Story card Maturity Model (SMM): A Process Improvement Framework for Agile Requirements Engineering Practices

Chetankumar Patel
Innovation North, Faculty of Information and Technology
Leeds Metropolitan University
Email: c.patel@leedsmet.ac.uk

Muthu Ramachandran
Innovation North, Faculty of Information and Technology
Leeds Metropolitan University

Abstract— **This paper describes an ongoing process to define a suitable process improvement model for story cards based requirement engineering process and practices at agile software development environments. Key features of the SMM (Story card Maturity Model) process are: solves the problems related to the story cards like requirements conflicts, missing requirements, ambiguous requirements, define standard structure of story cards, to address non-functional requirements from exploration phase, and the use of a simplified and tailored assessment method for story cards based requirements engineering practices based on the CMM, which is poorly addressed at CMM. CMM does not cover how the quality of the requirements engineering process should be secured or what activities should be present for the requirements engineering process to achieve a certain maturity level. It is difficult to know what is not addressed or what could be done to improve the process. We also presents how can be the identified areas of improvement from assessment can be mapped with best knowledge based story cards practices for agile software development environments.**

Index Terms— **Story card maturity model, Agile requirements, Requirements Engineering, SPI**

I. INTRODUCTION

Requirements elicitation process is one of the challenging processes in the software development methods. In traditional software development methods end users or stakeholders predefined their requirements and sent to the development team to do analysis and negotiation to produce requirement specification. Traditional software development has a problem to deal with requirement change after careful analysis and negotiation. This problem is well tackled by the XP, which is one of the agile software development methodologies.

Extreme (XP) programming is a conceptual framework of practices and principles to develop software faster, incrementally and to produce satisfied customer. It is a set of twelve practices and four principles, which makes XP successful and well known among all the agile software development methods. The goal of XP is to produce the software faster, incrementally and to produce satisfied customer (Beck, 2000). According to Bohem (1998) the cost of change grows exponentially as the project progresses through it lifecycle (Bohem 1981). The relative repair cost is 200 times greater in the maintenance phase than if it is caught in the requirement phase (Faluk, 1996). XP maintain the cost of change through iterative software development methods and Refactoring.

While CMM and CMMI or software process improvement has gained a lot of attention during the last decade. Due to the increasing competition in the software market faster delivery, high quality products and customer satisfaction are the major concerns for software organisations. A quality process can have a positive impact on services, cost, on-time delivery, development technology, quality people and quality of products (Zahran, 1998).

Getting requirements on story cards right continues to be a universal problems same as the requirements problems in the traditional methodology. Story cards errors can be costly in terms of low time, lost revenue, loss of reputation and even survival (Beecham, *et al*., 2005). A critical aspect of the requirements process is the selection of the an appropriate requirements set from the multitude of competing and conflicting expectation elicited from the various project stakeholders or from an onsite customers (Wiegers, 1997).

Looking at methods of CMM for process quality, measurement and improvement they tend to cover the area of requirements engineering poorly. It covers the area of requirements engineering inadequately. CMM does not cover how the quality of the requirements engineering process should be secured or what activities should be present for the requirements engineering process to achieve a certain maturity level. Some time it is really difficult to assess the maturity of a requirements engineering process for a certain projects, and it is

difficult to know what is not addressed or what could be done to improve the process.

As agile software development methodology is the iterative software development methodology based on the story cards, for small to medium organisation and main objectives are lower cost, high productivity and customer. The CMM tends not to focus the software process on an organisation's business objectives in their software process improvement programme (Paulk, 1998). The main thing is that CMM and ISO 9000 do not say much about requirements engineering and subsequently little about how the quality of the requirements engineering process should be maintained and ensured (Gorschek and Tejle, 2002). Herbsleb and Goldenson (1996) reported the use of the CMM in several software organisations. The study consistently showed significant organisational performance improvements that were directly associated with process maturity. The study also mentioned that the CMM improvement path is not always smooth, the efforts generally took longer and cost more than expected. While story card is agile software developments practice. Agile software development methodology is targeted to lower cost. Some of the KPAs have been found difficult to apply in small projects (Brodman and Johnson, 1997). This may be because CMM was originally structured for big enterprises (Lyard and Orci, 2000). CMM addresses practices such as document policies and procedure that large organisations need because of their size and management structure ((Brodman and Johnson, 1997).

Normally story cards for agile software development do not support the heavy documentation at all and people communicate verbally on on-going basis. Unlike CMM, CMMI does not just focus on software process management; it also considers other department such as marketing, finance and purchasing (Ahern, *et al.*, 2003). So it could be seen unnecessarily complex, when it is applied to agile software development practices like Extreme programming, Scrum and lean development.

CMM can be described a common sense application of the process management and quality improvement concept to software development and maintenance but it focuses on software development and does not cover the requirements engineering process (Somerville, *et al*., 2000; Kotoyana, *et al*., 1998). Without the standard for ensuring the quality of requirements engineering process, it is hard to ensure the result of the requirements engineering process. A consequence of this can be that requirements do not reflects the real needs of the customer of the system, requirements are inconsistent, incomplete requirements and requirements are not specified in a standardised and adequate manner (Gorschek and Tejle, 2002).

When businesses adapt the CMMI they should be familiar with the CMM practices (Menezes, 2002). CMMI Based upon the software CMM and has most of the same process areas. It may also inherit some of the same problems as CMM, such as the problem in reaching higher capability levels (Boehm, 2003). This is not acceptable against the agile software developments principle and motivation.

There is a need for a requirements process improvement model to suit story cards based requirements engineering process. Therefore the purpose of this paper is to propose and evaluate a requirements process improvement model for story cards based requirements engineering process and enhance the adaptability of story cards based requirements engineering process. We propose a model for assessing the story cards based requirements engineering process within software engineering projects. This model should cover the area of story cards based requirements engineering process and practices for agile software development. The model can be used to evaluate the story cards based requirements engineering process maturity for certain projects.

The Story card Maturity Model (SMM), requirements improvement framework offers many advantages. The SMM includes an assessment method that guides the user to understand current story cards based requirements engineering process. The rationale for building the SMM is as:

- To define a generic process model for Story cards based requirements engineering process improvement that is suitable for RE at agile software development environments.
- To design and implement an automated tool that support to apply the proposed model in order to help facilitate process improvement.
- Identify and define story cards based requirements engineering practices
- Recognise story cards based requirement engineering practices problems
- Access and agree story cards based RE practices improvement priorities
- Relate story cards based RE practices problem to RE practices improvement goals
- Contains guidelines for many story cards related requirements engineering activities
- Is designed to be tailored to focus on specific process areas
- Goal focused
- Maturity structure to help with process prioritisation.

This paper propose an approach for simplified and tailored assessment method for story cards based requirements engineering process using CMM(I), and present how assessments can be performed on story cards based RE process. In this paper we explain the main stage involved in developing a model that guides practitioners to under stand their story cards based requirements engineering process. We aim to support the practitioners by providing guidelines for story cards process

improvement within the familiar framework. This study focuses on the story cards based requirement engineering process and not the individual feature or behaviour of the system. In this section we discussed about the research challenges, CMM, CMMI and agile software development. This section is followed by the process improvement framework for story cards.

A. Capability Maturity Model (CMM)

Capability maturity model was produced by the software engineering institute at Carnegie Melon University during the 1980s. Many companies throughout the world use the software CMM as their software process improvement model. Result of using this method is generally positive with improved process leading to higher quality software (Beecham, S. et al., 2003). According to Humphrey (Humphrey, W. S. 1993)

"When faced with a problem software people generally find their own solutions, even when the problem has been solved many times before. The fact that it is so hard to build on other people's work is the single most important reason why software has made so little progress in the last 50 years".

The CMM is a conceptual framework based on industry best practices to assess the process maturity, capability, and performance of a software development organisation; it covers practices for planning, engineering, and managing software development and maintenance (Herbsleb and Goldenson 1996). CMM has been widely accepted as a reference model for process assessment and improvement. It has become the de facto standard for software process assessment and improvement (Persse, 2001, Paulk et al., 1995, Raynus, 1999, Li, Chen and Lee 2002). CMM consists of five levels of maturity as Initial, Repeatable, Defined, Managed and optimizing (Paulk, C. et al., 1993). The following figure 1 show and summarize the five capability levels (Paulk, C. et al., 1993, Paulk, C. et al., 1995).

- *Initial:* At this level the software process is characterised as ad hoc, and occasionally even chaotic. At this level few processes are defined and success depends on individual efforts and heroics.
- *Repeatable:* The basic project management processes or plans are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
- *Defined:* The software process for both management and engineering activities is documented, standardised, and integrated into a standard software process for the organisation. All the projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
- *Managed:* Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
- *Optimizing:* continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

As can be seen in the figure 1 each maturity level comprise of Key Process Area (KPA). The goals of KPA summerise the states that must exist for that key process area to be implemented in an effective and permanent way. The extent to which the goals have been accomplished is an indicator of how much capability the organisation has established at the maturity level. The Goals signify the scope, boundaries, and intent of each KPA (Paulk et al., 1995).

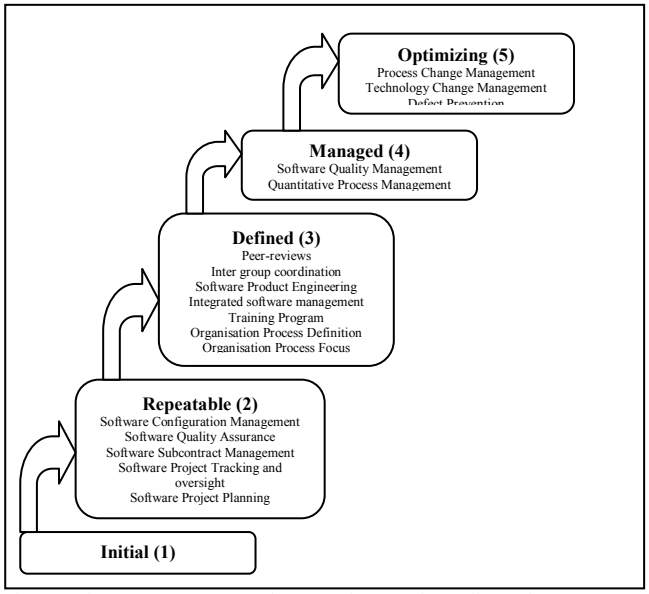


Figure 1 the Key Process Area by Maturity Level (Paulk, et al, 1995)

The Key process areas at maturity level one focus on Heroes and massive efforts, at level two focus on project management and commitment processes next maturity level, defined level focus on defined engineering process while level four and five, managed and optimizing focus on product and process quality, and continuous process improvement respectively(Zaharan 1998). The process rating maturity level rating of an organisation is measured as, the extent to which KPAs are implemented and institutionalised at each level. The extent of implementation for specific KPA is evaluated by assessing the common features as (Paulk et al., 1995).

- Commitment to perform policies and leadership
- Ability to perform
- Activities performed
- Measurement and Analysis
- Verification of implementation

B. CMMI (Capability Maturity Model Integration)

CMMI was developed by the SEI recently. The CMM is the original version of the CMMI, and as CMM was originally funded by the US Department of Defence (DoD) to help qualify DoD's Software vendors' capabilities (Chrissis, Konrad and Shrum 2003) CMMI integrates all CMM versions mainly to reduce implementation cost (Ahern, Clouse and Turner, 2003) by

- Eliminating Inconsistency
- Reducing duplication
- Maintaining common component rules
- Increasing clarity and understanding
- Providing common terminology
- Assuring consistency with ISO/IEC 15504

The CMMI model is for improving and assessing the performance of development organisations (Chrissis et al., 2003). CMMI is a powerful tool to guide process improvement initiatives, not only for software development but for many related field such as System engineering, product acquisition, and team management (Boehm et al., 2002). CMMI is a powerful tool to guide process improvement initiatives, not only for software development but for many related fields such as system engineering, product acquisition, team management (Boehm et al., 2002). It has been shown to reduce the risks associated with development projects, increase efficiency and improve the overall quality of products and deliverables (Ahern, Clouse and Turner, 2003). The CMMI has two separate model representations staged and continuous (Ahern, clouse and Turner 2003)

As can be seen in the figure 2 illiterates the five levels of CMMI which are similar to the CMM models, and Process areas are illustrated in table 1 and 2.

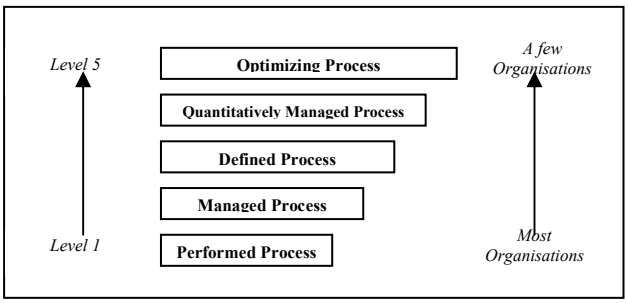


Figure 2 CMMI Staged Approach (Ahern et al., 2003)

Maturity Level	Process Area
Maturity Level 1	No Process Area Associated with the maturity level 1
Maturity Level 2	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
Maturity	Requirements Development
Level 3	Technical Solution Product Integration Verification Validation Organisation Process Focus Organisation Process Definition Organisation Training Integrated Project Management Risk Management Integrated Teaming Integrated supplier Management Decision analysis and resolution Organisational Environmental for integration
Maturity Level 4	Organisational Process Performance Quantitative Project Management
Maturity Level 5	Organisational Innovation and deployment Casual Analysis and Resolution

Table 1 the Process Areas for Each Maturity Level (Ahern et al., 2003)

Category	Process Area
Project Management	Project Planning Project Monitoring and control Supplier agreement management Integrated project management Integrated training Risk management Quantitative project management
Process Management	Organisational process focus Organisational process definition Organisational training Organisational process performance Organisational innovation and deployment
Support	Configuration management Process and product quality assurance Measurement and analysis Casual analysis and resolution Decision analysis and resolution Organisational environment for integration
Engineering	Requirements management Requirements management Technical solution Product integration Verification Validation

Table 2 the process areas in the continuous representation (Ahern et al., 2003)

C. Agile Software Development (ASD) Methodology

Agile software development methodology is a framework to develop the software. In the late 1990's several methodologies began to get increasing public attention. Each had a different combination of old ideas, new ideas, and transmuted old ideas. But they all emphasized close collaboration between the programmer team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organizing teams; and ways to craft the code and the team such that the inevitable requirements churn was not a crisis (Agile alliance). According to Ambler (2005), an author of Agile modelling 'Agile is an iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative

manner by self-organizing teams with "just enough" ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.' This methodology gives priority to incremental software development methods, which is called iteration in agile software development method. Pay attention on face to face communication over documents. A recent set of development techniques that apply a human-centred approach aims to deliver high-quality products faster, and to satisfy customer (Ceschi 2005) and more people oriented rather than process oriented.

D. The Agile Manifesto

In the Agile Manifesto (Agile Alliance, 2001) the members of the alliance say their aim is to develop better software (in other words productive software) while helping other to do likewise. The manifesto then goes on to least the four values of agility as follows (Agile Alliance, 2001):"

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more" The agile values are implemented or represented through a set of the following agile software principles (Agile Alliance, 2001):"

- Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months. With a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most effective and efficient method of conveying information to and within a development team is face-to-face communication
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
- Continuous attention to technical excellence and good design enhances agility
- Simplicity the art of maximizing the amount of work not done is essential.
- The best architectures, requirements, and designs emerge from self organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

There are several agile software development methods as:

- Extreme Programming (XP) (Kent Beck 1999)
- Scrum (Jim HighSmith)
- Adaptive Software Development (Jim Highsmith)
- Future Driven Development
- The Rational unified Process
- Crystal Family of Methodology (Alistair Cockburn)
- Open Source Software Development

Among these all agile software development methodology, Extreme Programming (XP) is one of the most popular agile methods. This is lightweight Agile Software Development Method (Beck K. 2000) and address to where customer requirements are vague and change over time. (Beck K. 2000). This method is suitable for small to medium enterprise.

II Process Improvement Framework for Story Cards

A. Story cards based Requirements engineering Maturity Model (SMM)

According to Christie (1999), defining processes is recognised as critical elements in the software process improvement (Christie, 1999) yet to be useful model must be clear simplification of the complex world it is modelling (David, 2000). To keep the representation clear, understandable and usable the SMM links the Story Cards based requirements engineering practices to maturity levels, but it is not an exhaustive representation of agile software development practices. The SMM model is based on the agile requirements engineering values, practices and principles.

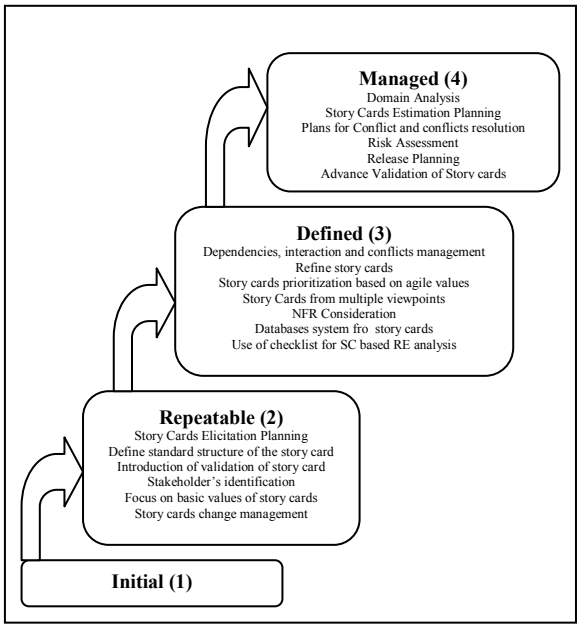
The SMM model is designed to improve and enhance the agile software development methodology and boost up the agile requirements principles and objectives like the lower cost, customer satisfaction, requirements quality, etc. Figure 3 introduces the SMM (Story card Maturity Model for agile software development requirements engineering). We divided our SMM model into four maturity level compared to the CMM five level maturity model This high level view of the model shows how story cards based requirements engineering practices mature from an initial or ad-hoc level to continuously

improving level based on the agile principles and practices. In this model each level has a pre defined goal to help practitioner or organisation focus on their improvement activities. The ultimate goal of the SMM (Story cards maturity model) is as:

- Customer satisfaction
- Maintain story cards (requirements) change
- Solution to Vague requirements
- Obtain an understanding of the user story on the story card
- Obtain commitment to the user story (user Requirements)
- Maintain bidirectional traceability of requirements
- Identify inconsistency between project work and user story
- Identify and involve stakeholders instead of single on-site customer
- Manage Requirements Stories, on-site customer, daily meeting,
- Establish Estimates of story cards and define acceptance tests with story cards
- Develop a project plan based on the story cards

1) Level 1: Initial Level (Not accommodating at all)

There is no process improvement goals defined at this unstructured level.



The story cards practices or story card based requirement engineering process is very slim at this level and not necessarily repeatable. Organisations typically do not provide a stable environment for story cards based requirements engineering practices. Level 1 company do not have defined story cards practices. Here at this level RE problems were found to be common. The main problems at this level relate to overtimes, schedule slips, communication, requirements quality and vague requirements. These companies operate in their own unique way and depend on particular people rather than whole team. Paulk et al (1995) describe for traditional software process, success at this level is depends on 'the competence and heroics of the people in the organisation and cannot be repeated unless the same individuals are assigned to the next project'.

2) Level 2: Explored

Level 2 denotes a more structured and complete software development practices than level 1. Organisation with level 2 capability experienced fewer problems with their software development process than their level 1 counterparts.

Problems with communication, complex requirements management and undefined RE process along with staff retention. Technically difficulty for level 2 companies centred on communication (mutual interaction), and to handle complex requirements.

An organisation at this level has focused on the cost, schedule and functionality and story cards elicitation process, the story cards elicitation practice is used to elicit user goals, elicit the functional requirements, An organisation at this level has introduced policies that ensure that story cards (Requirements) are specified and used the standard structure of the story cards and story cards are written by the on-site customer. Level 2 in general denotes that an organisation has devoted resources to the identification of story cards (requirements engineering) practices as a whole.

In general companies at this level 2 process capability have established the scope of the story cards, story cards based requirements elicitation and identification of stakeholders to track schedules, requirements (Story cards), cost and functionality.

The SMM at level 2 maturity aims to help developers and customers to identify and improve problems related to requirements elicitation and identification of stakeholders by learning from previous project success and failures. This is achieved by an assessment of current process and to identify where weakness lie will help development team gain a general overview and allow them to address any planning or requirements issues associated with individual projects. The Appendix 1 summaries goals, key process areas and assessment questionnaires for SMM maturity level 2.

3) Level 3: Defined Level

Level 3 denotes a more focus on practices related to customer relation ship management, consideration of dependencies, interaction, conflicts between story cards, acceptance testing on early stage of story cards, prioritization of story cards based on the agile values for iteration planning and stakeholders are consulted to improve the quality of the story cards.

The customer relationship is maintained very well at this level. At this level companies ensure a deeper understanding of acceptance testing for the requirements testing, and subsequently stakeholders are consulted to elicit the requirement from the multiple viewpoints.

At this level companies stored the story cards on the database or the computer system for story cards reuse compared to use and throw concept of the traditional story cards. Story cards analysis can be done through the checklist based on the story cards and agile requirements value and principles.

Level 3 companies had increased their control over their technical practices like requirements testing practices and furthermore the practices related to dependencies of story cards and requirements conflicts are focused, but saw little improvement on the support of the RE process. They improved user understanding of story cards or requirements, internal and external communication but continued to report problem on analysis of domain where the system is going to be implemented and estimation of story cards. At this level no structured risk assessment is performed. Furthermore no consideration is taken towards non-functional requirements.

The SMM at level 3 maturity aims to help developers identify and improve problems related to customer relationship; story cards early testing (Acceptance Testing), dependencies and conflicts management of story cards. This is achieved by an assessment of current process and to identify where weakness lie. The goals, key process areas and assessment questionnaires for SMM maturity level 3 is summarised in the Appendix 1

.

4) Level 4: Improved (People orientation and project management Practices)

Companies at this maturity level are in a position to collect detailed measure of the story cards based requirements engineering process or practices and product quality, both the story cards based requirements engineering practices and products are quantitatively understood and controlled using detailed measurements (Paulk et al. 1995)

At this level the system's environment is studied in grater detail, not only the technical aspect but also the demands coming from the application domain, as well as the business process where the system should support. The improved level of the SMM model is also focused on the estimation of story cards, risk assessment, release planning and testing for non functional requirements. At this level the development team also focus on the advance validation of the story cards and identifies the unit tests from the story cards for the development stage. This is an internal attribute of the team which is not directly visible to the customer. Level 4 denotes a more active and mandatory examination of risk.

The SMM at level 4 maturity aims to help developers or managers to identify the non-functional requirements to improve the quality and to improve the estimation and release planning. This is achieved by an assessment of current process and to identify where weakness lie. The Appendix 1 summaries goals, key process areas and assessment questionnaires for SMM maturity level 4.

III Software Process Improvement Roadmap for SMM

The SMM model is summarised in figure 4. The key features of the process are as:

- Adaptability and suitability assessment is carried out by the agile team members which are any like developers, coach, testers with collaboration of on-site customer. This is found to be a useful process during the SMM implementation. The purpose of involving this process is to ensure or to identify that the organisation follows the story cards based requirements engineering practices and process or not. If not then this adaptability and suitability recommends what they needs to do to follows the story cards based agile software development.

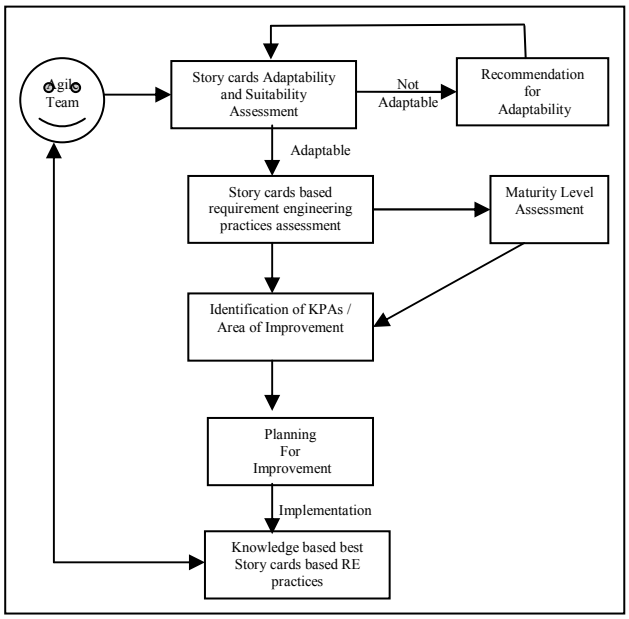


Figure 4 The SMM Process

- Early in the SMM programme the business objectives or business goal are defined by the agile team. The goals drive much of the subsequent activity, especially the selection of KPAs or maturity level and prioritisation of the area for improvements.
- A tailored version of the SMM assessment (similar like CMMI model but the key process areas and goals are entirely different than CMMI) is carried out by the agile team, to identify area for improvement. This is also indicating the maturity level of the software process.

- The plan for the improvement is identified based on the inputs provided to the assessment questionnaires for each maturity level key process areas. In this plan, practices should be identified to support the implementation of the prioritised area for improvements.
- After the identification of the KPAs for each maturity level, a guide based approach was designed to capture the best practices in order to improve the prioritised area for improvement.

IV The Adaptability and suitability assessment

Adaptability framework is based on the questionnaires, like the determining the main problems in the existing requirements engineering process or requirements engineering practices used or intend to use during the next project, existing knowledge on traditional requirements engineering practices and story cards based requirements engineering practices and process, customer relationship with development team, customer availability during the project, developers attitude or characteristic towards the process and by assessing their knowledge on agile requirements process.

An adaptability questionnaire is actually divided in the following four sections.

- Requirements engineering process used or intends to use.
- Problem identification during the story cards based requirements engineering process and Solution adopted or trying to adopt to solve problems
- Customer availability and relationship
- Developers and Managers knowledge on Agile requirements engineering

Our adaptability assessment brings three result based on the answers supplied on the adaptability Model. Those results are as following

1. Recommended to adopt story cards based requirements engineering methodology on you pilot project.
2. Ready to adopt a story cards based requirements engineering methodology but needs an improvement or needs to pay attention or focus on the recommended area.
3. Pilot project is not suitable for story cards based requirement engineering methodology, but they can still apply agility after adopting agile software development knowledge

The story cards based requirements engineering process adaptability assessment requires an extensive knowledge of story cards practices and requirements engineering process. This adaptability just not cover one aspect of the requirements engineering based on the story cards, it covers all aspect of the story cards based requirements engineering process and it puts people in the centre of the assessment instead of process itself.

V Story Cards Based Requirements Engineering Process Assessment method and identification of KPAs for improvement

The purpose of the assessment method is to assess the current story cards based requirements engineering process. Process assessment consists of the knowledge on story cards based requirements practices and business case workshop, which focus on process improvement and provides a roadmap for process improvement. The SMM assessment model is based on a story cards based requirements engineering practices and process. It is a modified and customisable version of the SW-CMM assessment questionnaires. Emphasis placed on story cards based requirements practices, developers and on-site customers. This process is expected to enhance the communication and understanding; in particular it is expected to clarify the actual issues of the people involved in the process improvement actions. SMM recommended having a shared vision of the process improvement and any one can control process improvement activities at any stage. The following figure 5 shows to identify the areas for process improvement.

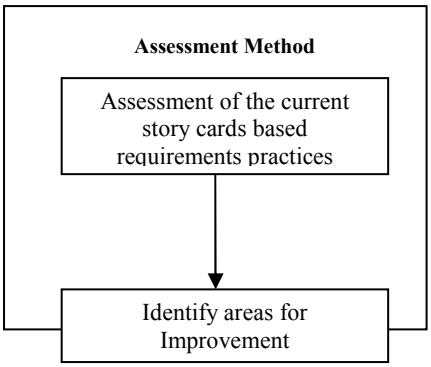


Figure 5 Areas for improvement assessment framework

SW-CMM provides a guideline for good management and engineering practices, with a strong emphasis on management, communication, and co-ordination for development and maintenance of the software process. But as can be seen earlier SW-CMM does not suitable or acceptable for the agile software development practices and requirements engineering practices are poorly addressed.

The SMM model's main objective is to tailor the story cards based requirements practices and process for the agile environments; therefore identifying the maturity level of the agile requirements practices is a crucial activity in the SMM model.

The main objective from the SW-CMM assessment is to assess the capability of an organisation and identify the key process area as opportunities for process improvement. The main objective of the SMM assessment is to identify the areas for improvement. This

approach is achieved by the SMM trough its own assessment questionnaires based on the story cards based requirements engineering practices, principles and values. See Appendix 1 for SMM assessment questionnaires. In SMM the KPA identifies the issues that must be addressed to achieve a maturity level in SMM maturity model. Each KPA identifies the cluster of goals considered important for enhancing process capability. These related activities are called the key practices. An automated tool has been built to facilitate the work of the SMM method.

The SMM assessment in this project is tailored to suit story cards based agile software development environments, their needs and objectives such as eliminating the practices which are not necessary for them and adding new practices which directly related to story cards based requirements engineering practices. Thus the SMM assessment method is flexible and does not involves any unnecessary KPAs or questionnaires.

Self assessment is the most common way of performing software process assessment (Dutta et al., 1999). The popularity for self assessment lies in its low cost, good accessibility and ownership of the result (Dutta et al., 1999). We are going to follow the self assessment for the story cards based requirements engineering process assessment. Automated assessment also considered for this approach.

SMM assessment questionnaires responses are: Yes, Partially, No, Not Applicable (N/A). This assessment response are very similar to SW-CMM response Yes, No, N/A and Don't Know. In our approach response partially permits the assumption that part of the process or work may have been performed or if performed then not fully addressed. N/A is selected when the practice is not possible to implement. If the answer is Yes than the practice is fully implemented and well addressed in the project. If No then it's not addressed at all.

In SMM assessment area of improvement is identified if the answer of the questionnaires is as Partially, No or N/A. Using these criteria the percentage for each KPAs can be calculated as follows:

$$\frac{\sum (Y_n) + \frac{1}{2} \sum (P_n) * 100}{\sum (T_n) - \sum (NA_n)}$$

Where Yn = Number of Yes answers
Pn = Number of Partially answers
Tn = Total Number of the questions
NAn = Number of N/A answers.

The following table 3 shows the general idea of analysing the questionnaires.

Answers	No of Answers	Total Questions	Total of Answers Expect	KPA rating
			N/A	
Yes	3	7	6	83.33 %
Partially	2			
No	1			
N/A	1			

Table 3 General idea of analysing the questionnaires

From table, 83.33 % in the KPA rating is representing the capability level of the assessed KPA. The interpretation of this as following

- ***Fully Achieved****:* 86% to 100% there is evidence of a complete and systematic approach to and full achievement of the defined key practices in the assessed KPA. No significant weaknesses exist across the defined organisation unit.
- ***Largely Achieved:*** 51% to 85% there is evidence of sound systematic approach to and significant achievement of the defined key practices in the assessed KPA. Performance of the key practices may vary in some areas.
- ***Partially Achieved:*** 16% to 50% there is evidence of sound systematic approach to and achievement of the defined key practices in the assessed KPA. Some aspect of achievement may be unpredictable.
- ***Not Achieved:*** 51% to 85% there is little or no evidence of achievement of the defined key practices in the assessed KPA.

V Mapping the Area of Improvement with Knowledge Based Best Agile Practices

Current software process improvement models or CMM models are not compatible or difficult to identify the area of improvement for the agile software development practices and as we discussed earlier they are not well addressed to mature the requirements engineering practices. Therefore we suggested using the knowledge of the best story cards based requirements engineering practices that have proven successful in solving problems. Consider the following figure 6, which shows how the identified are of improvements are mapped with the knowledge based best story cards based requirements engineering practices.

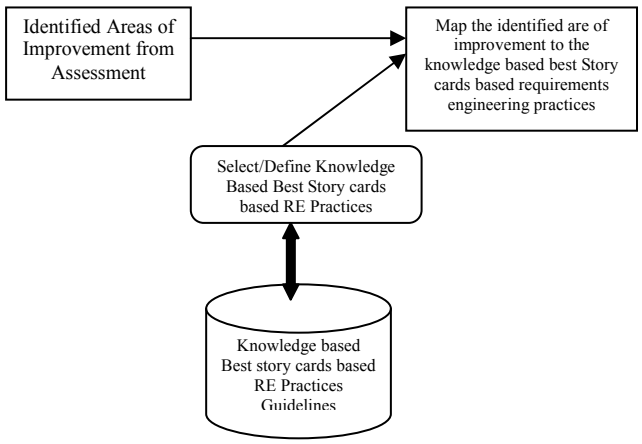


Figure 6 Capturing and mapping are of improvement with story cards based best RE Practices

The figure 6 is the conceptual framework and it is mainly concerned with capturing and enhancing the knowledge of story cards based RE practices. The primary concern of the framework is how the process improvement knowledge is captured or identified, how this knowledge is being stored, and how this knowledge of existing story cards based RE practices maps to the identified area of improvement. This guide is mainly concerned with the solving particular problems covered during the story cards based RE practices assessment, and enhances those related story cards based RE practices.

KPA	Areas for Improvements	User stories	Acceptance Testing	Release Planning	Planning Game	Product Backlog	Information Radiator	Daily Meeting	On-site Customer	Self organising team	Task & Effort estimation	Task on info. radiator	Reflection Workshop	Interviews for elicitation	Observation of story cards
2.6 Story Cards change management	2.6.1 Story card is going to divide into task cards			■	■						■	■			
	2.6.2 Story card is testable	■	■												
3.2 Refine Story cards	3.2.1 Story cards must come with acceptance tests	■	■						■						
	3.2.2 Story cards must represent the business requirements	■							■						
4.2 Story cards estimation planning	4.2.1 Story cards are easy to estimate	■	■								■				
	4.2.2 Provide a basis for estimation, cost and schedule			■	■						■				
	4.2.3 Use the past story cards estimation from the similar project area					■					■				

Table 4 Example of mapping process improvement best agile RE practices to area of improvement

In table 4, example cells shaded in grey means the corresponds practices in the header of the highlighted cell is mapped to the identified area of improvement within the row of the highlighted cell. That means the identified area of improvement takes the shaded correspondent practices as suggested by the agile team to be suitable to improve the identified area of improvement

We have developed a tool, story cards maturity model for measuring the success of story cards based RE process and also its impact on software process improvement models like CMM (Capability Maturity Model). The purpose of this form is to enable people with little or even no knowledge of story cards based RE practices, to estimate quickly easily whether story cards based RE methodology will fulfil their needs and requirements. The program consists of a form containing a handful of simple questions. The answers from these questions will provide immediate feedback on whether agile requirements practices are appropriate for the person who answered the question.

The form will ask questions about the critical areas surrounding agile requirements engineering practices; particularly story cards based requirements engineering practices. We need to identify with as few questions as possible whether story cards practices are, or are not appropriate. The following aspects have been identified as critical for story cards based requirements practices:

- •team size
- • client on site
- • team location

In order to provide a somewhat more subtle analysis, the following (less critical) aspects have also been selected:

- • requirements volatility
- • facilities strategy

Figure 7 is the illustration of our tool support which provides a web interface and online assessment forms to assess suitability for introducing story cards based RE practices into any organisation. The interface has been made simple thus allowing a first time user to fill in the form right away and getting a result within a few minutes. The results will be colour coded to help result interpretation and a summarised result will also be available. We have developed a web based tool which provides an assessment and analysis for migrating to agile requirements.

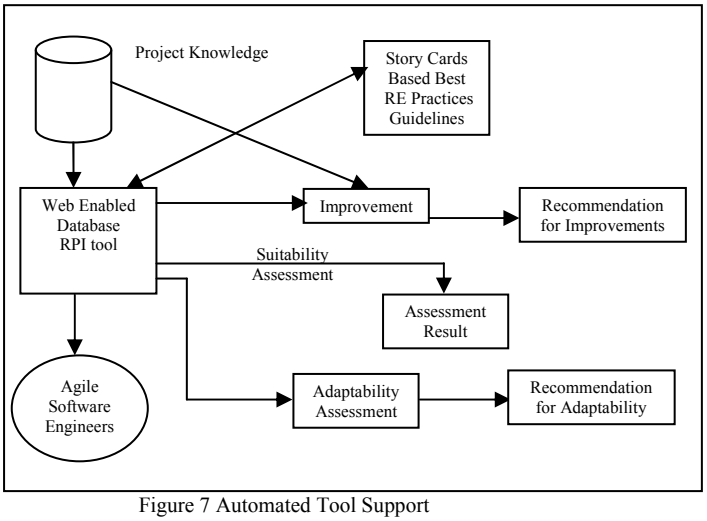


Figure 7 Automated Tool Support

VII RESULTS

We discussed our approach with three different organisations. The following table 5 summarize the participating companies.

	Type of company	Business Area	Total number of employees	Number of software developers
Company A	Independent	Flyer Design	28	11
Company B	Independent	Software House	23	9
Company C	Independent	Web development and hosting	19	12

Table 5 Participating companies

We are still at an early stage of this project, so conclusions are necessarily tentative, and based on informal observation and discussion. All of the technical managers were very supportive of the idea of Agile requirements process improvement framework for agile software development means Story card Maturity Model (SMM) were found in all the companies. Business managers tended to be somewhat more sceptical, and will require evidence of payback before becoming fully convinced of the usefulness of this approach. There was general acceptance and enthusiasm for a more quantitative approach. Company A is now well into the implementation phase of their SMM programme, and already report improvements in project planning and Agile requirements engineering process. However more analysis will be needed to determine if this is in fact a direct result of the improvements initiated as part of the SMM programme. It is important that improvements are applied in key process areas that will provide visible payback within a fairly short period. Certainly there should be measurable benefits visible with about a year from the outset, or else confidence and support for the SMM programme will be eroded. In all companies baseline measurements are being put in place that will allow us to measure the return on investment, and this will be the principal means by which we will evaluate the effectiveness of our approach.

VIII Conclusion

The capability maturity models for software and process improvement were applicable to the agile methods (agile RE Process as well) or not, this is a still challenging issue in the field of the software engineering. In this paper we describe why and how we have adapted the Agile requirements engineering process improvement framework and story card maturity model to focus on agile RE practices of agile software development methodology. We demonstrate an improvement methodology through a series of models that focus on the adaptability, suitability and improvement process of agile RE practices. Here we demonstrate how organisation can switch into agile organisation. In this paper we also developed questionnaires for each level which will identify the key process area for improvement and which best knowledge based agile RE practice need to be considered to improve that KPAs by mapping the Area of Improvement with knowledge based Best Agile RE Practices. This paper will provide a foundation for future development in the area of Agile RE process improvement for agile software development.

IX Future Work

A validation study of our SMM model is going to carry out with a group of experts in both research and industry. Future work includes creating a more flexible and an automated tool for an assessment to identify the KPAs or area of improvement for agile RE practices. Verification and evaluation is still required, and future work includes testing the model in an industrial setting.

Appendix A

Maturity Level	**KPAs**	**Areas for Improvements**
Level 1: Initial Level	No KPAs defined at this level	no process improvement goals defined at this unstructured level
Level 2	2.1 Story cards Elicitation Planning	2.1.1 Elicit Goal
		2.1.2 Elicit Functional requirements
		2.1.3 Define the scope of the story cards
		2.1.4 Identify non-functional requirements
	2.2 Defines standard structure of the story cards.	2.2.1 identification of structure fro the story cards
		2.2.2 Story cards Description section
		2.2.3 Story card is two to three sentences long, which include the summary or abstract of the user requirements requirement
		2.2.4 Story cards unique number
		2.2.5 Date of capturing
		2.2.6 Small and complete
		2.2.7 Other factors being considered
	2.3 Introduction of the validation of story cards	2.3.1 Requirement is traceable (Requirements traceability is ensured)
		2.3.2 Presented by the customer
		2.3.3 Specified acceptance tests criteria

	2.4 Stockholder's identification	2.4..1 conducted your own research to identify the stake holders of the story cards or requirements presented on the story cards
		2.4.2 On site customer is a domain expert
		2.4.3 Consider the stakeholders
		2.4.4 Identified where the system is going to be used
		2.4.5 There is a on-site customer to make and present the requirements
	2.5 Focus on the basic value of the story cards	2.5.1 Story card is simple
		2.5.2 Story card is testable
		2.5.3 Story card (Requirements presented on story card) is negotiable.
		2.5.4 Story card is valuable to customer
		2.5.5 Story card is easy to estimate
	2.6 Story Cards change management	2.6.1 Story card is going to divide into task cards
		2.6.2 Story card is testable
		2.6.3 It is easy to change story cards
		2.6.4 Story cards are negotiable
		2.6.5 Define modifiable story cards
Level 3	3.1 Dependencies, Interaction and Conflict management	3.1.1 Story cards are independent
		3.1.2 Define an unambiguous story cards
		3.1.3 Story card is consistent
		3.1.4 Use language simply, consistently and concisely
		3.1.5 considered the influencing factors from the human domain are when writing story card
	3.2 Refine Story cards	3.2.1 Story cards must come with acceptance tests
		3.2.2 Story cards must represent the business requirements
	3.3 Story cards prioritisation based on agile software development values	3.3.1 Have you got the plan or tool for story cards prioritisation
		3.3.2 Are you prioritise story cards based on the XP principles and values
		3.3.3 Are you able to prioritise story cards at any stage
		3.3.4 Are you reprioritise story cards when new story cards are arrived
		3.3.5 Do you re-prioritize story cards when any kind of changes occurs?
	3.4 Stakeholders are consulted through on site customer OR Story cards from multiple viewpoints.	3.4.1 Try to link story card requirements and stakeholders
		3.4.2 On-site customer is there until the date of delivery
		3.4.3 Write story cards from multiple view point (Collect requirements from multiple viewpoints)
	3.5 Non functional requirements consideration	3.5.1 Specify quantitative requirements also on the story cards
		3.5.2 Peer-reviews for NFR
	3.6 Database system for story cards	3.6.1 Use a database system to store story cards compared to write them and throw to destroy
		3.6.2 Do you use the story cards from other project developed in the same area
	3.7 Use of checklist for story cards based RE analysis	3.7.1 Use checklist to do requirements or story cards analysis
		3.7.2 Check the story cards meet the Agile/XP Principles and values
Level 4	4.1 Domain analysis	4.1.1 Look for domain analysis
		4.1.2 Define specialised terms used in the story cards
		4.1.3 Define the system

		boundaries during the domain analysis
	4.2 Story cards estimation planning	4.2.1 Story cards are easy to estimate
		4.2.2 Provide a basis for estimation, cost and schedule
		4.2.3 Use the past story cards estimation from the similar project area
	4.3 Plan for conflicts and conflicts resolutions	4.3.1 Uniquely identified each story cards and user stories on story cards
		4.3.2 Do you have any unique identifier plan or tool or technique for each story cards
		4.3.3 User stories are written by customer and programmer after the mutual understanding of them
		4.3.4 Reduce the development efforts
	4.4 Risk Assessment	4.4.1 Problem understanding
		4.4.2 Assess the story cards risk
		4.4.3 Do you apply the risk assessment to the story cards
	4.5 Release Planning	4.5.1 Have you classified story cards for the iteration planning
		4.5.2 Release planning is set by customers depends on the story cards
		4.5.3 Are you re-prioritise story cards with regularity
	4.6 Advance validation of story cards.	4.6.1Define or propose validation checklist for story cards
		4.6.2 Assess the story cards correctness
		4.6.3 Provides the baseline for the validation and verification
		4.6.4 Do you conduct the story cards review with on site customer and development team?
		4.6.5 Do you plan to write acceptance tests same time on as capturing story cards?
		4.6.6 Do you apply the technique for story cards inspections?
		4.6.7 are you considering non-functional requirements on story cards
		2.6.8 Do you quantify the non-functional requirements

REFERENCES

[1] Agile Manifesto, (2006) Manifesto for Agile Software Development. [internet], <http://agilemanifesto.org> [Accessed last 01-03-2006]

[2] Ahern, D. M., Clouse, A. and Turner, R. (2003) CMMI Distilled: A Practical Introduction to integrated Process Improvement 2nd ed. UK Addison Wesley.

[3] Beck, K., (2000) Extreme Programming Explained: Embrace Change, Addison- Wesley Press.

[4] Beecham, S., Hall, T. and Rainer, A. (2003), Defining a Requirements Process Improvement Model.. Software Quality Journal Volume 13 Number 13 septembre 2005, Springer, pp.247-279.

[5] Boehm, B. (2003), Value-Based Software Engineering ACM SIGSOFT Software Engineering Notes, 28(2) March 2003, pp 3-15

[6] Boehm, B., Port, D., Jain, A., and Basili, V. (2002), Achieving CMMI Level 5 Improvements with MBASE and the CeBASE Method. [Internet] Cross Talk Journal, Available from : <http://www.stsc.hill.af.mil/CrossTalk/2002/may/boehm.asp> [Accessed 11-10-2007]

[7] Bohem, B (1981) Software Engineering Economics 1981 Prentice Hall

[8] Brodman, J. and Johnson, D (1997), A Software Process Improvement Approach for small organisation and small projects. Proceedings of the 19th International Conference in Software Engineering, 19th may 1997, Boston- MA, ACM Press, pp661-662.

[9] Casey, V. and Richardson, I. (2002), A Practical Application of Ideal Model. Product Focused Software Process Improvement, 4th International Conference (PROFES), December 9-11, Rovaniemi – Finland, Springer, pp.172-184.

[10] Chrissis, M. B., Konrad, M. and Shrun, S. (2003) CMMI: Guidelines for Process Integration and Product Improvement, UK, Addison Wesley.

[11] Chrissis, M. B., Wemyss, G., Goldenson, D., Konrad, M., Smith, K. and Svolou, A. (2003) CMMI Interpretive Guidance Project : Preliminary Report [Internet], Software Engineering Institute, Available from : <http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03sr007-body-revised.pdf> [Accessed 01-06-2004].

[12] Christie, A. M. (1999), Simulation in support of CMM-based process improvement. Journal of Systems and Software, (46): 107-112.

[13] Cockburn, A., (2004) Crystal Clear A Human-Powered Methodology for Small Teams, and it Addison- Wesley Press.

[14] Dangle, K., Larsen, P. and Shaw, M. (2005) software process improvement in small organisation: A case study, IEEE Software November/December 22(6) pp 68-75.

[15] Dutta, S., Lee, M. and Wassenhove, L. K. (1999) Software Engineering in Europe : A study of best practices, IEEE Software Volume (16), Issue(3).
[16] Dyba, T. (2003), Factoes 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 held jointly with 10th ACM ACM SIGSOFT International symposium on foundations of software Engineering, September 2003, Helsinki – Finland, ACM Press, pp.148-157.
[17] Faulk, S (1996) Software Requirements: A Tutorial. Software Engineering 1996 M.Dorfman and R. H. Thayer Eds. pp 82-103.
[18] Glib, T. (2003), Software Project Management Adding Stakeholder Metrics to Agile Projects. The European Journal for the Informatics Professional, IV(4) August 2003, pp.5-9
[19] Gorschek, T. and Tejle, K. (2002) A Method For Assessing Requirements Engineering Process Maturity in Software Projects. Master Thesis, Blekinge Institute of Technology.
[20] Herbsleb, J. D. and Goldenson, D. R. (1996), A Systematic Survey of CMM experience and results. Proceedings of the 18th international conference on Software Engineering, May 1996, Berlin, Germany, IEEE Computer Society, pp.323-330.
[21] Highsmith, J., (2004) Agile Project Management, Creating innovative products, Addison- Wesley.
[22] Ihme, T. and Abrahamsson, P., (2005) The Use of Architechural Patterns in the Agile Software Development of Mobile Applications, International Journal of Agile Manufacturing, Vol. 8, issue 2, 97-112.
[23] Johnson, J., Boucher, K. D., Connors, K. and Robinson, J. (2001), Collaborating on Project Success. [Internet], Software Magazine, Available from : <http://www.softwaremag.com/l.cfm?Doc=archive/2001feb/collaborativeMgt.html> [Accessed 20-jan-2008]
[24] Kotonya, G. and Somerville, I. (1998), Requirements Engineering – Processes and Techniques, John Wiley & Sons, Chichester, UK.
[25] Li, E., Chen, H. and Lee., T. (2002), Software Process Improvement of Top Companies in Taiwan:a comparative study. Total Quality Management 13(5) March 2002, pp701-703.
[26] Lyard, A. and Orci, T. (2000), Dynamic CMM for Small organisations. Proceedings ASSE 2000, the first Argentine Symposium on Software Engineering, September 2000, Tandil- Argentina, pp.133-149.
[27] Nawrocki, J., Walter, B. and Wojciechowski, A. (2001), Towards the maturity model for extreme programming, 27th Euromicro Proceedings 4-6 September 2001, pp.233-239
[28] Paulk, M. C. (2001) Extreme Programming from a CMM Perspective IEEE Software 18(6) November/December 2001, pp 19-26
[29] Paulk, M. C. (2001), Extreme Programming from a CMM Perspective, IEEE Software November-December 2001
[30] Paulk, M. C., Weber, C. V., Curtis, B. and Chrissis, M. B. (1995) The Capability Maturity Model for Software : Guidelines for Improving the Software Process (SEI). USA, Addison Wesley.
[31] Paulk, M., Curtis, M., and Weber, C.,(1993) Software Process Maturity Questionnaire : Capability Model version 1.1. [internet], Carnegie Mellon-Software Engineering Institute, Available from <http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html> [Accessed 11-01-2007]
[32] Paulk, M., Weber, C. and Curtis, M.(1999), The Capability Maturity Model for Software. In K. Emam & N. Madhavji (Eds.), Elements of software process Assessment and Improvement. IEEE Computer Society Press, PP.3-22.
[33] Paulk, M.C. (1998), Using the Software CMM in Small Organisations. The Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality, 13-14 October 1998, Portland, USA, Software Engineering Institute, PP.350-361.
[34] Persse, J. R. (2001) Implementing the Capability Maturity Model, USA, John Wiley & Sons Inc.
[35] Pikkarainen, M. and Mantyniemi, A., (2006) An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies, SPICE 2006 conference, Luxemburg.
[36] Raynus, J. (1999) Software process improvement with CMM, London, Artech House
[37] Schwaber, S. and Beedle, M., (2002) Agile Software Development With Scrum, Prentice Hall.
[38] Somerville, I. and Sawyer, P. (2000), Requirements Engineering – A Good Practice Guide, John Wiley & Sons, Chichester, UK
[39] Tim, K. (2004) Practical insight into the CMMI, USA Artech house
[40] Turner, R. and Jain, A. (2002) Agile meets CMMI: Culture clash or common cause XP/Agile universe 2002, LNCS 2418 pp. 60-69.
[41] Wiegers, K. First Things First: Prioritizing requirements software Development Vol. 7, no 9 Sept 1999 Available from <http:// www.processimpact.com/pubs.shtml#requirements > [Accessed 11-01-2007]
[42] Zaharan, S. (1998) Software Process Improvement: Practical Guidelines for Business Success, 1st. USA, Addision-Wesley.

Chetankumar Patel is a research student in the software engineering research group at Leeds Metropolitan University,. His research interest is on Agile software development methodology, Extreme Programming, story (Requirement Engineering Process for XP) and agile process improvement. Contact him at c.patel@leedsmet.ac.uk

Dr. Muthu ramachandran is a principal lecturer at Leeds metropolitan university. Contact him at m.ramachandran@leedsmet.ac.uk