that regression models take the effect of project size on the productivity of a project into account. One important thing to be noted is the residual value in the data increases if the project size is ignored. This results in the requirement of larger sample sizes to obtain significant and consistent results.

4.3.3. Results In order to find the real practice of traditional SPI and knowledge based SPI, an analysis on two different organizations' project databases were carried out during 2006. The first organization is an ISO certified institute and the second is a CMM level 2 corporate working towards level 3. Two different experimental setups were considered so that the measure of productivity might be considerably significant.

Table 3. Results of the measurement

Maturity Level	**Conventional Method** $P^{(!spi)} \sim P^{(spi)}$	**Regression Model Method**		
		$log(S_{fp})$	$log(S_{fp}):K$	E_{hr}
Institute	14.78	182.33	0.413	29.65
CMM 2	11.25	169.47	0.725	22.32
CMM 3	9.64	158.65	0.963	18.47

For the regression model method, the independent functional point variable, the knowledge process maturity constant value, K, and a size/effort ratio are identified as the attributes for measurement of the productivity. From table 3 we infer the following:

- In conventional model the productivity of the CMM level 3 corporate shows nearly 35% increase than the institute. But when productivity is factored as a function of knowledge based maturity, we appreciate only 2% of the differences in productivity by SPI.
- In the regression model, after several rigorous tests and experimentation we conclude that the effort in hours estimated for the CMM level 3 organization shows an appreciable value of 37% increase in productivity than the institute. The explained variance is observed as R^2=0.60. That is 60% of variance in effort can be explained by knowledge process maturity and size.

5. Implementing KSPI: A Case Study

In the preceding sections an elaborate study of the knowledge based software process improvement program has been done with due measurement methodologies and results. The implementation of the knowledge driven software improvement program was conducted in an organization, located in Chennai, India, working towards CMM level 1 status. The organization's primary occupation is development of application softwares and is currently handling 4 projects with clients in India, USA and East-Asian countries. A four month study is illustrated with an example case.

A software developer X is working under a project leader Y, who is aided by the knowledge worker Z. X faces a problem in writing a design specification for a problem defined by Y. So she enquires with Z, who in turn collects some relevant and similar information available in the knowledge bank. With two more model design documents she drafts a new one for the said problem. Now she has the tacit knowledge in writing the deign specification. She now records this experience and it becomes explicit knowledge. Now Y wants to share this situation with his other personals, and thus explicit knowledge is shared. Formal meetings and e-library are the tools that incorporate knowledge-sharing. Thus all the components in KSPI are involved in dynamic activity. X took 14days to draft the design document.

After two months time a similar situation was handled with an improved approach. It took just 6days to complete the design spec. As people in the organization learn through experiences they acquire knowledge in that particular domain, which is recorded periodically in the knowledge banks. This organization uses web-based knowledge banks and intranet mailing tools for sharing and communications. Project leaders and top-level management felt that improvement could be felt once the people in the organization appreciate the process of knowledge based endeavors in their improvement program. Still other issues in applying this approach to larger software development organizations remain unexplored.

6. Conclusion

The foremost issue of demand in the organizations, partially or wholly, participating in software development needs an improvement in it. Though custom-made and tailor-made processes are worked out based on the standards and models, the need for refinement and improvement becomes mandatory [5], [7]. As standards don't provide path for improvement, SPI executors find improvement challenging in their every maneuver. Hence, this work attempted to integrate SPI with a knowledge base, rather than database, and derived a protocol for a SPI program that is knowledge driven. Though the model, KDM, has it's base from IDEAL model, it has its own significance in the way of its knowledge based implementation. The outcome of the KSPI is experimented with two such organizations aspiring for improvement in the software development process and the results are encouraging.

Future Work: This rationale on KSPI has a wider scope and its wings of understanding and exploration are beyond cultural and continental barriers. The further ventures of this work are: 1) Knowledge engineering techniques that constitute Knowledge Management Tools. 2) The role of KMTs in software process improvement programs. 3) A survey of software organizations aspiring for improvement and experiment the KDM to arrive at instructive and appreciable results. 4) Persuading the total participation of software developing organizations in the knowledge based software process improvement program.

Product development goals and generation of productive end-products drove the knowledge based SPI initiative. Top-level and middle level management were committed to short-term goals and product specific plans. Their total indifference in sharing the experience with the next generation of executives caused a huge setback in the part of the entire organization [9]. Hence experiences are codified in the right form and managed to be shared among several generations of workers. To succeed in the improvement effort, organizations need to be informed on multidimensional knowledge based decision making systems. Hence knowledge based Software process improvement has become, to a large extent, a need based service for software development.

7. Acknowledgements

The work is partially funded by K.L.N. Sourashtra Council for Engineering Colleges and Madurai Kamaraj University, Madurai, TN, IND.

8. References

[1] Kellyann Jeletic, Rose Pajerski and Cindy Brown, “Software Process Improvement Guidebook”, NASA, Greenbelt, Maryland, March 1996.

[2] K. Alagarsamy, Justus.S, and K. Iyakutyi, “A Theoretical Perspective on Knowledge based Organizational Learning”, *Proc. 13th APSEC ‘06*, Published in the Digital Library, *IEEE Computer Society Journal*, 2006, pp. 393-400, doi:www.ieeecomputersociety.org/10.1109/APSEC.2006.15.

[3] Jesper Arent and Jacob Norbjerg, “Software Process Improvement as Organizational Knowledge Creation: A Multiple Case Analysis”, *Proc. 33rd Hawaii International Conference on System Sciences*, 2000.

[4] A.D. Marwick, “Knowledge Management Technology”, *IBM System Journal* Vol 40, No 4, 2001.

[5] E.F. Koren and S. Elisabeth Hove, “Using a Knowledge Survey to plan Software Process Improvement Activities – a Case Study”, 2002 http://heim.ifi.uio.no/~isu/INCO/Papers/Using_efk_eurospi2002_paper.pdf

[6] W. Humphrey, *Managing the Software Process*, 1st. edn. Addison-Wesley Publishing Company, 1989.

[7] M. C. Paulk, C. V. Weber, B. Curtis and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process,* Pittsburgh: Addison Wesley, 1993.

[8] B. McFeeley, *IDEAL: A User's Guide for Software Process Improvement*, SEI, Pittsburgh, 1996, Handbook CMU/SEI-96-HB-001.

[9] Seija Komi-Sirvio, *Development and Evaluation of Software Process Improvement Methods*, VTT publications 535, VTT Research Center of Finland, 2004.

[10] I. Ramos, J.C. Riquelme and J. Aroba, “Improvement in the decisión making in software projects”, *Proc. 3rd. Int. Conf. on Enterprise Information Systems*, ICEIS 2001.

[11] Wesley Vestal, *Integrating Knowledge Management and Organizational Learning*, American Productivity & Quality Center, 2005.

[12] B. Fitzgerald, Y. O'Kane, “A Longitudinal Study of Software Process Improvement”, *IEEE Software* May/June 1999, pp. 37-45.

[13] *CMMI of Development*, ver 1.2, SEI, Pitsburgh, 2006, PA 15213-390, CMU/SEI-2006-TR-008.

[14] K. Schneider, and J.P. von Hunnius, “Expeience Repositories for Software Engineering”, *Proc. 25th Int. Conf on Software Engineering*, Portland, Oregon, 2003, pp. 534-539.

[15] K. Kautz and K. Thaysen, “Knowledge, learning and IT Support in a Small Software Company”, *Proc. Conf. of BPRC,* 2000.

[16] Mohd Syazwan Abdullah, Ian Benest, Andy Evans and Chris Kimble, “Knowledge Modeling for developing knowledge management systems”, *Proc. 3rd European Conference on Knowledge Management*, Dublin, Ireland, 2002, pp. 15-25, ISBN:0-9540488-6-5.

[17] Haris Papoutsakis and Ramon Salvador Vallès, “Linking Knowledge Management and Information Technology to Business Performance: A Literature Review and a Proposed Model”, *Journal of Knowledge Management Practice*, 7, 1, 2006.

[18] S.D.Conte, H.E. Dunsmore, and V.Y. Shen, *Software Engineering Metrics and Models*, Benjamin/Cummings Publishing, Menlo Park, CA, 1986.

[19] J. Peter, M.H.Kutner, C.J.Nachtsheim, W.Wasserman, *Applied Linear Statistical Models,* 4th edn. WCB/McGraw-Hill, Boston, MA, 1996

[20] G.K. Bhattacharyya, and R.A. Johnson, *Statistical Concepts and Methods,* Wiley Series in Probability and Statistics. Wiley-Interscience, NY, 1977.

[21] Medha Umarji and Carolyn Seaman, “Predicting Acceptance of Software Process Improvement”, *Human and Social Factors of Software Engineering,* St. Louis, Missouri, USA, 2005.

[22] Joost Schalken1, Sjaak Brinkkemper2, and Hans van Vliet1, Using Linear Regression Models to Analyse the Effect of Software Process Improvement, Source:www.cs.vu.nl/~hans/publications/y2006/profes-2006.pdf