AGILE CMMI
from SMEs perspective

Ahmed Omran
Arab Academy for Banking and Financial Science http://www.aabfs.org
Ahmed.omran.sy@gmail.com

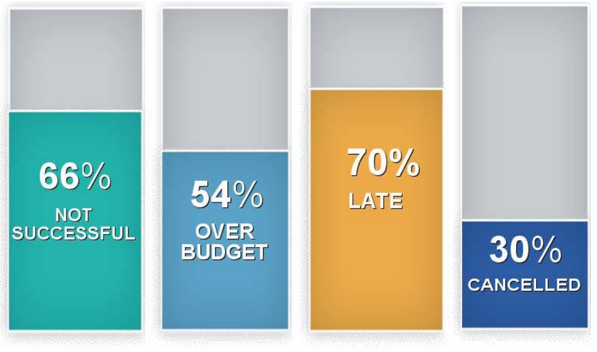
***Abstract*— CMMI has been well accepted by the world as a standard for project development process control, quality improvement, and capacity evaluation. More and more companies are involved in the CMMI certifications rush. However, to most of them, CMMI is a very challenging goal. CMMI becomes critical to seek a suitable way to promote the competence of the small- and mid-sized companies in worldwide competition.**

This research suggests that small and mid sized companies can adopt Agile Development Methodologies (ADM) while following the CMMI standard if obtain a new additional values, thus they can get a new competence value in their environments.

Keywords-component; Software development framework; CMMI; SMEs; agile methodologies; XP;

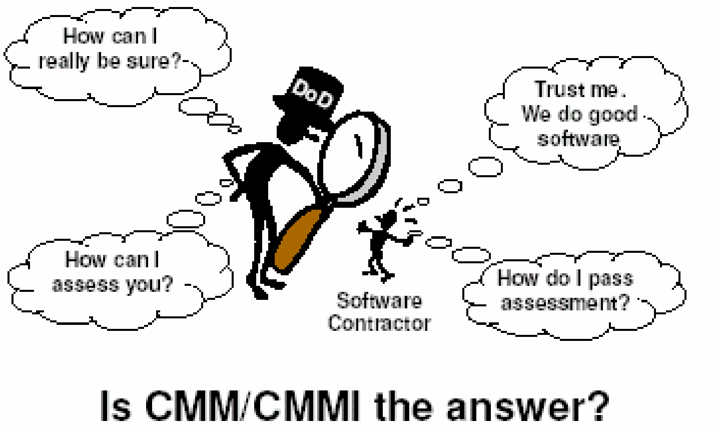
I. INTRODUCTION

According to the three dimensions in software management (project, process, people); software development evolved through three different eras. First era; Project was the primary software development focus in the 1960s when the software development discipline was new. At the end of this era software crises emerged, empirical and academic studies began to characterize the problem. Second era; The early 1970s brought a shift in focus to the development process. The emphasis on the Waterfall Model in software development defined and enforced through standards. However although these formal methodologies may encourage logical thinking and programming, they are not the silver bullet that developers are looking for.



Effects in Applications Development – real project results

Third era: In the mid-1980s introduced the Software Engineering Institute's Capability Maturity Model® as an approach to stabilizing the development process and improving quality and productivity. CMM and CMMI were the frameworks that Department of Defense (DoD) uses to evaluate its contractors.


How can I really be sure?
DoD
Trust me. We do good software
How can I assess you?
Software Contractor
How do I pass assessment?
Is CMM/CMMI the answer?

However both Methodologies and their frameworks still have many problems. First problem; empirical researches show that methodologies used in practice is rather limited. Second problem; those developers who do use methodologies tend to use different combinations and parts of methodologies rather than following all the steps required by a particular methodology. Third problem; researches have also indicated that methodologies are not universally applicable, are too bureaucratic, and unable to handle people factors.

As a result to these problems and other issues there was a tendency to bypass Analysis&Design and proceed directly to implementation. This trend is accepted largely in the new environments, especially with the systems that use more complex technologies such as Web systems and hypermedia.

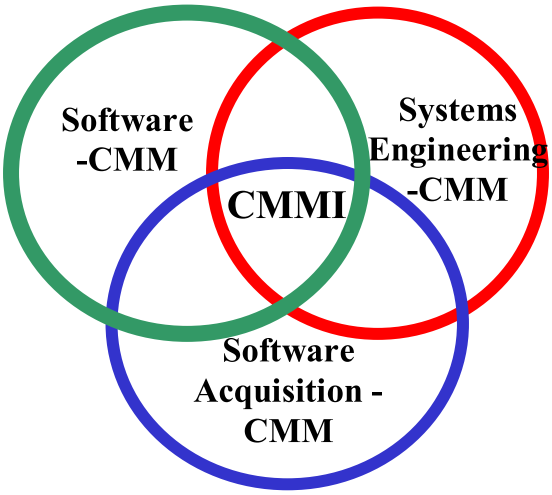
As a result of this shift, Agile-Methods have appeared. These methods advocate a return toward less formal approaches to software development. Agile methods (XP, Scrum, Crystal…) are still evolving and recently many academic researchers have focused on this area.

II. CMMI

A. What is cmmi!?

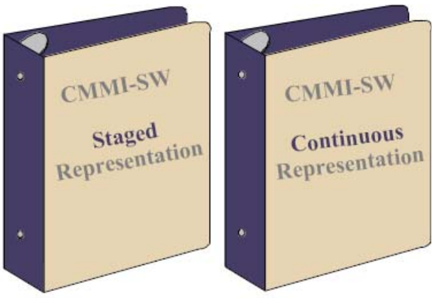
A model is a simplified representation of the world, and The Capability Maturity Model (CMM) is a method for evaluating and measuring the maturity of the software development process of organizations on a scale of 1 to 5. Capability Maturity Model Integration (CMMI) model provides guidance to be used when developing processes. CMMI model is not processes or process descriptions. [15]

The CMMI is a suite of products for process improvement that have been built with cross model interoperability as a goal. In theory, the user of the CMMI can choose a product that includes the specific disciplines that fit the users needs. [9]



B. Staged or Continuous

CMMI has two representation, continuous or staged, and it must be selected in order to determine the body of knowledge that will be used in the model the organization. The CMMI is both a capability model and a maturity model. The model can be implemented in a staged fashion. This is the idea that an organization achieves maturity level 2 across all relevant practices and then level 3 and so on. It can also be implemented in a continuous fashion where capability in individual practices is measured. An organization can achieve a high capability level in a particular practice whilst holding only a level 2 capability in other practices.



It is the staged method of assigning maturity levels that is associated with the American government contracting market and is hence best known in North America. [3]

In CMMI model with a staged representation, there are five maturity levels:

1. Initial
2. Managed
3. Defined
4. Quantitatively Managed
5. Optimizing

Level	Process Areas
1- Initial	• None
2- Managed	• Requirements Management • Project Planning • Project Monitoring and Control • Supplier Agreement Management • Measurement and Analysis • Process and Product Quality Assurance • Configuration Management
3- Defined	• Requirements Development • Technical Solution • Product Integration • Verification • Validation • Organizational Process Focus • Organizational Process Definition • Organizational Training • Integrated Project Management • Risk Management • Decision Analysis and Resolution
4- Quantitatively Managed	• Organizational Process Performance • Quantitative Project Management
5- Optimizing	• Organizational Innovation and Deployment • Causal Analysis and Resolution

C. *Does higher level means better software project?*

What Software Development Maturity Really Means?

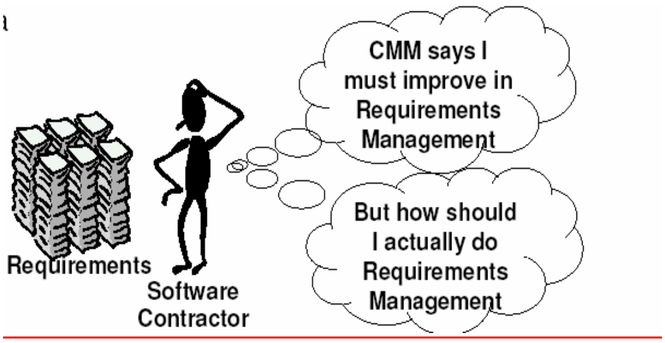
- Developing Really Good project..!?

- Achieving Real Process Improvement..!?

1) *What do individuals and small organizations say?*

"Organization even at the high levels of maturity may produce products with usability problems."[12]

"It is easy to make a good process measurable; but it is hard to make a measurable process good" [6]


CMM says I must improve in Requirements Management
But how should I actually do Requirements Management
Requirements
Software Contractor

The main issues that may be obstacles to development teams in the SMEs:

- CMMI focuses on project management rather than product development.
- CMMI ignores the use of some technologies such as rapid prototyping.
- CMMI does not incorporate risk analysis as a key process area.
- CMMI is expensive to use; too much overhead and reporting for a small company.

2) *What do individuals and small organizations say?*

- Boeing Australia, Limited: 1400 people in 13 locations in Australia; started implementing CMMI in 2001; 33% decrease in the average cost to correct a defect; 50% reduction in time for delivery of a release; Increased ability to configure builds; 60% reduction in the preparation, conduct, and rework form pre-test to post-test audits. [4]
- Lockheed Martin Management & Data Systems: 9,000 employees in Pennsylvania, Arizona, California, Colorado, and Washington DC; Deeply involved in process improvement for more than ten years; achieved CMM Level 5 in 2000 (SE-CMM and SW-CMM); achieved CMMI Level 5 in 2002; notable performance improvements; 30% increase in software productivity; 20% decrease in unit software cost; 15% decrease in costs to find and fix defects. [4]

III. NEW AGILE METHODOLOGIES

A. *Introduction*

Arrays of light rose on the SMEs through new concepts of agility development, and they find the wide road to success in these methods. “Agile means investing heavily in individual skill-building rather than organizational rule sets.” [13] Agile methods derive much of their agility by relying on the tacit knowledge embodied in the team, rather than writing the knowledge down in plans. [2] In the late 1990's several methodologies began to get increasing public attention. Each had a different combination of old ideas, new 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 changeable requirements was not a crisis. Early 2001 saw a workshop in Snowbird, Utah, USA, where various originators and practitioners of these methodologies met to figure out just what it was they had in common. They picked the word "agile" for an umbrella term and crafted the Manifesto for Agile Software Development, whose most important part was a statement of shared development values:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *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. [14]

Agile manifesto emerged and methodologies began to be discussed, one of the most common method with a lot of papers emerged from the research labs was "XP methodology". eXtreme Programming (XP) is the name that Kent Beck has given to the lightweight team process which he has been evolving over the years. In XP, we use a very lightweight combination of practices to create a team that can rapidly produce eXtremely reliable, efficient, well-factored software. Many of the XP practices were created and tested as part of the Chrysler C3 project, which is a very successful payroll system implemented in Smalltalk.[14]

B. *eXtreme Programming*

eXtreme Programming (XP) is a disciplined approach to software development. XP is successful because it stresses customer satisfaction. The methodology is designed to deliver the software your customer needs when it is needed. XP empowers your developers to confidently respond to changing customer requirements, even late in the life cycle. This methodology also emphasizes team work. Managers, customers, and developers are all part of a team dedicated to delivering quality software.

1) *XP's Activivties*

XP improves a software project in four essential ways:

1. Communication: developers communicate with their customers effectively.
2. Simplicity: Design and models should be simple and clean.
3. Feedback: developers feedback by testing their software starting on day one.
4. Courage: courageously respond to changing requirements and technology.[16]

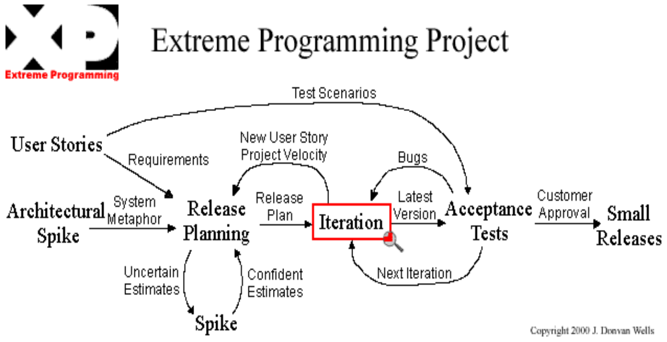
2) *Principles in practice*

The method typically consists of 12 basic elements: Planning game: The customer decides scope, priority, and dates from a business perspective, whereas technical people estimate and track progress.

1. Small releases: Put a simple system into production quickly. Release new versions on a very short (two-week) cycle.
2. Metaphor: Guide all development with a simple story of how the overall system works.
3. Simple design: Design as simply as possible at any given moment.
4. Testing: Developers continually write unit tests; customers write tests to demonstrate that functions are finished. “Test, then code” means that a test case is an entry for writing code.
5. Refactoring: Restructure the system without changing its behavior to remove duplication, improve communication, simplify, or add flexibility.
6. Pair programming: All production code is written by two programmers at one machine.
7. Collective ownership: Anyone can improve any system code anywhere at any time.
8. Continuous integration: Integrate and build the system many times a day (every time a task is finished). Continual regression testing prevents functionality regressions when requirements change.
9. 40-hour weeks: Work no more than 40 hours per week whenever possible; never work overtime.
10. On-Site customer: Have an actual user on the team full-time to answer questions.
11. Coding standards: Have rules that emphasize communication throughout the code.[14]

C. *How do I start this XP thing?*

Start out collecting user stories and conducting spike solutions for things that seem risky. Spend only a few weeks doing this. Then schedule a release planning meeting. Invite customers, developers, and managers to create a schedule that everyone agrees on. Begin your iterative development with an iteration planning meeting. Now you're started. Arrays of light rose on the SMEs through new concepts of agility development, and they find the wide road to success in these methods. “Agile means investing heavily in individual skill-building rather than organizational rule sets.” [13] Agile methods derive much of their agility



IV. Agile CMMI

Now the problem in SMEs is: How can SMEs develop their projects while following the framework standards? Some standards are overloading the SMEs because it is originally organized for the large organizations, so it is useless to guide to standards that have no values for the SMEs. They really need their own frameworks that enable them to compete in the international environments.

As an alternative solutions for this dilemma, some suggested to cerate a new XCMMI framework.[10] While others suggest a new Maturity model for SMEs which based on a scaled-down CMM or scaled-up PSP[5]

We would explain the overlapping of (XP,CMMI) from the SMEs activities; as an example of how SMEs can apply a framework standards while keeping the values of their agile activities.

The discussion is conducted according to the questionnaire and interviews in Small and Mid-size Enterprises. The result of the survey was none of the SMEs can easily apply all the KPAs.

V. XP and CMMI

“Although many of their advocates consider the agile and plan-driven software development methods polar opposites, synthesizing the two can provide developers with a comprehensive spectrum of tools and options.” Barry Boehm University of Southern California [6]

"XP and CMMI can work together for companies that need that level of reassurance" Kent Beck

“At Microsoft, we’ve stretched our MSF (Microsoft Framework) for Agile Software Development method to fit the requirements for CMMI Level 3."[3]

Some people think that XP and CMMI are like oil and water, while others believe that running concurrently those two software approaches (XP and CMMI) allowed us to get projects with both advantages with decreased risk failures. This section discusses each KPA (from SMEs perspective) if it is applicable with XP principles or not; then a table in the appendix simplify the results of mapping the KPAs with XP activities.

A. XP and level 1

- Identify the scope of the work
- Perform the work

Then the organization can be rated as CMMI level 1. There no KPA in this level (this doesn't mean that bad practices can be rated as CMMI level1).

B. XP and level 2

1) Requirements Management

The purpose of Requirements Management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products, i.e. the ability to define and mange each requirement should be proved.

XP addresses Level 2's requirements management KPA through its use of user-stories, an On-Site customer, and continuous integration. Although system requirements might evolve dramatically over time, XP integrates feedback on customer expectations and needs by emphasizing short release cycles and continual customer involvement. "Common understanding" is established and maintained through the customer's continual involvement in building stories and selecting them for the next release.

2) Project Planning

The purpose of Project Planning is to establish and maintain plans that define project activities.

XP addresses software project planning in the planning game and small releases. And always: in XP's planning strategy "If you can't plan well, plan often." However small projects need small plans, consequently SMEs using planning game can plan its projects well.

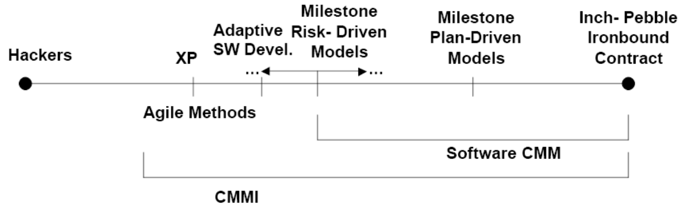

Hackers
XP
Adaptive SW Devel.
Milestone Risk- Driven Models
Milestone Plan-Driven Models
Inch- Pebble Ironbound Contract
Agile Methods
Software CMM
CMMI

Figure 1. The Planning Spectrum [2]

3) Project Monitoring and Control

The purpose of Project Monitoring and Control is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

XP addresses software Project Monitoring and Control with the "big visual chart," where the velocity of the project is stated clearly, and commitments (stories) for small releases. This visual chart usually developed by both customer and the XP team.

4) Supplier Agreement Management

The purpose of Supplier Agreement Management is to manage the acquisition of products from suppliers for which there exists a formal agreement.

SMEs can not apply such KPA, and it seems to consumes lot of the resources from small teams.

5) Measurement and Analysis

The purpose of Measurement and Analysis is to develop and sustain a measurement capability that is used to support management information needs.

XP claims that refactoring makes such software Measurement and Analysis through its refactoring practice where we altering internal structure of the system without changing its external behavior. [9]

6) Process and Product Quality Assurance

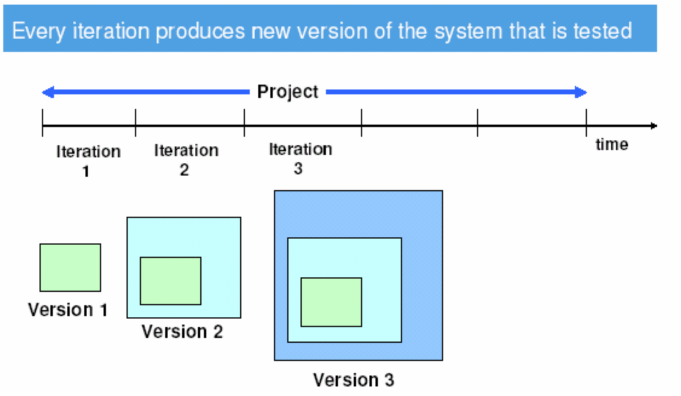
The purpose of Process and Product Quality Assurance is to provide staff and management with objective insight into processes and associated work products.

SQA could be addressed by the pair-programming culture. Peer pressure in an XP environment can achieve SQA's aim of assuring conformance to standards. However, larger teams typically require formal mechanisms for objectively verifying adherence to requirements, standards, and procedures. While in small and mid size organizations SQA can be merged into the organizational processes.

7) Configuration Management

The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

Although this not completely and explicitly addressed, Software Configuration Management can be implied in many XP's activities: Planning game, collective ownership, small releases, and continuous integration all these lead to detailed control and *software configuration management.*



C. *XP and level 3*

Processes are well characterized and understood, and are described in standards, procedures, tools, and methods.

1) *Requirements Development*

The purpose of Requirements Development is to produce and analyze customer, product, and product-component requirements.

On-Site customer, user stories and iterative development can manage to Requirements Development so the XP team can easily achieve this KPA.

2) *Technical Solution*

The purpose of Technical Solution is to design, develop, and implement solutions to requirements.

XP's practices can address Technical Solution through different practices (metaphor, iterative solutions, testing units) will leads to a high quality of technical solutions.

3) *Product Integration*

The purpose of Product Integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.

XP's practices emphasis on Product Integration through the planning game, iterative cycles, and unit testing.[1]

4) *Verification*

The purpose of Verification is to ensure that selected work products meet their specified requirements.

On-Site customer, user stories and iterative development will ensure this KPA.

5) *Validation*

The purpose of Validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment.

Pair programming addresses peer reviews, iterative development, unit test, On-Site customer can ensure the validation area.

6) *Organizational Process Focus*

The purpose of Organizational Process Focus is to plan and implement organizational process improvement based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.

XP addresses organization process focus at the team level rather than organizational level. XP focuses on the software engineering process rather than organizational infrastructure issues, so SMEs that adopting XP can address this and other organization-level processes.

7) *Organizational Process Definition*

The purpose of Organizational Process Definition is to establish and maintain a usable set of organizational process assets.

XP address the team process definition, but organizational assets are outside the scope of the XP method itself. So the XP team can adopt this KPA as an intellectual asset. [1]

8) *Organizational Training*

The purpose of Organizational Training is to develop the skills and knowledge of people so they can perform their roles effectively and efficiently.

XP address this KPA through its collective ownership of project that infer no one can complete his tasks without Organizational Training especially in small organizations where the main assets is the individual of these organizations so organizational training mean individual development which is applicable through the XP practices.[1]

9) *Integrated Project Management*

The purpose of Integrated Project Management is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

XP partially address integrated software management, through different practices (planning game, visual charts, iterative developments), but it is difficult to mange stakeholders outside the organization.

10) *Risk Management*

The purpose of Risk Management is to identify potential problems before they occur, so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

XP doesn't address Risk Management in wide range. When On-Site customer and XP-tester create the user-stories for such scenarios, consequently the developers will work on their project with covered scenarios. However On-Site customer and XP-testers conduct such analysis but when defining the user-stories and the acceptance-test.

11) *Decision Analysis and Resolution*

The purpose of software Decision Analysis and Resolution is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria

XP doesn't address Software Decision Analysis and Resolution. Plan where you need pan, design what is important, code what can pass the unit tests and user-story; All these main activities in XP depend on the tacit knowledge of the XP,s team so it is difficult to address this KPA.

D. *XP and level 4*

The performance of processes is controlled using statistical and other quantitative techniques, and is quantitatively predictable.

1) *Organizational Process Performance*

The purpose of Organizational Process Performance is to establish and maintain a quantitative understanding of the performance of the organization's set of standard processes in support of quality and process-performance objectives, and to provide the process performance data, baselines, and models to quantitatively manage the organization's projects.

XP can not address this KPA [8].

2) *Quantitative Project Management*

The purpose of the Quantitative Project Management process area is to quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives.

XP can not address this KPA [8].

In SMEs the XP team can tune the performance of their processes but it is not a quantitative tuning, so even in SMEs level4 KPAs can not be addressed.

E. *XP and level54*

Continually improving process performance through both incremental and innovative technological improvements.

1) *Organizational Innovation and Deployment*

The purpose of Organizational Innovation and Deployment is to select and deploy incremental and innovative improvements that measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.

SMEs address the incremental improvements that improve the processes at the team level through different practices such as (simplicity, refactoring, feedback).

2) *Causal Analysis and Resolution*

The purpose of Causal Analysis and Resolution is to identify causes of defects and other problems and take action to prevent them from occurring in the future.

XP partially address this area through: planning game, peer review, and the feedback during rapid cycles from the On-Site customer.

VI. Conclusion

From SMEs perspective "Framework" should not mean to follow strict processes, but to standardize the best practices. CMMI focuses on best practices that an organization can use to improve processes in the process areas that are within the maturity level it chooses to achieve. Consequently SMEs need their own modified frameworks that can used to add values to its practices.

The advice for SMEs: "You need to modify your processes or create new ones where you get additional values; SMEs should not create new processes with no values, just to achieve a certain KPA or just to get higher level "; "you do not need to choose between agility and software engineering. Instead, define a software engineering approach that is agile" [11]

VII. References

[1] Alan S. Koch, "CMM-compliant XP", http://www.askprocess.com/Articles/CMM-XP.pdf

[2] Barry Boehm, "Get Ready for Agile Methods, with Care", www.agilealliance.com/articles/demarcotomboehmbarryt/file

[3] David J. Anderson, " Stretching Agile to fit CMMI Level 3", Microsoft Corporation, Conference Denver July 2005, http://agile2005.org/XR14.pdf

[4] Henry Schneider, "TransitioningfromCMMtoCMMI", http://www.teraquest.com/resource/documents/Schneider-TransitioningfromCMMtoCMMI.pdf

[5] Hossein Saiedian, Natsu Carr, "Charchterising a software process maturity for small organizations", Source: ACM SIGICE Bulletin, Volume 23, Issue 1, Pages: 2 - 11

[6] Ivar Jacobson, "CMMI and UP", http://www.ivarjacobson.com/html/content/presentations

[7] Kent Beck, "Extreme programming explained: embrace change", Publisher: Addison-Wesley

[8] Kieran Conboy, "A Framework for the Study of Modeling Techniques", Workshop in Software Model Engineering, http://www.metamodel.com/wisme-2003/14.pdf

[9] PATRICK EMERY, "The dangers of extreme programming", http://members.cox.net/cobbler/XPDangers.htm

[10] Patrick Emery, "eXtreme CMMI", http://www.members.cox.net/cobbler/XCMMI.ppt

[11] Roger S Pressman, "Software Engineering: A Practitioner's Approach", 6th Edition, chapter4 Agile Development

[12] Timo Jokela, Tuomo Lalli, "Usability and CMMI: Does A Higher Maturity Level in Product Development Mean Better Usability?", Publisher: ACM.org Press

[13] Tom Demarco, Barry Boehm, "The Agile Methods Fray", Publisher, IEEE Computer Society Press

[14] http://www.agilealliance.com/intro

[15] http://www.sei.cmu.edu/cmmi/models/sw-staged.doc

[16] http://www.extremeprogramming.org

VIII. Glossary

1) *DoD: Department of Defense*

2) *KPA: Key Process Area*

3) *Chrysler C3 project: C3 was the short name of the Chrysler Comprehensive Compensation project, a payroll project at Chrysler which has since become famous as the 'birth project' of Extreme Programming.*

4) *Smalltalk: a class-based OO programming; it can be work as An operating system.*

5) *SE-CMM: System Engineering – CMM*

6) *SECAM: Systems Engineering Capability Assessment Model.*

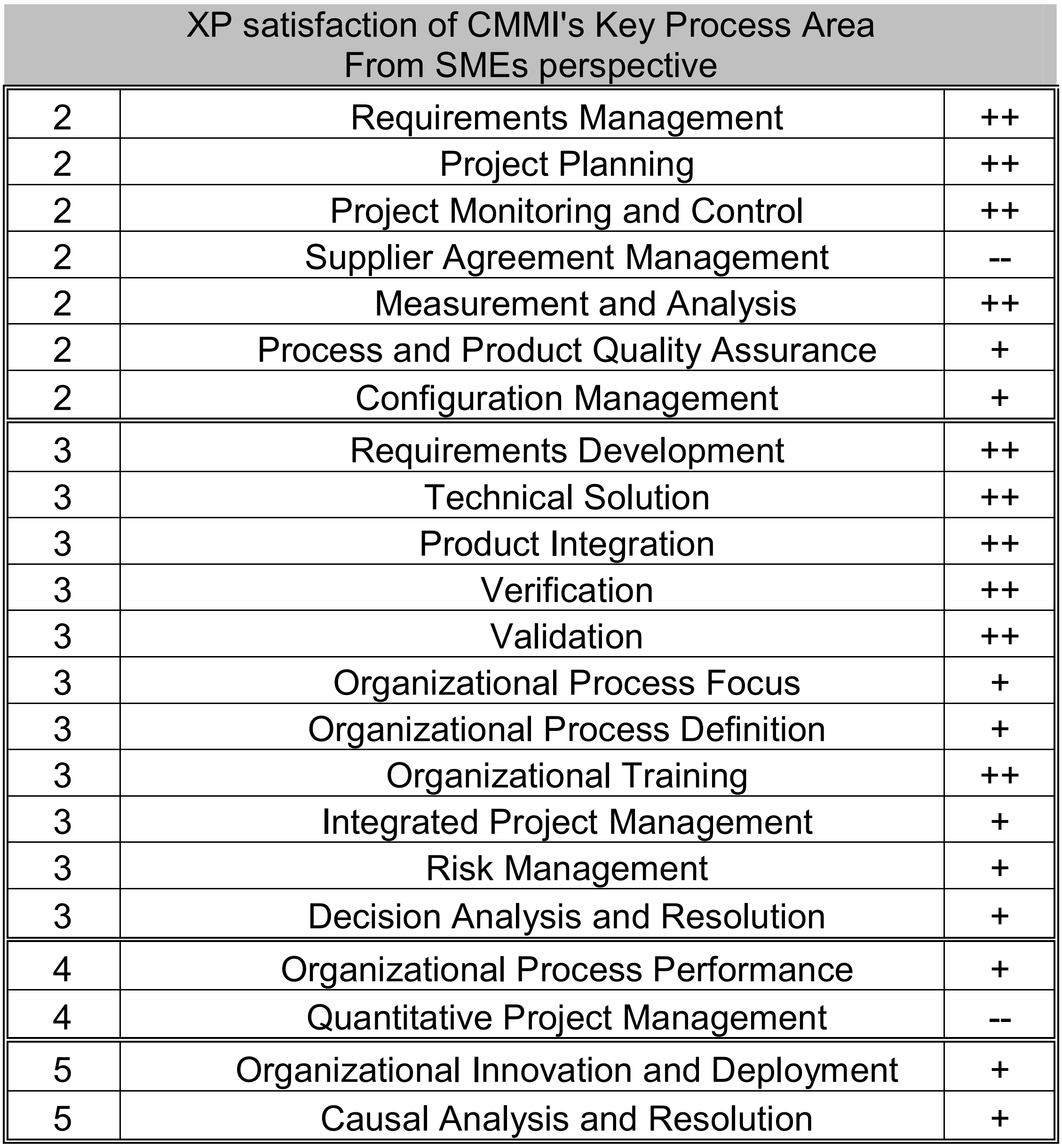
7) *SA-CMM: Software Acquisition Capability Maturity Model.*

8) *EIA/IS: Electronic Industries Association Interim Standard.*

9) *IPD-CMM: Integrated Product Development-CMM.*

10) *PSP: Personal Software Process*

IX. Appendix



XP satisfaction of CMMI's Key Process Area
From SMEs perspective

2	Requirements Management	++
2	Project Planning	++
2	Project Monitoring and Control	++
2	Supplier Agreement Management	--
2	Measurement and Analysis	++
2	Process and Product Quality Assurance	+
2	Configuration Management	+
3	Requirements Development	++
3	Technical Solution	++
3	Product Integration	++
3	Verification	++
3	Validation	++
3	Organizational Process Focus	+
3	Organizational Process Definition	+
3	Organizational Training	++
3	Integrated Project Management	+
3	Risk Management	+
3	Decision Analysis and Resolution	+
4	Organizational Process Performance	+
4	Quantitative Project Management	--
5	Organizational Innovation and Deployment	+
5	Causal Analysis and Resolution	+

(+): Partially addressed (SMEs only)
(++): Largely addressed (big and small organizations)
(--): Not addressed