A Survey on Requirements Engineering Process Maturity Assessment and Improvement Model

Anurag Shrivastava
Researcher,
Department of Computer Science & Engineering,
Gautam Buddh Tehnical University, Lucknow, India.

S.P.Tripathi
Associate Professor
Department of Computer Science & Engineering,
Institute of Engineering & Technology, Lucknow, India.

ABSTRACT
A good Requirements Engineering (RE) Process description will provide guidance to the people involved and reduce the probability that activities will be forgotten about or carried out in a perfunctory way. If RE Processes of an organization is over budget and/or does it takes longer than predicted, people involved in RE complain that they do not have enough time or resources to do the job properly, there are complaints about the understandability or the completeness of the requirement documents, system designers complain to rework resulting from requirements errors, there is a very high volume of change request immediately after the system is delivered to the customers, the RE Processes takes a very long time to agree on system changes, resulting from new requirements, then there is almost certainly the scope for the RE Process Improvement In this paper we have critically examined the developments in Requirements Engineering Process Assessments & Improvements focusing on the deficiencies and outcomes of the previous researches .

General Terms
Requirements Engineering Process Maturity Assessment and Improvement

Keywords
Requirements Engineering Processes, Process Assessment and Improvement

1. INTRODUCTION
Requirements are the description of how the system should behave; Requirements are the description of a system property or attributes; Requirements may be constraints on the development process of the system; a Requirement might describe: a user level facility, a very general system property, a specific constraint on the system, a constraint on the development of the system; Requirements should always be statement of what a system should do rather than a statement of how it should do. Requirements therefore invariably contain a mixture of problem information, statement of system behavior and properties, and design and manufacturing constraints. "Requirements Engineering" is a relatively new term, which has been invented to cover all the activities involved in discovering, documenting, and maintaining a set of requirements for a computer based system, the use of term "Engineering" in Requirements Engineering implies that "systematic" and "repeatable techniques" should be used to ensure that system requirements are complete, consistent and relevant. The problems of Requirements Engineering, increases exponentially, with the size of system. A RE Process is a structured set of activities, which are followed to derive, validate and maintain system requirements. A complete Requirements Engineering Process description should include, what activities are carried out, the structuring or scheduling for each activity, the inputs and outputs to/from the activity and the tool used to support Requirements Engineering.

2. LITERATURE SURVEY
Our rationale for concentrating on this early phase of the software process was that problems in this area have a profound effect on system development costs and functionality, and that a very large number of organizations have poorly defined and informed RE processes. Boehm suggested that the errors in system requirements could be 100 times more expensive to fix than errors introduced during system implementation [5]. Lutz showed that 60% of errors in critical systems were the results of requirements errors [20]. Espiti in a survey of European Companies found that more than 60% of them considered RE problems as very significant [13]. Hall carried out a case study of 12 companies, and discovered that, out of a total of 268 development problems cited 50% (128) were requirements problems [16]. According to Verner (2005); Evanco (2005) the state of industry indicates that only about 60% of organizations keep a record on a single repository, highly significant factor in the success. Clearly Significant benefits can accrue from improving the quality of requirements and by implication RE Processes.

The notion of process improvement through increased product quality and productivity is rooted in the work of Deming's, in his work on Statistical Quality Control [10]. Deming's approach was adopted by Humphrey [18] and others at the Software Engineering Institute in the 1980's and they developed it for Software Process Improvement (SPI). The seminal results of Software Process Improvement (SPI) were process maturity and the identification of a process maturity model with the discrete level of process maturity. This was the development of the Software Capability Maturity Model (SW-CMM) described by Paulk [23], the general notion of a process maturity model has been extended at Software Engineering Institute. The development of the Software Engineering

Institute's Capability Maturity Model (SEICMM), described by Paulk et al. (1993, 1995), a comprehensive model that is predicted on a set of software engineering capabilities that should be present as organizations reach different level of process maturity. The CMM plays an important role in the Software Process Improvement (SPI) of organizations worldwide [31]. The process was developed by the Software Engineering Institute at Carnegie Mellon University in 1986. Its goal is to improve, over time, the application of an organization's software technologies. The model provides a guide for organizations to select software process improvement strategies by facilitating the determination of current capabilities and the identification of critical issues. The CMM process is made up of five well-defined levels of sequential development: initial, repeatable, defined, managed, and optimizing [14]. The Software Engineering Institute has associated Key Process Areas (KPA) with each of the maturity levels. The KPA describe those software engineering functions that must be present to satisfy good practice at a particular level. Each KPA is described by goal, commitment, abilities, and activities characteristics. The key practices are policies, procedures, and activities, which must occur before a KPA has been fully instituted. The studies consistently showed significant organizational performance improvements that were directly associated with process maturity. The data did not indicate any differences in the success in using the CMM for organizations of assorted sizes and types. However, there were hints that small companies found pieces of the CMM irrelevant and hard to apply. A related article by Brodman and Johnson [6] discussed a modified version of the CMM that was more suitable for small organizations and small projects. Problems typically reported with the CMM when used by these organizations were: Documentation overload, Unrelated management structure, Inapplicable scope of reviews, High resource requirements, High training costs, Lack of need guidance, Unrelated practices. CMM allows the maturity status of the software development organizations to be assessed, but does not cover the Requirements Engineering processes. The Software Capability Maturity Models (SW-CMM) has criticized as being too descriptive, as missing many important activities and setting incorrect priorities [23]. The Software Process Improvement literature highlights some of Software Engineering Institute's SW-CMM design flaw; one of the fundamental design flaws is the weak link between process improvement goals and customer expectations. SW-CMM is Complex and hard to implement, hard to use by the Small and Medium Enterprise, Ignores people. Despite this failing, there are many compelling reasons in favor of using SW-CMM concepts as a basis for creating, specialized Requirement Capability Maturity Model (R-CMM). The augmentation of SW-CMM is possible, as its framework has been adapted both "inside" and "outside" the field of Software Engineering. There are reportedly 34 CMM developed by different groups [24]. The SW-CMM is a "living model" and is actively supported by the Software Engineering Institute, recognizes the complex needs of the software industry.

ISO 9001 is an international standard for quality assurance in design, development, production, installation, and service [30]. It is broken down into twenty elements. ISO 9001-3 relates to the development, supply, and maintenance of software. Almost 90 percent of the companies that completed ISO 9001 implementation reported improved internal documentation as one of the most important benefits of registration. Other benefits included higher product quality, greater internal quality awareness, and increased competitive advantage. ISO 9001 is similar to the CMM in the following areas: emphasis on process, documented processes, practiced processes, address the "what" and not the "how" [31]. Differences between the two approaches occur in the areas of focus, dimensions, assessment and certification, coverage, supplier's role, and level of detail. Among the challenges encountered during the installation were a lack of guidance, action knowledge, maturity, and quality personnel.

The Personal Software Process (PSP) is a process-based method developed by the SEI for software engineers to use to apply process definition and measurement to their personal tasks [18]. Most important, the PSP shows developers how to manage product quality, meet commitments, and justify their plans with data. In addition, the PSP follows the concepts of the CMM. The key message of the PSP is that developers should use process management concepts to identify the methods most effective for them. A typical PSP course uses ten software development exercises, a structured sequence of defined processes, and five data analysis exercises to demonstrate the process. An article by Silberberg [29] reported on the successful application of the PSP to Ada software development. The paper also provided a brief introduction to the PSP along with a comparison of the PSP with the CMM.

ISO/IEC 15504 was the result of the SPICE (Software Process Improvement and Capability dEtermination) project [31]. It provides a reference model for focused self-assessments and includes a capability scale that is simple to understand. SPICE defines a two-dimensional model used in a process assessment to describe processes and process capability [12]. The first dimension details the processes, an organization should use to supply, develop, operate, evolve, and support software. The second dimension is made up of nine generic attributes used to characterize the capability of a process. These attributes are grouped into six capability levels (0-5).

Team Software Process (TSP) is a defined method for a group of software developers to create quality software in an efficient manner [19]. It provides process scripts, guidelines, tools, and techniques for a team to develop software applications. The process is based on an incremental model that divides effort into "development cycles." Each cycle involves producing software that satisfies a subset of the total software requirements. TSP was an excellent mechanism to emphasize software quality.

The concept has also been taken up and incorporated into other process improvement models such as European BOOTSTRAP MODEL [17], methodology has been developed to assure the conformance with emerging ISO standard for software process assessment and improvement, the objective of BOOTSTRAP methodology was to provide support for evaluation of process capability against, a set of recognized software engineering best practices and to include internationally recognized software engineering standards as source for identification best practices. BOOTSTRAP is a European method for software process assessment and improvement that was developed to speed up the application of software engineering technology in the European software industry [31]. The BOOTSTRAP methodology is based on the CMM. However, it has been extended and adapted to include ISO 9000 guidelines and the European Space Agency software

engineering standard. Unlike the CMM, BOOTSTRAP does not assume strict adherence to a distinct key practice model and allows the use of alternative approaches [31]. This has been a key factor in its success. In addition, BOOTSTRAP has proven suitable for use by all kinds and sizes of software development organizations. The main features of BOOTSTRAP are: Questionnaires for site and project evaluation, Uniform procedure and mandatory assessor qualification/training, Constructive instead of a normative approach, Open questions, immediate feedback and action planning. The BOOTSTRAP method adopted a process model which addresses processes and practices for both the software producing unit and the project. Process areas were divided into organization, methodology, and technology.

Canadian TRILLIUM MODEL [7], was used by Bell Canada to assess the product development and support capability of prospective and existing suppliers of telecommunications or information technology based product, TRILLIUM can also be used as a reference benchmark in an internal capability improvement program. The Trillium model was initially designed for use with embedded software systems and is based on the CMM [8]. Its architecture differs from the CMM in the following ways: Architecture is based on roadmaps instead of key process areas, a product rather than a software perspective, wider coverage of capability impacting issues, Customer focus and a telecommunications orientation. Trillium is comprised of five levels (1-5). These are unstructured, repeatable and project oriented, defined and process oriented, managed and integrated, and fully integrated [31].

The CMMI (CMM Integration) Model [1] integrate different models and that has both staged and continuous version. CMMI is a process improvement approach that provides organizations with the essential elements of effective processes that ultimately improve their performance. CMMI can be used to guide process improvement across a project, a division, or an entire organization. It helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes. The benefits can be expected from using CMMI include the following: Organization's activities are explicitly linked to business objectives, visibility into the organization's activities is increased to ensure that your product or service meets the customer's expectations, Learn from new areas of best practice. CMMI models are collections of best practices that can compare organization's best practices and guide improvement to processes. CMMI for Development, V1.2- Released in August 2006, and this model is designed for development organizations that want to improve their ability to develop products and services. CMMI for Acquisition, V1.2- Released in November 2007, and this model is designed for acquisition organizations that want to improve their ability to acquire products and services. CMMI for Services, V1.2- Released in February 2009, and this model is designed for service provider organizations that want to improve their ability to establish, manage, and deliver services. One of the problems of these frameworks is that, for taking an approach that targets organization-wide process improvement, spanning management and engineering practices. They do not go into details about specific areas; RE is an example of an area that is not covered in detail. CMMI has only two process areas, Requirement Management and Requirement Development, which are dedicated to RE.

Beecham has designed a Requirements Capability Maturity Model (R-CMM) that guides users towards a view of RE that is based on goals and is problem driven [3]. R-CMM describes how the RE process is decomposed and prioritized in accordance with maturity goals set by the Software Engineering Institute's Software Capability Maturity Model (SW-CMM). R-CMM builds on the Software Engineering Institute's framework by identifying and defining recommended RE sub process that meets the maturity goals. This new focus will help practitioners to define their RE processes with a view to setting realistic goals for improvement. R-CMM proposed the solutions to help practitioners with their technical and organizational RE problems. The work of R-CMM differentiates itself from other RE Models such as Sommerville and sawyer's good practice guide [26] as R-CMM aligns with the SW-CMM rather than developing a new maturity structure. R-CMM is a suite of models that takes practitioners from a high level view of the process, through to a detailed guidelines and finally to a process assessment method to guide companies towards satisfying particular company goals. R-CMM empirical research led us to conclude that the SW-CMM in its current form is not helping practitioners to: 1. Identify and define both technical and organizational aspects of the RE processes. 2. Recognize RE process problems. 3. Assess and agree RE improvement priorities. 4. Related RE process problems to RE goals. 5. Related RE process improvement goals to the general CMM guidelines and activities.

The R-CMM is designed to help practitioners strengthen their RE Process by implementing sub processes or best practices in the logical order. The R-CMM has five maturity levels: R-CMM Level 1 - There are no process improvement goals defined at this unstructured level. R-CMM Level-2-repeatable- The technical difficulty for repeatable level companies centered on complex requirements, requirements growth and undefined RE process. R-CMM Level-3- defined- companies have company-wide communication and standardization of requirement processes instituted across all projects. R-CMM Level 4- managed- Companies have sub processes that are measured to control the RE process and assess where improvements are needed. The goal of R-CMM level-5 is to implement an optimizing RE process. R-CMM level-5 companies have improved requirements method/tools that are instituted with in a stable and predictable environment. R-CMM adapts the Goal Question Metric (GQM) paradigm to include a "process" element. GQM shows how the R-CMM supports continuous improvement [10]. Organizations need to define their improvement goals otherwise improvement activities will turn out to be chaotic as the development process itself [28]. The assessment method used in R-CMM is a tried and tested technique where the 'Approach', the 'Deployment' and the 'Results' of each sub process are measured. The process measurement method is adopted from a model developed by Motorola [9]. R-CMM is still in the development and Evaluation Stage. R-CMM has only been evaluated through experts' opinions and its validity in the real world is still questionable. The R-CMM focuses on the RE process defined within the retired Software Engineering Institute's (SEI's) Software Capability Maturity Model (SW_CMM) process improvement framework. To continue its relevance and usefulness, [25] re-define the whole -CMM within the characteristics of the latest Capability Maturity Model for Integration (CMMI) for Development (CMMI-DEV) v1.2. This describes how the

CMMI-DEV characteristics are used to re-define the R-CMM, and rationale for re-building the RE model based on the latest process improvement framework. Also, explains how the redefined R-CMM adapts to the goals and practices set by the CMMI-DEV. R-CMMi uses a staged improvement path, for organization to reach a particular RE maturity level. This Process assessment method is adapted from the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) Version 1.2 developed by the Software Engineering Institute.

RE Maturity Level1: Initial: RE Process is usually ad hoc and chaotic. The common problems found in level 1 organization relates to "vague requirements, lack of traceability, undefined RE process, insufficient RE resource, lack of training and poor levels of skills" [4]. RE Maturity Level2: Managed : RE process is planned and executed in accordance with policy; requirements changes are managed, requirements traceability is maintained; requirements status tracking is established; involve relevant stakeholders; RE process plan is in place; adequate resources are allocated; RE process is monitored, controlled and reviewed; and RE process adherence is evaluated. RE Maturity Level3: Defined: RE Maturity process is established in organizational standard, procedures, and methods and improved over time. A defined RE process clearly states the practices to elicit needs, analyze, negotiate, document, and verify and validate requirements. In general, the R-CMMi adapts the generic and specific practices of the Requirements Development (RD) key process area (in level 3 of the CMMI-DEV) and adds complimentary RE practices from the literature in model. RE Maturity Level4: Quantitatively Managed: The organization and project establish quantitative objectives for product quality and RE process performance and use them in managing the RE process. The RE process performance and requirements management process, particularly, are controlled using quantitative techniques and are quantitatively predictable. The R-CMMi is guided primarily by the R-CMM and they also rely on the CMMI-DEV to specify the RE practices. RE Maturity Level5: Optimizing: Unlike the level 5 of the R-CMM that only focuses at the "requirements defect prevention activities [4], the R-CMMi maturity level 5 also focuses on continually improving the RE process performance through "incremental and innovative" process improvements. Each RE practice consists of sub practices, typical work products and practice elaboration. A RE practice of a RE maturity level in the R-CMMi assessment process is rated either "fully implemented (FI)", or "largely implemented (LI)", or "partially implemented (P)" or "not implemented (NI)" or "not yet (NY)". For an organization to mature at a RE maturity level, all of the defined set of RE practices for the RE level must be rated either FI or LI. The R-CMMi Model is not validated and its validation is in real world is questionable. The R-CMMi model needed a tool support that can help organization performs internal process appraisal, finds its strength and weaknesses, and be presented with the improvement suggestion which could be adapted by the RE practitioners in the organization.

Gorschek have developed, a five level, Requirements Engineering Process Maturity (REPM) Model [15], aimed at small and medium enterprises, have introduced a light weight evaluation method, which use to evaluate industry projects. The REPM Model is inspired mainly by Sommerville's REAIMS project [26][27]. The individual task of which the REPM Model, is comprised, is called actions. Actions are the smallest constituents of the REPM Model and are in turn mapped to one of the three main categories, called the Main Process Area (MPA) is 1. Requirements Elicitation. 2. Requirements Analysis and Negotiations. 3. Requirements Management. Every action resides on a certain REPM maturity level, spanning level-1 to level-5, where level-1 represents a "Rudimentary Requirements Engineering Processes" and level-5, represents a highly mature process. The actions on each level ensure a consistent and coherent RE process for the particular maturity level. Based on the REPM Model a checklist is constructed, which use to guide the structured interviews. This checklist takes each action and formulates it as a question, which can be answered with one of the three answers: 1.Complete. 2. Incomplete. 3. Satisfied-explained- Satisfied-explained denotes an action that is not completed but the organization doing the evaluation deems the action" Not Applicable" to their project. The results of the project evaluation can be presented as four tables, one for each Main Process Area (MPA) and one summarizing all of the results. Risk Assessment seems to be a neglected area and interaction between requirements do not appear to be mapped, which of course cause can sever problems if there are in fact conflicting and/or volatile requirements. In REPM Model, there is no way of telling when the requirement is fulfilled. A quality requirement, which is often missed and required the most, is a subject with much research focus and methods are still needed for quantifying these quality aspects of software systems. The Main Process Area (MPA) of Requirements Management is generally the one needing most improvement. The main purpose of the REPM Model was to give an idea of the problem scope pertaining to RE practices in industry and to test a method for quickly ascertaining the status of RE in companies. REPM Model was constructed to get a fast and cost effective evaluation of a RE processes. In REPM Model the emphasis was put on speed and ease, not exhaustiveness. For researchers, the question seems to be, to find effective and attractive method for risk assessment and for requirement management. If we look at the REPM Model there is a need for further evaluation, refinement and validation.

Sommerville; Sawyer; and Ransom developed a RE Process Maturity Assessment and Improvement Model derived from existing standards [26][27]. RE Process Improvement Model was concerned with the improvement of processes for the safety critical system development. The RE process maturity assessment helps organizations to compare their RE processes to industrial good RE practices and to identify possible process improvement. The RE Processes Maturity Model proposed by Ian Sommerville and Jane Ransome, contain three levels of RE process maturity corresponding to the first three levels in the Capability Maturity Model, 1.Initial 2.Repeatable 3.Defined. The level of process maturity reflects the extent that good RE Practices used and standardized in a company. Ian Sommerville and Jane Ransome identified 66 good practice guidelines, a method of assigning scores to them and an algorithm that determines the RE maturity levels of an organization. These good RE practices fall into three categories, Basic, Intermediate and Advanced. To assess the process maturity, the assessor examines, the RE processes used in a company and decides which practices that the company has adopted. This assessment can not be carried out mechanically but requires judgment to recognize when particular ways of working corresponds to the good RE practice recommendations. To reflect whether or not a practice has been institutionalized, the assessor allocates a weighted score to each practice, according to its use with in the

organization as follows: 1.Never Used (Score=0), 2.Discretionary –Used at the discretion of individuals (Score=1), 3.Normal-Used by many teams but in different ways (Score=2), 4.Standard- Used throughout the company in a standardized way (Score=3). The maturity level is computed by summing these weighted scores for all basic guidelines and for the Intermediate/Advanced guidelines. The threshold level of the model is: Initial Level - The Score of above 54 in basic guidelines. Repeatable Level- Score of above 54 in basic guidelines and below 40 in (Intermediate + Advanced) guidelines. Defined Level- The Score of above 54 in basic guidelines and above 39 in (Intermediate + Advanced) guidelines.

In order to investigate Sommerville's RE Process Maturity Assessment and Improvement Model, Australian Practitioners, conducted an empirical study and found that the measurement processes designed for practice was very confusing and could lead organization to undesirable results [22]. Sommerville's RE Process Maturity Assessment and Improvement Model does not provide the detailed practice based assessment except for the provision of an indication of the RE maturity levels. It is very important for organizations to systematically discover which RE Practices are weak or strong, RE Process Maturity Assessment and Improvement Model does not provide this detailed practice based assessment except for the provision of an indication of RE Maturity Levels. RE Process Maturity Assessment and Improvement Model evaluates RE processes using a single dimension [21]. Research has shown that measurement process with in RE Process Maturity Assessment and Improvement Model is ambiguous and implementation of his RE Process Maturity Assessment and improvement Model leads to confusion.

Mahmood Niazi, Karl Kox, June Verner, proposed a new RE Maturity Measurement Framework (REMMF) based on Sommerville's RE Maturity Assessment and Improvement Model and to provide initial validation of REMMF [22]. REMMF evaluates each key process area activity as a score between one to ten, which is then rolled into an average score for each key process area. Any key process area average score that falls below seven is considered a weakness [11]. In the REMMF each RE practice is assessed into three dimensions: 1.Approach: The organization's commitment and management support for the practices as well as the organization's ability to implement the practices. 2. Deployment: The breadth and consistency of practice implementation across project areas. 3. Results: The breadth and consistency of positive results over time and project areas. Each dimension of the RE practice is assessed into one of the six categories, 1.Poor and Score is zero, 2.Weak and Score is two, 3.Fair and Score is four, 4.Marginally Qualified and Score is six, 5.Qualified and Score is eight, 6.Outstanding and Score is ten. The 66 good practices designed by the Sommerville can be divided into eight RE Process categories: Requirements Documents, Requirements Elicitation, Requirements Validation, Requirements Management, Requirements Analysis and Negotiations, Describing Requirements, System Modeling, Requirements for Critical Systems. These RE process categories contain good practice guidelines designed for RE processes. REMFF includes the following procedure, for each good practice, in order to measure its maturity. Step1: For each practice, a key participant who is involved in the RE process assessment calculates a three-dimensional score. Step2: The three-dimensional scores for each practice are added together, divided by three and round up. A score for each practice is placed in the evaluation sheet. Step3: This procedure is repeated for each practice. The score for each practice is then summed and then an average is used to gain an overall score for each RE Process category. Step4: A score of seven or higher for each 'RE Process Category' indicates that specific category maturity has been successfully achieved. A RE Process Category maturity score that falls below seven is considered a weakness.Step5: It is possible that some practices may not be appropriate for an organization and need never implemented. For such practices a 'Not Applicable' (NA) option is selected. This Not Applicable practice should be ignored when calculating the average Score of a 'RE Process Category'. REMMF is still needed a tool support to perform calculations and to generate different assessment reports. REMMF itself is not fit into the RE Maturity Assessment in the small and medium scale software companies, due to cumbersome , lengthy and time consuming procedure for assessment of RE Processes. The problem has two main causes. First, smaller enterprises often lack of specialists with significant experience in tailoring state of the art software development to the organization's need. Second, they typically lack the time and money for improvement activities because they invest primarily in customer satisfaction. Recently, Sami Jantunen [20] presented the results of an exploration of software engineering practices in five small and medium-sized organizations. Despite the research work not focusing on RE practices, the study reveals interesting issues about software development practices in small organizations. We believe collaboration issues are important to understand RE practices as RE is a Communication-intensive activity. Although the work. Georgi [23] presented a RE improvement project conducted in a Siemens business unit, describes the situation at the business unit before the process improvement project, gives a short overview on how the project was implemented and the techniques applied to solve the various problems the organization facing.

3. DEFICIENCIES AND OUTCOMES

The RE process maturity assessment and improvement models we have discussed so far are not feasible in small and medium scale software companies in india, the cost and time of introduction of RE practices proposed by the discussed RE process maturity assessment and improvement models are very high, these models do not consider the real needs of small and medium scale software companies in india, realizing that such small and medium scale enterprises are often know their domain and context well. RE process assessment and improvement models should integrate employees into the improvement process; enabling them to make improvement decisions rather than having external consultant make the decisions for them, a clear and unambiguous Maturity Measurement Framework Model, covers all the needs of small and medium scale software companies. Medium and Small scale software companies are so varied that we can't make general assumption about their contexts however we are presenting the factors that should be in a Maturity Measurement Framework for small and medium scale companies.

1. Ease and Inexpensiveness: Improvement methods should allow fast, inexpensive assessments. Small and medium scale software companies should be able to identify practices that they can introduce in one step with limited effort and money. Instead

of a few big iterations with unpredictable consequences, they should be able to run many in small iterations that focus on one problem.

2. Understandability and Predictability: All stakeholders should be able to easily understand the problems and the rationale behind chosen improvements. Because smaller enterprises are at a high risk for inappropriate changes, the benefits and challenges must be predictable as possible.

3. Flexibility: Improvement methods should provide suggestions that are flexible and easy to adapt to the organization's development Context, philosophy, and Requirements Engineering Processes.

4. Enabling Participation: Improvement methods should enable all relevant stakeholders to participate. The methods should motivate stakeholders to make changes and should profit from stakeholders knowledge about the context.

5. Concreteness: Improvement methods should provide concrete improvements, including information on implementing changes and performing suggested activities. Otherwise, organizations with limited knowledge of state-of-art methods will have trouble establishing improvements.

4. CONCLUSION

The state of Requirement Engineering practices in small and medium scale software companies has still been largely unexplored. We have found that the Requirements Engineering process maturity assessment model needs improvement in the area of small and medium scale software companies, in India. The above survey leads to a final conclusion that the practice-based improvement using self-assessment is appropriate for smaller companies; and they should focus on tailoring the context dependent practices.

5. REFERENCES

[1] Ahern, D. M., Clouse, A., and Turner, R. 2001. CMMI Distilled. Addison-Wesley, Reading, Mass.

[2] Basili, V.R.1995. Measurement Frameworks, Software Quality Assurance and Measurement: A world class perspective, eds. N.Fenton,R.

[3] Beecham S., Hall, T., AND Rainer A., "Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis," Empirical Software Engineering, Kluwer Academic Publishers, The Netherlands, Vol. 8. No.1, pp.7-42, 2003.

[4] Beecham S., Hall T., and Rainer A., "Building a Requirements Process Improvement Model," Technical Report No. 378. University of Hertfordshire: Hatfield, 2003.

[5] Boehm, B. W. 1983. The economics of software maintenance. In Proceedings of Software MaintenanceWorkshop (Washington, D.C.), 9–37.

[6] Brodman, J., & Johnson, D. (1997). A software process improvement approach for small organizations and small projects. Paper presented at the International Conference on Software Engineering, Boston, MA.

[7] Coallier, F. 1999. Trillium: A model for the assessment of telecom product development and support capability. In Software Process Improvement, R. B. Hunter and R. H. Thayer, Eds. IEEE Computer Society Press, Los Alamitos, Calif.

[8] Coallier, F., Mayrand, J., & Lague, B. (1999). Risk Management in Software Product Procurement. In K. Emam & N. Madhavji (Eds.), Elements of Software Process Assessment and Improvement (pp. 23-44). Washington, DC: IEEE Computer Society Press.

[9] Daskalantonakis, M.K. 1994,Achieving higher SEI levels, IEEE software 11(4).

[10] Deming, W. E. 1982. Out of the Crisis. MIT Press International, Cambridge, Mass.

[11] Diaz, M., & Siigo, J. (1997). How software process improvement helped Motorola. IEEE Software, 14(5),75–81.

[12] Drouin, J. (1999). The SPICE Project. In K. Emam & N. Madhavji (Eds.), Elements of Software Process Assessment and Improvement (pp. 45-55). Washington, DC: IEEE Computer Society Press.

[13] Espiti. 1996. Software process improvement on the right road with ESPITI—The ESPITI European Survey Results. ESPITI Newsletter Issue 2. Available at: http://www.cse.dcu.ie/cse/international/trispin/News2.html #espiti.

[14] Freedman, A. (2000). Computer Desktop Encyclopedia (Vol. 13.4). Point Pleasant, PA: The Computer Language Company.

[15] Gorscheck, T., Svahnberg, M., & Kaarina, T. (2003). Introduction and application of a lightweight requirements engineering process evaluation method. In Proceedings of the Requirements Engineering Foundations for Software Quality (REFSQ'03) (pp. 83–92). Klagenfurt/Velden, Austria.

[16] Hall, T., Beecham, S. & Rainer, A. (2002). Requirements problems in twelve software companies: An empirical analysis. IEE Proceedings—Software, 149(5), 153–160.

[17] Hasses, V., Messnarz, R., Koch, G., Kugler, H. J., and Decrinis, P. 1994. Bootstrap: Fine tuning process assessment. IEEE Software 11, 4, 25–35.

[18] Humphrey, W. 1989. Managing the Software Process. Addison-Wesley, Reading, Mass.

[19] Hilburn, T. (2000). Teams need a process! Paper presented at the Annual Joint Conference Integrating Technology into Computer Science Education, Helsinki, Finland.

[20] Jantunen, S., "Exploring software engineering practices in small and medium-sized organizations", Proceedings of the 2010 ICSE Workshop on Cooperative and Human aspects of Software engineering (CHASE'10), pp.99-101,2010.

[21] Lutz, R. R. 1993. Analysing software requirements errors in safety-critical embedded systems. In Proceedings of RE'93 (San Diego Calif.).

[22] Marjo, K., & Sari, K. (2001). Assessing Requirements Engineering Processes with the REAIMS model: Lessons

learned. In proceedings of the Eleventh Annual International Symposium of the International Council on Systems Engineering (INCOSE2001).

[23] Markov, G.A., Hoffmann, A., & Creighton, O., "Requirements Engineering Process Improvement : an Industrial case study", In REFSQ'11 Proceesings of the 17th international working conference on Requirements Engineering: Foundation for software quality, Springer-Verlag Berlin, Heidelberg,2011.

[24] Niazi, M., Cox, K., & Verner, J. (2005a). An empirical study identifying high perceived value requirements engineering practices. In Fourteenth International Conference on Information Systems Development (ISD'2005). Karlstad University, Sweden August 15–17.

[25] Paulk M.C., and Chrissis M.B., The 2001 High Maturity Workshop (CMU/SEI-2001-SR-014), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 2002.

[26] Reifer, D.J. 2000. The CMMI: It's formidable, Journal of System and Software 50:97-98.

[27] Solemon B., Sahibuddin S.Ghani A.A.A., 2009, "Re-defining the Requirements Engineering Process Model", 2009. In proceeding, 16th IEEE Asia-Pacific Software Engineering Conference.

[28] Sommerville, I. and Sawyer, P. 1997. Requirements Engineering: A Good Practice Guide. Wiley, Chichester.

[29] Sommerville, I., & Ransom, J. (2005). An empirical study of industrial requirements engineering process assessment and improvement. ACM Transactions on Software Engineering and Methodology, 14(1),85–117.

[30] Solingen, R.V. and Berghout, E. 1999. The Goal/Question/Metric Method: A Practice Guide for Quality Improvement of Software Development, Maidenhead, UK, McGra-Hill Publishing Company.

[31] Silberberg, D. (1998). Applying the personal software process (PSP) with Ada. Paper presented at the Annual International Conference on Ada.

[32] Weissfelner, S. (1999). ISO 9001 for Software Organizations. In K. Emam & N.

[33] Zahran. (1998). Software Process Improvement: Practical Guidelines for Business Success. Reading, MA: Addison-Wesley.