

## Software Process Improvement in SMEs: A Comparative View

Deepti Mishra<sup>1</sup> and Alok Mishra<sup>2</sup>

<sup>1</sup>Department of Computer Engineering,

<sup>2</sup>Department of Software Engineering

Atilim University, Ankara, Turkey

{deepti,alok}@atilim.edu.tr

**Abstract.** The majority of software development organizations all over the world are small and medium enterprises. These organizations have realized that it is crucial for their business to improve their processes and working methods but they are lacking the knowledge and resources to do it. Successful implementation of SPI methodologies in small and medium-sized software enterprises (SMEs) is generally not possible because such organizations are not capable of investing the cost of implementing these programs. Limited resources and strict deadlines to complete the projects make it further difficult to implement SPI programs which can also affect quality issues in software project. There are various SPI methodologies to address these issues which have been also deployed in these organizations. In this paper, recent and significant SPI methodologies (OWPL, ASPE-MSD, iFLAP, PRISMS, SPM, MESOPYME) for SMEs are compared and discussed. This will facilitate the maturity of software process improvement in SMEs, standardization and also contribute in the development of automation tools for SPIs in future.

**Key words:** software process improvement, software quality, small and medium enterprises, SME

### 1. Introduction

It is well known that software quality is largely dependent on the process that is used to create it. [82]. For many years now software process improvement (SPI) has been recognized as an effective way for companies to improve the quality of the software they produce and the productivity with which they work [73]. A software process can be defined as an environment of capable interrelated resources managing a sequence of activities using appropriate methods and practices to develop a software product that conforms to customer's requirements [88]. The desired result is high quality software at low cost. As each software development project is an instance of the process it follows, it is essentially the process that determines the expected outcomes of a project [37]. Software processes play an important role in coordinating

different teams in large organizations so that their practices don't lose touch with one another [23]. Ideally, these processes should combine the need for flexibility and creativity, but that balance is hard to achieve [28]. There is evidence that the majority of small software organizations are not adopting existing standards as they perceive them as being orientated towards large organizations and studies have shown that small firms' negative perceptions of process model standards are primarily driven by negative views of cost, documentation and bureaucracy [47]. A vast majority of software producers, have not yet implemented a methodology for software process improvement, are paying high costs of production and systems maintenance, and are therefore being displaced from the global market, not being on the same competitive level as companies that possesses a process improvement method [34]. There are several models for software process improvement, such as the Capability Maturity Model Integration (CMMI), the Software Process Improvement and Capability dEtermination (SPICE) and the ISO 9000 norms from the International Standardization Organization. These models provide quality patterns that a company should implement to improve its software development process [34]. Unfortunately, it has been observed that the successful implementation of such models is generally not possible within the context of small and medium-sized software organizations because they are not capable of bearing the cost of implementing these software process improvement programs [40, 84] and the proper implementation of software engineering techniques is difficult task for SMEs as they often operate on limited resources and with strict time constraints [84]. Small companies generally need external assistance in planning and implementing process improvement to keep abreast of state-of-the-art software-engineering research and practice [62]. Many small and medium software development organizations have recognized the need to improve their software product and evaluating the software product alone seems insufficient since it is known that its quality is largely dependent on the process that is used to create it [82]. Therefore, these organizations are looking for evaluation of their software processes and products. It is further supported by many researchers that small software organizations are characterized by their insufficient human resources, lack of development and supporting environment, lack of budget and dependency on the large organizations [8] and find software process improvement a major challenge [51]. Another main problem with these organizations is that they do not have a process culture and in a process culture people's customs and behaviors are influenced by process-oriented thinking and process management principles [83]. Dyba [23] indicated that SPI can be used as a competitive advancement strategy for both small and large organizations [23]. Further, Cater-Steel [16] found in their study that the process improvement program was effective in improving the process capability of many of these small software development firms. Today, the software industry is one of the most rapidly growing sectors and this situation stimulates especially the constant creation of small companies which play an important role in the economy [84] and in the last few years, a great number of organizations have been interested in SPI [14]. A considerable amount of

software is produced world-wide by SMEs ranging from 1 to about 50 employees [30]. In this context, German and Brazil; an software market of these companies was around 77% and 69% during 2001 [58]. Richardson [69] observed that there is a need for small software companies in the Irish sector to improve their software process. 73% of software companies in NI (Northern Ireland) would desire to engage in some form of software process improvement program particularly using lighter approaches such as class CMMI appraisal methods [50]. The term *small setting* has been defined as an organization or company of fewer than approximately 100 people, and a project of fewer than approximately 20 people [76]. As mentioned in the Software Engineering Institute Web site for small settings, a major aspect to be considered in these environments is that the amount of resources used to support a process improvement effort would be a large percentage of an organization's operating budget, [76]. Johnson and Brodman define a small organization as fewer than 50 software developers and a small project as fewer than 20 developers [38].

## 2. Related Works and Rationale of SPI in SMEs

The motivation for carrying out process assessment and improvement activities is to collect information as to what needs to be changed and to establish how to pursue the improvements in order to minimize development cost and maximize the quality of products produced [67]. Process assessment is utilized to evaluate the capability of a process [49]. Existing software engineering and organization development literature acknowledges that there are fundamental operational differences between small and large organizations [23]. Small organizations seem more concerned about practice, while large organizations seem more concerned about formal process [23]. Russ and McGregor [71] observed that software development process can be just as critical to a small project's success as it is to that of a large one due to the number of external dependencies per team member. They further argued that its goal is to produce high quality and timely results for today's market without imposing a large overhead on a small project. Paquin [63] identifies assessments, project focus, documentation, required functions and maturity characteristics as process issues for small organization. Abbott [1] identifies six keys to software process improvement in small organizations:

- Senior management support
- Adequate staffing
- Applying project management principles to process improvement
- Integration with ISO 9001
- Assistance from process improvement consultants
- Focus on providing value to projects and to the business

Johnson and Brodman [38] identify seven small organization/small project challenges:

- Handling requirements
- Generating Documentation
- Managing Projects
- Allocating Resources
- Measuring Progress
- Conducting Reviews
- Providing training

Larsen and Kautz [48] also viewed that these organizations are afraid of the initial expenses which they assume are large both with regard to direct costs for process assessment, training and tools, but also due to indirect costs for personal and time resources when implementing improvement actions. Kuvaja et al. [44] further supports that it is quite difficult for any SME to choose an improvement approach, and to apply it in their organization without the help of external consultants or substantial investment in the time of their software managers. Cultural issues like resistance to change from the employees or the management areas, who regard the extra work required for quality assurance as a useless and complicated burden put on the developing team. According to Biro et al. [11] national culture also affects the process improvement methods. Kuvaja et al. [44] mentioned that the main problem of the small companies is that they cannot afford to maintain substantial expertise of software process improvement within their companies, but they have to buy it from external sources. A further problem related to the lack of expertise is to find how to start the improvement and which experts to use. Due to budget constraints, the services of a consultant organization to improve the software quality are not possible. Still the need for a good quality assurance program is becoming more evident, and managers are striving to achieve international quality standards that, in the long run, result in lower production costs [34]. According to Kautz [40], the software process improvement is rewarding and advantageous also for small organizations if it takes into account the peculiarities of such organizations. Dyba [23] also found empirically that small organizations implemented SPI as effectively as large organizations, and in turn, achieved high organizational performance. According to his study, the main lesson to be learned is that to implement SPI at least as effectively as their larger counterparts, small software organizations should capitalize on their relative strengths in employee participation and exploration of new knowledge. There are various approaches, languages and tools for process definition [3]. However, are rarely applied in practice [17] specifically with small organizations [60]. Furthermore, only few studies in the context of small software companies have been performed [43, 72, 74]. In order to get an edge in an ever-growing highly competitive software development world, it is significant for an organization to regularly monitor the software process. It is important for an organization to continuously improve its software process on the basis of

feedback from various stakeholders. It is also supported by Mintzberg [59] that for smaller organization' where much of the work is coordinated through direct supervision and mutual adjustment, it is important to find a balance between these mechanisms and formal, defined and highly detailed documented procedures to facilitate organizational learning [61]. Despite the fact that even in the US most software producing units are comparably small and state a need for improvement [13], little is known about software process improvement in this kind of organization [40]. Kautz [40] further supported the view that even small organizations with little more than two developers can profit from some basic formal routines. According to his research project conclusion if procedures are defined, concisely described and tested, feedback from these tests can be used as feedback to improve the procedures and routines. According to Kuvaja [44], it is common knowledge amongst the SMEs that full-scale assessment methods are only useful in large organizations and do not serve the SMEs appropriately. Nevertheless, small software development teams can improve their software processes beneficially as well as large organizations [64, 20].

To summarize, most of the previous reported research is focused on SPI models, assessment methods and case studies/projects applicable in SMEs. In contrast, this paper discusses and compares software process improvement methodologies for SMEs from a comparative view. To the best of our knowledge, the most related work is performed by Mishra and Mishra [55], who reviewed and compared various SPI methodologies on different significant attributes supported by various studies. This study extends previous work [55] in a substantial way. The major additions of this work over previous work are (1) Two recent SPI models: iFLAP and OWPL; (2) Additional attributes of comparison; (3) Revised and extended discussions; and (4) Rigorous and up-to-date literature review with latest references related with SPI in SMEs.

The remainder of this paper is organized as follows: The following section discusses software process improvement methodologies for SMEs. Later, these methodologies are compared and discussed. Finally, the paper concludes with limitations and directions for future research in this area.

### **3. Software Process Improvement Models for SMEs**

Any software process improvement plan requires a qualified statement about the current status of software development in the companies and a description of strengths and weaknesses identifying areas for improvement. On the basis of a literature survey, we have selected the following six SPI methodologies which have been implemented in SMEs. Due to limited resources and the size of the organizations, an extensive, formal assessment of the software practices following defined comprehensive approaches like the Capability Maturity Model [65], the ISO/IEC 90003:2004 [36], the TickIT scheme [81], Bootstrap [45] and IDEAL [42] model was not considered to be

necessary or appropriate in this context. It is also supported by Kautz [40]. Further MESOPYME objectives are similar to those of the IDEAL model [57] from the Software Engineering Institute (SEI).

Salient features of selected software process improvement methodologies for SMEs are discussed in this section.

### **3.1. OWPL: A Gradual Approach for Software Process Improvement in SMEs**

This approach is based on a three-stage software process improvement framework [6].

**Stage 1: Micro-assessment.** At this stage, a very simplified questionnaire called the micro-evaluation is used to collect information about the current software practices in small structures and to make people aware of the importance of software quality aspects [6]. The questionnaire covering 6 practice areas (quality management, customer management, subcontractor management, development and project management, product management, training and human resources management) is used by an assessor to interview a representative of the organization being evaluated [31]. The assessor must have sufficient expertise in software quality and software process improvement. The interviewee should have sufficient knowledge of the organization's IT activities [31]. Micro-evaluation results are presented in a report written by an evaluator. It includes general recommendations and specific improvement actions prioritized with respect to the organization's context and goal [31].

**Stage 2: OWPL evaluation.** As micro-evaluation's main goal is to give a first analysis, an in-depth analysis is achieved through this step. The OWPL evaluation covers 10 process areas and requires more than one person to be interviewed for each process [31]. Each process is described by its goal along with its description and a list of practices that compose the process [68]. The systematic description of a practice provides the goal, inputs, outputs and resources needed by that practice [68]. OWPL is designed to quickly identify practices related to software development in need of improvement, and also, to help draw a simple plan of action aiming at improving those practices and measuring the improvements [68]. Usually, an OWPL assessment only focuses on some of the processes, which are selected on the basis of a former assessment such as a micro evaluation or of a quick interview prior to conducting the OWPL assessment [68]. Focus on particular processes can also depend on the explicit demand of the enterprise to assess or on the results of the enterprise value chain analysis [68].

**Stage 3: SPICE assessment.** Bigger companies with medium/high quality level are eventually invited to undertake an ISO/IEC15504 or a CMMI evaluation if this appears appropriate [6].

### 3.2. Software Process Matrix (SPM) Model

This model helps the organization in finding the relative importance of software processes. For the high priority processes, the practices that need to be worked on are determined by Software Process Matrix (SPM). SPM is based on Quality Function Deployment (QFD). In QFD, the 'voice of the customer' is collected, and the relative importance of each customer requirement is measured. In the house of quality matrix, these requirements are used to identify design characteristics which have the greatest impact on customer requirements. Although QFD consists of many matrices, the main focus is often this matrix, as using it alone can have a significant effect on the product development process [26]. Using QFD, the software process model is treated as the customer where software processes are the customer requirements. These processes were identified from software process literature. The design characteristics are the practices which must be followed for the processes to be successful. These practices were also identified from the software process literature.

A crucial part of the development of the software process matrix was to identify the relationships between processes and practices. Those which are explicitly mentioned in the literature were easily identified. Using expert opinions and various statistical techniques, other relationships between processes and practices were identified, resulting in the development and verification of the software process matrix which was then validated in the industry.

For a small company to use any software process model to their advantage, it is imperative that the effort expended is minimal. The SPM provides them with a generic section that has been completed previously and can be used in their company. A questionnaire is provided to assess the current performance, planned future performance and importance to the company of every process. From the company's point of view, all they need to provide are the measurements for calculating the overall importance of the software process considering the following [69]:

- Current capability as assessed using a self-assessment questionnaire.
- Future capability as input from management.
- Importance of software process to the business.
- Competitive analysis
- Market leverage for company specific requirement e.g. ISO-certification.

Allowing management to choose whether or not to include figures for competitive analysis and market leverage allows flexibility within the model.

Practices with the highest values are the most important, and therefore it is suggested that these should be worked on first in the organization. From this, the priorities to be included in any software process improvement action plan are established and can help the organization to determine their improvement strategy. The complete SPM provides the organization with a ranked list of actions which can be input to their software process improvement strategy. This ranked list can be combined with cost figures and time-effective calculations thus taking these factors into account when determining the action plan for the organization.

### **3.3. An Approach for Software Process Establishment in Micro and Small Companies (ASPE-MS)**

An Approach for Software Process Establishment in Micro and Small Companies (ASPE-MS) is defined by integrating and adapting existing approaches [5,9,10,19,46,52,75] to the characteristics of small software companies. The principal phases of the approach are:

**Planning.** In the beginning, the process establishment is planned on a high level. Later on, during strategic analysis, the plan is revised, completed and adapted in accordance to the decisions made.

*Phase 1, Diagnosis.* The objective of this phase is to contextualize the organization and to obtain a high-level snapshot of the actual software process in place. Such a baseline can be established through a software process assessment using, e.g. MARES [29], an ISO/IEC 15504 conformant process assessment method tailored to small companies.

*Phase 2, Strategic analysis.* The objective of this phase is to specify the scope and to prioritize candidate processes to be established based on the results of the diagnosis and in accordance with the organization's business and improvement goals. This can be done by using, e.g. an adaptation of the SWOT (Strengths/Weaknesses/Opportunities/Threats) analysis technique [39] relating the importance of processes and their assessed/estimated capability.

*Phase 3, Definition.* The objective of this phase is to define the selected software process (es) in the form of a process guide in order to support process performers. Generally, the definition of the selected process(es) begins with the descriptive modeling of the actual process(es) in place. This activity is composed of a process familiarization phase and a detailed elicitation phase [9]. During the process familiarization phase an overview of the software process and its general structure, interaction and sequence is obtained and documented, for example, in a process flow diagram. In the next step, roles, competencies and responsibilities related to each activity are identified.

*Phase 4, Implementation.* First, the evaluation of the defined process(es) is planned in parallel to their implementation. This includes the revision and/or



definition of measures in order to monitor and determine the effectiveness and suitability of the process(es) and whether the expected benefits are achieved.

**Monitoring & Control.** The complete establishment of the process (es) is monitored and controlled. Therefore, data is collected and analyzed by the process engineer and assistant. If required, corrective actions are initiated and the plan is updated.

**Post-mortem.** Once a complete process establishment cycle is terminated, the process establishment approach is evaluated as a basis for continuous improvement. This is done by collecting and analyzing feedback from process performers, sponsors, and the process engineer and assistant in a feedback meeting or by questionnaires.

#### **3.4. PRISMS: An Approach to Software Process Improvement for Small to Medium Enterprises [7]**

PRISMS is an action research project, with a team of three researchers from Leeds Metropolitan University, working alongside managers and developers in participating companies advising and assisting with the planning and implementation of software process improvement programmes, over a three year period.

The key features of the process are:

- The existing informal process is examined, and if resources permit, an explicit model is created.
- In the PRISMS programme, the business goals are defined earlier by management. These goals drive much of the subsequent activity, especially the selection and prioritization of key process areas for improvement and the selection of measurements.
- A consultation exercise is carried out, involving all members of development teams. A brainstorming session, and/or questionnaire-based survey help the developer's team to take ownership of the SPI programme, and to be involved in the programme from the earliest stage.
- A tailored version of the CMM assessment is carried out by members of the research team, primarily to help identify key process areas (KPAs) for improvement.
- Using these inputs the KPAs for improvement are identified and prioritized. The main criteria here should be the extent to which the KPAs are likely to contribute to the identified business goals. One company has found a weighted selection approach of the type described by Martin [53] to be useful. The process/practice matrix approach described by Richardson [70] could also be used.

- Measurements are defined as an integral part of the SPI planning process. Managers are generally keen to have more precise ways of tracking key resource and quality indicators. The Goal Question Metric paradigm can be used to measure selected attributes based on the business goals defined for the SPI programme [12].
- The SPI plan is periodically reviewed, and there is provision to collect feedback from stakeholders.

The most important aspects of measurement for SPI programmes in smaller organization is that they should be simple to gather and interpret and that they should be used in planning and decision making. Simple automation can help to reduce the overheads associated with data collection and processing.

### **3.5. Improvement Framework Utilizing Light Weight Assessment and Improvement Planning (iFLAP)**

iFlap may be used to evaluate a single process area or it can be scaled so that any or all process areas can be assessed [67]. iFlap consists of three main steps [67]:

**Step 1: Selection.** It is important to select the right people as participants in the study from the organization so that assessment and improvement phases reflect the opinion of the entire staff [67]. Assessors should have a basic understanding of the organization before this step commences [67]. Workshops with representatives from the organization are conducted to help assessors understand the company [67]. The selection step is done in three major steps: first choosing projects to study, then selecting roles (both in project and line organizations) and finally appointing actual people that can represent each role [67].

**Step 2: Assessment.** In this step, improvement issues are gathered from the organization through interviews with practitioners. The improvement issues gathered are triangulated against project and process documentation for confirmation [67]. An assessment consists of two main parts: a project study, scrutinizing one or more projects, and a line study, that examines the relevant parts of the organization that are not part of a particular project [67].

**Step 3: Improvement Planning.** Firstly, selection of company's representatives, who are going to take part in this step, is done. Practitioners who took part in the assessment step can be retained for this step also but roles not directly associated with system development may be removed [67]. As the risks and cost of implementing all improvements at once are too high it is important that improvement effort focuses on a limited number of issues at a time taking evolutionary steps [67]. So, improvement issues are prioritized

based on factors such as business goals, practical restrictions and cost of implementation etc.[67]. Lastly, packaging of improvement issues is done in order to guide planning and implementation of changes in the process.

### 3.6. MESOPYME [14]

MESOPYME has been defined, taking into account a generic SPI model defined by ISPI [25] in four stages—whose objectives are similar to those of the IDEAL model [57] from the SEI. The key features of MESOPYME are as follows:

**Stage 1: Commitment to improvement.** Its objective is to obtain the support of senior management to carry out the improvement project.

**Stage 2: Software process assessment.** Its objective is to obtain strengths and weaknesses of the process assessed with respect to a software process model— CMM (Capability Maturity Model). From this assessment, processes (usually 1 to 3) to be improved are selected.

**Stage 3: Improvement solution.** Its objective is to provide the needed infrastructure to carry out improvement (in selected processes), and to create the plan to follow in order to define and implement improvement in these selected processes. The improvement solution stage is performed through the application of a generic set of components that we have called an Action Package. An Action Package is a general solution to a particular software process area that should be customized to a company, taking into account its business goals and assessment results. An action package is implemented in some selected pilot projects.

**Stage 4: Institutionalize.** Finally, improvement must be institutionalized.

## 4. Discussion

As these SPI methodologies are divergent in characteristics, therefore it is required to find out some significant but common attributes so that we can find a comparative view of all selected SPI approaches. Kautz et al. [42] concluded in their findings that first lesson for the small organizations, which wish to perform improvement activities, is that it makes sense to use a structural model to organize the process. They further suggested that the second lesson is that the model should be adjusted to the particular conditions of the organizations and the third lesson is that it makes sense to perform the improvement activities as a project with clearly assigned and documented roles, responsibilities and resources. Beyond the adjustment of general models (which is in fact a base for these approaches), Kautz [41]

points out the significance of factors to be studied further like management support and commitment, project planning and organization, education and training, assessment, monitoring and evaluation, staff involvement, support and knowledge transfer by external consultants, usability and validity of the introduced changes and cultural feasibility for process improvement in software SMEs. As SMEs have limited budgets and resources [40, 86, 8, 15], following factors are important providing the base in terms of comparison of SPI models and well supported by researchers:

- a) If it is based on already established SPI methods like CMMI then it may be better in the long run. Although this factor is not important right now, later organization may grow and may wish to achieve a specific established method like CMMI, SPICE etc. If the SPI model organizations are choosing right now is based on, for example CMMI, then it will be easy to switch over in future. Paulk [66] also suggests to begin with the “as is” process, not the “should be” process, to leverage effective practices and co-opt resisters. The “should be” process may or may not, be feasible in the given culture and environment.
- b) There are two key questions: where am I and what needs to be improved? and how to improve it? If a SPI model answers both these questions successfully, then it is easier for the organization to use and implement it.
- c) There are three major steps in most of the SPI models: Assessment step, SPI planning by analyzing data collected in assessment step, implementation of SPI plan. If all these three steps are clearly defined in terms of what needs to be done in each step and who are the people involved in each step then such model can be easily applied in an organization.
- d) Whether it takes into consideration specific needs of the organization then it is better for the organization as also supported by Ginsberg [27] and Ade [4] that processes need to be tailored to the needs of the project. Although standard processes provide a foundation, each project will also have unique needs and unreasonable constraints on tailoring which can lead to significant resistance to the following process [66]. As Hoffman [35] expresses it: “Don’t require processes that don’t make sense”.
- e) If it provides some flexibility to the organization like choice of different methods for assessment etc. then it is better. It is also supported by Glass [28] that these processes should combine the need for flexibility and creativity. Further Richardson [69, 70] found flexibility as a significant characteristic for software process and included it in her proposed model. Alexandre et al. [6] also observed that because of their comparative size, SMEs are not able to impose their methodological approach, even if it is a very good one. They have to follow the methodological guidelines of their (larger) customers. As a consequence, flexibility becomes a vital concern.
- f) Whether it is continuous or staged? An organization may choose one over another. Continuous representation allows an organization to select the order of improvement that best meets the organization’s business objectives and mitigates the organization’s areas of risk. On the other

hand, staged representation provides a proven sequence of improvements, beginning with basic management practices and progressing through a predefined and proven path of successive levels, each serving as a foundation for the next [77].

- g) Commitment of senior management in SPI initiative is very important. Lack of senior management commitment is recognized by [2,22,24,86] as a major bottleneck to the success of SPI initiatives, but interestingly Cater-Steel et al. [15] found in their study that in most small organizations, the business operator is often involved in all aspects of the business and would therefore instigate the SPI and participate heavily in it. Dangle et al. [21] also supported in their case study to properly ensure that process improvement resources are committed while Mikaliunas and Reingardtas [54] recommended getting full support and commitment from company top management.
- h) Involvement of software development team members from the start is very important. Their views should be considered while deciding what needs to be improved. It may help in securing their confidence and commitment in SPI initiative. Otherwise they may resist SPI initiative later on. Commitment from management is one of the most crucial factors for successful SPI. However, it is not enough and there must be commitment and involvement by middle management, staff and developers [67].
- i) Whether it requires SME's staff, who will take part in SPI initiative, to have prior experience in this field. If it does, it may not be suitable as SMEs have difficulty in recruiting and retaining experienced staff. An association was found between assessed capability levels and the experience and education level of staff employed by the assessed firms [16].
- j) Whether it requires the need to take the help of external experts. If this is the case, it might be difficult for the organization as they have to bear the extra cost. In this context it is important to note recommendations by Cater-Steel et al. [15] that firms are advised to draw on expertise of external assessors/consultants as mentors. However, research has shown that small firms are averse to consultants and reluctant to seek external help [18]. This was further supported by Hall, Rainer and Baddoo [32] that companies did not highly value the input of external consultants. Therefore, the assessors, as external consultants, need to develop a relationship with the developers in small firms [15].
- k) Whether roles and responsibilities are clearly assigned to all people taking part in SPI initiative. Also, if they need training, it should be provided. Both these factors are important for any successful SPI initiative as mentioned by Kautz [41]. Cater-Steel et al. [15] advised that SPI action plan, derived from the assessment recommendations, should differentiate between short term objectives achievable within evaluation time-scale and longer term improvement initiatives.
- l) If a tool can be used for self assessment, it will be easier to assess the current status and to determine the areas which need to be improved. Additionally, more people can be involved during this phase without much substantial effort. Today, the tool is used in all projects and all change

requests are handled through the tool following the defined process and process performers found that tools provide a good overview are simple to use and provide easy access to the process guides [84]. They found in their case studies that the process guides continuously evolve and are updated. Therefore, their availability in electronic form is essential in order to enable efficient maintenance. Russ and McGregor [71] also suggested that automation of process monitoring and evaluation will further free team members to focus on the project's goal of producing quality software.

- m) Data collection and evaluation is integral in any SPI initiative. It can be difficult for software practitioners to do this if an organization does not have a special team to do this task. Use of tools for this purpose can make the job of software practitioners easy in this case. Wangenheim et al. [84] observed that a beneficial way of implementing a process can be indirectly by adapting support tools or workflow management tools to the process definition, using the process guide itself as a basis for the adaptation or development of such tool support and keeping it as an additional source of information.
- n) Sometimes origin of an SPI method is also important. A particular SPI method originated in a particular country is tested in the software development organizations of that country. Although, due to the emergence of global standards in software development, organizations all over the world are similar to each other in terms of platforms, technical tools and other things they are using. Still cultural factors play an important role, and one SPI initiative which was tested successfully in one country may not get equal success in another country. This is also supported by Biro et al. [11] that national culture affects the process improvement methods. Additionally, people who developed a particular SPI model may be available for help for the organizations situated in their country. Important factors which influence the selection of the method for software process improvement are geographical characteristics, corporate culture and business objective of the organization [80].

These models for SMEs except ASPEC-MSD and iFLAP are based on some existing methods like CMMI, SPICE, QFD, GQM etc. These approaches are adapted and simplified either by incorporating a matrix (in SPM model) or process guides (in ASPE-MSD) or action packages (in MESOPYME) or gradual improvement using successively different assessment methods (OWPL) so that they can be used by these organizations.

One key point is that all methods consider business objectives of the organization while making the SPI plan. Moreover, these methods are flexible enough that although methods for identifying and prioritizing areas of improvement are suggested organizations can also choose any other method. Furthermore, organizations have the flexibility to select processes more important to them for SPI plan. These methods not only detect what needs to be improved but also provide the roadmap for how to improve it.

In MESOPYME, there is a step specifically to obtain support of senior management at the beginning of SPI initiative while in PRISMS and SPM methods; management is involved in assessment step. In ASPE-MSD, director(s) of the company are involved in specifying the scope and prioritize candidate processes in SPI planning stage. In OWPL and iFLAP methods, management is not involved specifically in any step but it is mentioned that management support from the beginning is important for any successful SPI initiative.

Software practitioners are involved from the beginning in all methods. They take an active participation during self assessment. All practitioners' views, regarding which processes need to be improved, were taken into consideration [55].

All methods are continuous in nature. As far as practitioner's knowledge level is concerned, OWPL, ASPEC-MSD, iFLAP and MESOPYME do not require much experience at assessment and SPI planning stages as most of the work at these two stages is done by an external quality expert while SPM model needs much knowledge and experience to assess current capabilities of the process. PRISMS uses a web-based tool for the assessment stage. Use of an external quality expert to carry out these two stages may be extra burden for SME. Every model requires the practitioner's knowledge level to be considerably high to carry out SPI plan implementation stage. SMEs generally do not have people dedicated for quality work alone. A person has many roles in these organizations, for example people who are doing software development are also responsible for SPI initiative. These individuals may or may not have experience dealing with SPI initiative so it may not be easier for them to use any of these models without the help of some external consultant [56].

These SPI models are specifically developed for SMEs as these organizations do not have the resources and cannot bear the cost to implement CMMI, SPICE etc [79]. In this context it is important to note some outcomes. For instance, SPIRE results indicated that "of the small software development units who applied to be involved in SPIRE, 27% dropped out. The most common reasons given were resource or funding problems" [78]. Wieggers says [85], "the most common point of failure in SPI is lack of follow-through into action planning and action plan implementation." Also performance of these activities is expensive- the yearly cost of improvement is \$245,000 [33], and time consuming – a full process improvement cycle could take between 18 and 24 months [87]. Moreover, this is more difficult to perform in SMEs because they do not have the resources to carry out improvement implementation [14]. For these reasons, this SPI approach is restricted to large organizations but Dyba [23] found that small organizations can and do implement SPI elements as effectively as large organizations, and in turn, achieve high organizational performance. Therefore, this indicates that SPI can be used as a competitive advancement strategy for both small and large software organizations. But whether a small or medium scale organization can implement these methods without the help of some external quality consultant is yet to be proven.

**Table 1.** Comparison of various software process improvement models (OWPL, SPM Model, ASPE-MS, PRISMS, iFLAP, MESOPYME) for small and medium enterprises

<b>SPI Models→ Criteria ↓</b>	<b>OWPL</b>	<b>SPM Model</b>	<b>ASPE-MS</b>	<b>PRISMS</b>	<b>iFLAP</b>	<b>MESOPYME</b>
<b>Based on</b>	SPICE	QFD	Many existing approaches	CMM and GQM	Inductive method	CMM
<b>Key Question</b>	Where am I? What needs to be improved? How to improve?	What needs to be improved? How to improve?	Where am I? What needs to be improved? How to improve?	Where am I? What needs to be improved? How to improve?	Where am I? What needs to be improved? How to improve?	Where am I? What needs to be improved? How to improve?
<b>What is new?</b>	Three assessment methods (micro-evaluation, OWPL evaluation, SPICE assessment) can be used in succession to achieve process maturity.	SPM (software process matrix) that identifies practices needed for software processes to be improved.	Iterative-Incremental approach for assessment, identification and implementation of SPI plan.	Adapting CMM by incorporating business objectives with the help of GQM paradigm	Applicable even if the organization does not exhibit extensive maturity. It can be used to assess and improve a single, many or all process area at a time.	Emphasis on SPI implementation step with the help of action packages developed by problem domain experts.
<b>Assessment Step Who Does it?</b>	An Assessor (Quality expert) and a representative of	Representatives of the organization	A PE (process engineer) typically an external consultant and an	Assessors (Quality experts) and representative	Assessors (Quality experts) and selected representatives	Assessors (Quality experts)



Software Process Improvement in SMEs: A Comparative View

	the organization.		Assistant PE (representative from the organization).	s of the organization.	of the organization	
<b>Assessment Step</b> How it is done?	An Assessor interviews a representative of organization to assess the current capability of the organization.	Current capabilities are assessed by a self-assessment questionnaire. Other information (future capabilities, importance of the software process to the business etc.) is also provided by the organization.	First assistant PE is trained and then diagnosis of current capabilities is done using an assessment method tailored for small companies.	An awareness and business case workshop is conducted to identify a road map for process improvement followed by a series of assessment interviews based on modified version of CMM assessment questionnaire. This step can be carried out using a web-based self assessment	First participants from the organization are selected. Then workshops are carried out so that assessors can understand the organization and vice-versa. Improvement issues are gathered from the organization through interviews with practitioners and getting information from projects and other process documentation.	Strengths and weaknesses of the process with respect to a software process model—CMM (Capability Maturity Model) are identified. From this assessment, processes (usually 1 to 3) to be improved are selected.

				tool.		
<b>SPI Planning</b> Who Does it?	Evaluator (experts of quality)	Evaluator	A PE (process engineer) typically an external consultant and an Assistant PE (representative from the organization)	Assessors (Quality experts)	Assessors (Quality experts) and selected representatives of the organization	Assessors (Quality experts) and a SEPG (Software Engineering Process Group) formed within the organization.
<b>SPI Planning</b> How it is done?	Information collected through interview is evaluated and a list of actions/recommendations are prioritized with respect to the organization's context and goal	First prioritized list of processes for software process improvement according to the business objectives and other factors are made. Then a ranked list of actions is made with the help of SPM to improve above mentioned processes.	Prioritized list of candidate processes is made according to the diagnosis, business objectives and improvement goals. Later these processes are defined in form of a process guide.	KPAs for improvement are identified based on information collected in the assessment phase (current capabilities, business goals given by the management and after consultation with developers). Later process improvement	The improvement issues gathered through interviews are triangulated against project and process documentation for confirmation. Then, workshops are conducted to identify prioritized small improvement packages based on the need of the organization, practical	Action package for each process area consisting of action plan, infrastructure needed, techniques, tools, metrics etc., is revised according to the business goals and assessment results. Training is provided to SEPG group about the action packages related with the processes to be

Software Process Improvement in SMEs: A Comparative View

				plan, consisting of prioritized list of KPAs, is made.	restrictions and cost of implementation.	improved. Pilot projects are selected.
<b>Implementation of SPI plan</b> Who Does it?	A team, consists of people from the organization with roles and responsibilities, carry out this phase.	Practitioners within the organization	A PE (process engineer), an Assistant PE and other representatives from the organization	Practitioners from the organizations	Selected representatives of the organization	SEPG group formed within the organization
<b>Implementation of SPI plan</b> How it is done?	Actions/recommendations from previous phases are implemented	List of actions made for SPI in the previous phase are implemented.	SPI plan, consisting of prioritized list of candidate processes to be improved, is implemented with the help of candidate processes' process guide and evaluated continuously.	Process improvement plan is implemented.	Representatives from the organization are selected for this phase. Solutions related with an improvement issue are implemented by practitioners.	Action packages related with the processes to be improved are implemented in some pilot projects. Finally improvement is institutionalized.
<b>Flexibility</b>	Flexible. One particular process area can be	Flexible. Company is not required to include all	Flexible. Methods for assessing current capabilities and to prioritize	Flexible. Methods to identify and prioritize	Flexible. Improvement issues gathered from the	Flexible. Can be tailored to the specific need of an organization.

Deepti Mishra and Alok Mishra

	evaluated at a time. Three assessment methods can be used in succession.	factors of measurement for overall importance of software processes. Only processes important for company are considered for improvement.	processes are suggested but organization can use other methods also. Only processes important for the company need to be considered for improvement.	KPAs for improvement are suggested but organizations can use other methods also.	assessment phase can be implemented in an order suitable to the organization's needs, available resources etc.	
<b>Continuous/ Stage</b>	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous
<b>Commitment from higher Management</b>	Yes	Yes, Management provides important data during the assessment stage.	Yes, director(s) of the company are involved to specify the scope and priorities candidate processes in SPI planning stage.	Yes, a top level manager is involved in the assessment stage.	Yes	Support of senior management is obtained at the beginning of SPI initiative.
<b>Involvement of software development team members from the very beginning</b>	Yes	Yes, They give information about processes which need to be improved.	Only one person (process engineer or assistant PE) is involved during assessment. Others are involved during implementation.	Yes, They give information about which processes needs to be improved.	Yes	Not mentioned. It is not clear that who decides which process needs to be improved. They are involved during

Software Process Improvement in SMEs: A Comparative View

						implementation.
<b>Practitioner's knowledge level</b>	Needs considerable experience to implement SPI plan.	Needs considerable experience to assess current capabilities and planned future performance of a software process and importance of the process to the company	Needs considerable experience to assess current capabilities, SPI planning and implementation. If the organization lacks required expertise, an external expert (PE) is used with an employee acting as an assistant to enable knowledge transfer.	Needs considerable experience to assess current process, identify KPAs for improvement. Also, development and implementation of process improvement plan requires experienced person.	Needs considerable experience	Doesn't need much experience as action packages are developed by domain experts according to the organizations' business goals and current capabilities.
<b>Roles, responsibilities and training</b>	Assessors interview selected practitioners to assess the organizations and then make prioritized list of actions/recommendations. For	Only quality assurance engineer is responsible for SPI. Other roles are not mentioned.	Roles, competencies and responsibilities related to each activity are defined and an assistant process engineer is given training, if required.	The only role is process improvement champion who is responsible for implementation of improvement actions. Other	Practitioners are selected and their roles and responsibilities are identified before initiating each step. Workshops are conducted to give assessors	Roles and responsibilities are established for every improvement initiative. Special training is given before implementation.

Deepti Mishra and Alok Mishra

	the implementation phase, a team consisting of people from the organization itself is selected with defined roles and responsibilities.			roles are not mentioned.	(quality expert) information about organization and vice-versa.	
<b>Tool for self assessment</b>	Not mentioned	Not mentioned	Not mentioned	Yes, based on a modified and customizable version of questionnaire.	Not mentioned	Not mentioned
<b>Automated data collection for measurement and evaluation</b>	Not mentioned	Not mentioned	Not mentioned for data collection. Tool support for process selected for improvement.	Can be done if organization is ready to devote resources.	Not mentioned	Not mentioned for data collection. Tool support for process selected for improvement.
<b>Outcome</b>	A prioritized list of actions/recommendations with respect to the organization's	Ranked list of actions for software process improvement.	Process guide for processes selected for improvement that contains entry/exit criteria, techniques	Process improvement plan and later revised process model after	Prioritized small improvement packages based on need of the organization, practical	Action package containing action plan, infrastructure needed, techniques,

Software Process Improvement in SMEs: A Comparative View

	context and goal		and tools to use, examples, required infrastructure, checklists etc.	implementation of process improvement plan.	restrictions and cost of implementation.	tools, metrics etc. according to the business goals and assessment results for each process area.
<b>Constraints</b>	Assessors (quality experts) are needed to gather information about current situation, analyzing the collected information and making a prioritized list of actions/recommendations with respect to the organization's context and goal	Measuring the importance of software processes needs considerable experience.	Considerable experience is needed for process assessment, process prioritization and development of process guides.	Considerable experience is needed to identify current process model and process improvement plan.	Assessors (quality experts) are needed to select practitioners from the organization, gather information about current situation, identifying improvement issues and making prioritized small improvement packages	Needs assistance of domain expert to develop action packages according to the organizations' business goals and current capabilities.
<b>Origin</b>	Belgium	Ireland	Brazil	Britain	Sweden	Spain

## 5. Conclusion

In this paper, we have analyzed six significant and recent SPI methodologies for SMEs and also presented and discussed their significant characteristics from comparative perspective. Each method has its advantages and limitations. Enterprises should choose the particular process methodology with reference to their business goals, models, characteristics and resource limitations. These methodologies can be tailored and customized according to the specific organizational environment. A limitation of our study is that numerical experimentation could not be included as empirical data is available in only very few case studies related to SPIs in SMEs. Further, some additional attributes for comparison may come up due to new software process improvement and assessment models in future.

Future work in this regard is suggested to perform comparative case studies and empirical validation in real software development environment which can provide fresh insights. It would be interesting to study the impact, efforts and comparison of these approaches on SPI in SMEs. Dyba [23] also supported that future studies should focus on the specific needs of small software organizations in more depth; for example, through longitudinal, multiple case studies. Further research should be related to the study of new and improved measures of SPI success, comparison of measurement instruments, and validation of SPI success measures [23]. These further experiences will move towards tailoring software engineering methods and improvement strategies [23].

## Acknowledgement

We would like to thank Editor-in-Chief and two referees for their valuable comments and directions to improve the quality of this paper. We would also like to thank David J.W. Boddington for nicely editing the manuscript.

## 6. References

1. Abbott, J.J.: Software Process Improvement in a Small Commercial Software Company. Proceedings of the 1997 Software Engineering Process Group Conference, San Jose, CA, 17-20. (1997)
2. Abrahamsson, P.: The role of commitment in software process improvement. PhD, University of Oulu. (2002)
3. Acuna, S.T., Antonio, A.D., Ferre, X., Lopez, M., Mate, L., Estero, S.: The Software Process: Modeling, Evaluation and Improvement. World Scientific Publishing Company, Argentina. (2000)
4. Ade, R.W., Bailey, J.P.: CMM Lite: SEPG tailoring Guidance for Applying the Capability Maturity model for Software to Small Projects. Proceedings of the 1996



- Software engineering Process Group Conference: Wednesday Papers, Atlantic City, NJ, 20-23. (May 1996)
5. Ahonen, J.J., Forsell, M., Taskinen, S.K.: A modest but practical software process modeling technique for software process improvement. *Software Process Improvement and Practice*, Volume 7, Number 1, 33-44. (2002)
  6. Alexandre, S., Renault, A., Habra, N.: OWPL: A Gradual Approach for Software Process Improvement in SMEs. *Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA06)*, 328-335. (2006)
  7. Allen, P., Ramachandran, M., Abushama, H.: PRISMS: an Approach to Software Process Improvement for Small to Medium Enterprises. *Proceedings of the Third International Conference On Quality Software (QSIC '03)*, IEEE, Dallas, Texas, USA, November 6 - 7, 211-214. (2003)
  8. Bae, D-H.: Panel: Software Process Improvement for Small organizations. *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, IEEE Computer Society. (2007)
  9. Becker-Kornstaedt, U.: Towards Systematic Knowledge Elicitation for Descriptive Software Process Modeling. *Proceedings of Third International Conference on Product Focused Software Process Improvement (PROFES)*, Germany, 312-325. (2001)
  10. Becker-Kornstaedt, U., Hamann, D., Verlage, M.: Descriptive Modeling of Software Processes. *IESE-Report 045.97/E*, Fraunhofer Institute IESE, Germany. (1997)
  11. Biro, M., Messnarz, R., Davison, A.G.: The impact of national cultural factors on the effectiveness of process improvement methods: The third dimension. *Software Quality Professional*, 4(4), 34-41. (2002)
  12. Briand, L., Differding, C., Rombach, H.D.: Practical Guidelines for Measurement-Based Process Improvement. *Software Process: Improvement and Practice*, Vol. 2, 253-280. (1996)
  13. Broadman, J.G., Johnson, D.L.: What small businesses and small organizations say about the CMM. *Proceedings of the 16<sup>th</sup> International Conference on Software Engineering*, IEEE Computer Society Press, Sorrento, Italy, 331-340. (1994)
  14. Calvo-Manzano, J.A., Agustin, G.C., Gilabert, T.S.F., Seco, A.D.A., Sanchez, L.Z., Cota, M.P.: Experiences in the Application of Software Process Improvement in SMES. *Software Quality Journal*, 10, 261-273. (2002)
  15. Cater-Steel, A., Toleman, M., Rout, T.: Process improvement for small firms: An evaluation of the RAPID assessment-based method. *Information and Software Technology*, 48, 323-334. (2006)
  16. Cater-Steel, A.P.: Low-rigour, rapid software assessments for small software development firms. *Proceedings of Australian Software Engineering Conference (ASWEC)*, Melbourne, 368-377. (2004)
  17. Christie, M. et al.: Software Process Automation: Interviews, Survey and Workshop results. *Technical Report CMU/SEI-97-TR-008*, Carnegie Mellon University/SEI. (1997)
  18. Cragg, P.B.: Benchmarking information technology practices in small firms. *European Journal of Information Systems*, 11(4), 267-282. (2002)
  19. Curtis, B., Kellner, M.I., Over J.: Process modeling. *Communications of the ACM*, 35(9), 75-90. (1992)
  20. Damele, G., Bazzana, G., Giunchi, M., Caielli, G., Maiocchi, M., Andreis, F.: Quantifying the benefits of software process improvement in Italtel Linea UT Exchange. *Proc. Of International Switching Symposium*, Berlin, April. (1995)

21. Dangle, K.C., Larsen, P., Shaw, M., Zelkowitz, M.: Software Process Improvement in Small Organizations: A Case Study. *IEEE Software*, 22(6), 68-75. (2005)
22. Debou, C., Kuntzmann-Combelles, A.: Linking software process improvement to business strategies: experiences from industry. *Software Process: Improvement and Practice*, 5(1), 55-65. (2000)
23. Dyba, T.: Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context. Proceedings of the 9th European software engineering conference (ESEC/FSE' 03) September 1-5, Helsinki, Finland, 148-157. (2003)
24. El Emam, K., Goldenson, D.R., McCurley, J., Herbsleb, J.: Success or Failure? Modeling the Likelihood of Software Process Improvement. International Software Engineering Research Network (ISERN) ISERN-98-15. (1998)
25. ESSI, IBERIA, LAE.: SPIE: Software Process Improvement and Experimentation. ESSI Project: No 10344, February. (1994)
26. Fortuna, R.M.: Beyond quality: Taking SPC upstream. *Quality Progress*, June 1988, pp. 23-28. (1988)
27. Ginsberg, M., Quinn, L.: Process tailoring and the Software capability maturity model. Software Engineering Institute, CMU/SEI-94-TR-024, November. (1995)
28. Glass, R.L.: *Software Creativity*. Englewood Cliffs, New Jersey: Prentice- Hall. (1995)
29. Gresse von Wangenheim, C., Anacleto, A., Salviano, C.F.: Helping Small Companies Assess Software Processes. *IEEE Software*, 23(1), 91-98. (2006)
30. Gresse, C., Punter, T., Anacleto, A.: Software measurement for small and medium enterprises –A Brazilian-German view on extending the GQM method. Proceeding of 7<sup>th</sup> international conference on empirical assessment in software engineering EASE, Keele, UK. (2003)
31. Habra, N., Alexandre, S., Desharnais, J., Laporte, C. Y., Renault, A.: Initiating software process improvement in very small enterprises. *Information and Software Technology*, 50, 7-8, 763-771. (2008)
32. Hall, T., Rainer, A., Baddoo, N.: Implementing software process improvement: an empirical study, *Software Process: Improvement and Practice*, 7(1), 75-81. (2002)
33. Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., Zubrow, D.: Benefits of CMM-based Software Process Improvement: Initial Results. Technical Report: CMU/SEI-94-TR-013, August, Pittsburgh. (1994)
34. Herrera, E.M., Trejo Ramirez, R.A.: A Methodology for self-diagnosis for software quality assurance in small and medium-sized industries in Latin America. *The Electronic Journal on Information Systems in Developing Countries*, 15, 4, 1-13. (2003)
35. Hoffman, L.: Small Projects and the CMM. in Key Practices to the CMM: Inappropriate for Small projects? Panel, Rita Hadden moderator, Proceedings of the 1998 Software Engineering Process Group Conference, Chicago, IL, 9-12 March. (1998)
36. ISO/IEC 90003:2004: Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software. (2004) available at [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=35867](http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=35867)
37. Jalote, P.: *An Integrated Approach to Software Engineering*. 2nd edition, Narosa Publishing House. (2000)
38. Johnson, D. L., Brodman, J.G.: Applying the CMM to Small Organizations and Small Projects. Proceedings of the 1998 Software Engineering Process Group Conference, Chicago, IL. 9-12 March. (1998)

39. Johnson G., Scholes K., Sexty R.W.: Exploring Strategic Management. Prentice Hall, Englewood Cliffs, NJ. (1989)
40. Kautz, K.: Software Process Improvement in Very Small Enterprises: Does it Pay Off ?, *Software Process – Improvement and Practice*, 4(4), 209-226. (1998)
41. Kautz, K.: Making Sense of Measurements for Small Organizations. *IEEE Software*, Vol. 16, No. 2, pp. 14-20. (1999)
42. Kautz, K., Hansen, H.W., Thaysen, K.: Applying and Adjusting a Software Process Improvement Model in Practice: The use of the IDEAL Model in a Small Software Enterprise. *Proceedings of ICSE 2000, Limerick, ACM Press*, 626-633. (2000)
43. Kurniawati, F., Jeffery R.: The Long-term effects of an EPG/ER in a small software organization. *Proceedings of the Australian Software Engineering Conference, Australia*, 128-136. (2004)
44. Kuvaja, P., Palo, J., Bicego, A.: TAPISTRY- A Software Process Improvement Approach Tailored for Small Enterprises. *Software Quality Journal*, 8, 149-156. (1999)
45. Kuvaja, P., Simila, L., Krzanik, L., Bicego, A., Koch, G., Sankonen, S.: *Software Process Assessment and Improvement: the BOOTSTRAP Approach*. Blackwell Publishers, Oxford, UK. (1994)
46. Kellner, M.I., Becker-Kornstaedt, U., Riddle, W.E., Tomal, J., Verlage, M.: *Process Guides: Effective Guidance for Process Participants*. *Proceedings of the Fifth International Conference on the Software Process: Computer Supported Organizational Work, USA*, 11-25. (1998)
47. Laporte, C.Y., Alexandre, S., O'Connor, R.V.: A Software Engineering Lifecycle Standard for Very Small Entrprises. *Euro SPI 2008, CCIS*, 129-141. (2008)
48. Larsen, E.A., Kautz, K., Quality Assurance and software process improvement in Norway. *Software Process – Improvement and Practice*, 3, 71-86.(1997)
49. Makinen, T., Varkoi, T.: Assessment Driven Process Modeling for Software Process Improvement. *PICMET 2008 Proceedings*, 27-31 July, Cape Town, South Africa. (2008)
50. Mcfall, D., Wilkie, F-G, Mc Caffery, F., Lester, N., Sterritt, R.: *Software Processes and Process Improvement in Northern Ireland*. 16th International Conference on Software & Systems Engineering and their Applications. (2003)
51. McCaffery, F., Taylor, P.S., Coleman, G.: Adept: A Unified Assessment Method for Small Software Companies. *IEEE Software*, 24(1), 24-31. (2007)
52. Madhavji, N.H, Holtje, D., Hong, W., Bruckhaus, T.: Elicit: A Method for Eliciting Process Models. *Proceedings of the Third International Conference on the Software Process, SA*, 111-122. (1994)
53. Martin, S.: *Business Process Improvement*. McGraw Hill. (2002)
54. Mikaliunas, G., Reingardtas, M.: *Software Process Improvement in Lithuania – AB ALNA Case Study*. *Information Technology and Control*, Vol. 34, No. 2A, 215-218. (2005)
55. Mishra, D., Mishra, A.: *Software Process Improvement Methodologies for Small and Medium Enterprises*. A. Jedlitsschka and O. Salo (Eds.), *PROFES 2008: LNCS 5089*, 273-288. (2008)
56. Mishra, D., Mishra, A.: *A Study of Software Process Improvement in Small and Medium Organizations*. In *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, Hanna Oktaba and Mario Piattini, 141-159, IGI Global, USA. (2008)
57. McFeeley, Bob.: *IDEALSM: A user's guide for software process improvement*. Handbook CMU/SEI-96-HB-001, Software Engineering Institute, Carnegie Mellon University. (1996)

58. Ministerio da Ciencia e Tecnologia: Quality and Productivity of the Brazilian Software Sector (in Portuguese). Ministerio da Ciencia e Tecnologia, Brazil. (2001) (Government report – No Author)
59. Mintzberg, H.: Structures in Fives: Designing Effective Organizations. Prentice Hall International, Englewood Cliffs, NJ, USA. (1993)
60. Moe, N.B., Dingsoyr, T., Johansen, T.: Process guides as Software Process Improvement in a small company. Proceedings of the EuroSPI Conference, Germany. (2002)
61. Nonaka, I.: A dynamic theory of organizational knowledge creation. *Organization Science*, 5, 14-37. (1994)
62. Oktaba, H., Garcia, F., Piattini, M., Ruiz, F., Alquicira, C.: Software Process Improvement: The Competisoft Project, *IEEE Computer*, 40(10), 21-28. (2007)
63. Paquin, S.: Struggling with the CMM: Real Life and Small Projects. in “Key Practices to the CMM: Inappropriate for Small Projects?” panel, Rita Hadden moderator, Proceedings of the 1998 Software Engineering Process Group Conference, Chicago, IL, 9-12 March. (1998)
64. Paulish, D.J.: Case studies of software process improvement methods. SEI Technical Reports, CMI SEI-93-TR-26. (1993)
65. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model version 1.1. *IEEE Software*, 10(4), 18-27. (1993)
66. Paulk, M.C.: Using the Software CMM in Small Organizations. Joint 1998 Proc. Pacific Northwest Software Quality Conf. And 8th International Conf. Software Quality, PNSQC/Pacific Agenda, 350-361. (1998) available at [www.pnsqc.org/proceedings/pnsqc98.pdf](http://www.pnsqc.org/proceedings/pnsqc98.pdf).
67. Pettersson, F., Ivarsson, M., Gorschek, T., Öhman, P.: A practitioner’s guide to light weight software process assessment and improvement planning. *The Journal of Systems and Software*, 81, 972-995. (2008)
68. Renault, A., Habra, N., Niyitugabira, E., Alexandre, S., Deprez J.-C.: OWPL model: Software process improvement for VSE, SME and Low maturity Enterprises. Technical Report, CETIC\_FUNDP, Belgium. (2006) available at [http://www.cetic.be/IMG/pdf/OWPL-Model\\_V1-2-2-EN-V22.pdf](http://www.cetic.be/IMG/pdf/OWPL-Model_V1-2-2-EN-V22.pdf)
69. Richardson, I.: SPI models: What characteristics are required for small software development companies?. *Software Quality Journal*, 10, 101-114. (2002)
70. Richardson I.: Software Process Matrix: a Small Company SPI Model. *Software Process: Improvement and Practice*, Vol. 6, 157-165. (2001)
71. Russ, M.L., McGregor, J.D.: A Software Development Process for small projects. *IEEE Software*, 17(5), 96-101. (2000)
72. Scott, L., Carvalho, L., Jeffery, R., D’Ambara, J., Becker-Kornstaedt, U.: Understanding the use of an electronic process guide. *Information and Software Technology*, 44(10), 601-616. (2002)
73. Scott, L., Jeffery, R., Carvalho, L., D’Ambara, J., Rutherford, P.: Practical Software Process Improvement-The IMPACT Project. 13th Australian Software Engineering Conference (ASWEC’01), 182-189. (2001)
74. Scott, L., Jeffery, R., Becker-Kornstaedt, U.: Preliminary results of an industrial EPG evaluation. Proceedings of Fourth ICSE Workshop on Software Engineering over the internet, Canada. (2001)
75. Scott, L., Zettel, J., Hamann, D.: Supporting Process Engineering in Practice: An Experience Based Scenario. Proceedings of the Conference on Quality Engineering in Software Technology (CONQUEST), Germany. (2000)
76. Software Engineering Institute: Improving processes in small settings. A research initiative of the SEI’s IPRC. (2006) available at <http://www.sei.cmu.edu/iprc/iprc-overview.pdf>

77. Software Engineering Institute: Capability Maturity Model@Integration (CMMISM) Version 1.1. (2001) available at <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr004.pdf>
78. SPIRE: Software Process Improvement in Regions of Europe. European Analysis Report v2.0, ESSI Project: No. 23873, Dissemination action, April. (1999) available at <http://www.cse.dcu.ie/spire>
79. Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., Murphy, R.: An exploratory study of why organizations do not adopt CMMI. *The Journal of Systems and Software*, 80(6), 883-895. (2007)
80. Suwanya, S., Kurutach, W.: An Analysis of Software Process Improvement for Sustainable Development in Thailand. 8<sup>th</sup> International IEEE Conference On Computer and Information Technology, CIT 2008. (2008)
81. TickIT: A Guide to software quality management system construction and certification using EN29001. Issue 2.0, UK Department of Trade and Industry, London, UK. (1992)
82. Trudel, S., Lavoie, J-M, Pare, M-C, Suryan, W.: PEM: The small company-dedicated software process quality evaluation method combining CMMI and ISO/IEC 14598. *Software Quality Journal*, 14(1), 7-23. (2006)
83. Valtanen, A., Ahonen, J.J.: Big Improvements with small changes: Improving the Processes of a Small Software Company. *Profes*, 2008, LNCS 5089, 258-272. (2008)
84. Wangenheim, C.G. V., Weber, S., Hauck, JCR, Trentin, G.: Experiences on establishing software processes in small companies. *Information and Software Technology*, 48(2006), 890-900. (2006)
85. Wieggers, K.E., Sturzenberger, D.C.: A Modular Software Process Mini-Assessment Method. *IEEE Software*, 17(1), 62-69. (2000)
86. Wilson, D.N., Hall, T., Baddoo, N.: A Framework for evaluation and prediction of software process improvement success. *The Journal of Systems and Software*, 59(2001), 135-142. (2001)
87. Zahran, S.: *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley. (1998)
88. Zeineddine, R., Mansour, N.: SQIMSO: Quality Improvement for Small Software Organizations. *Journal of Computer Science*, 1(3), 316-322. (2005)

**Deepti Mishra** is an Assistant Professor of Computer Engineering at Atilim University, Ankara Turkey. She has earned her Ph.D. in Computer Science (Software Engineering) and Masters in Computer Science and Applications. Her research interests include software process improvement, software quality, requirement engineering and databases. She has published many research papers and book chapters at international and national levels. She is recipient of department of information technology scholarship.

**Alok Mishra** is Associate Professor of Computer and Software Engineering at Atilim University, Ankara Turkey. He had his Ph.D. in Computer Science (Software Engineering) besides dual Masters in Computer Science & Applications and Human Resource Management. His areas of interest and research are software engineering, information system, information & knowledge management and object oriented analysis & design. He had

Deepti Mishra and Alok Mishra

extensive experience of distance and online education related to computers and management courses. He has published articles, book chapters and book-reviews related to software engineering and information system in refereed journals, books and conferences including *International Journal of Information Management*, *Government Information Quarterly*, *IET-Software*, *Computer Standards & Interfaces*, *Behaviour and Information Technology*, *Public Personnel Management*, *European Journal of Engineering Education*, *International Journal of Information Technology and Management*. He is also in editorial boards of *Journal of Universal Computer Science*, *Computer Standards & Interfaces*, *Computer Science and Information Systems*, *SCS International Journal of Information Technology*, *Journal of Software*, *Electronic Government: International Journal* and others. He has awarded excellence in online education by U21Global Singapore. In research he has awarded by the Scientific and Technical Research Council of Turkey (TÜBİTAK) and Atilim University, Board of management for publications in Science/Social science citation index (SCI/SSCI) journals under international scientific publications promotion program in 2004-2008. He had also served as chief examiner computer science of the International Baccalaureate (IB) organization. He is recipient of various scholarships throughout his academic career including national merit scholarship and department of information technology scholarship.

*Received: October 30, 2008; Accepted: May 15, 2009.*