ISSN: 2395-1664 (ONLINE)
DOI: 10.21917/ijms.2022.0245



# A STUDY ON ASSIGNING PERFORMERS TO PRACTICE AREAS IN SMALL SIZE SOFTWARE ORGANIZATIONS COMBINED CMMI AND XP

**Ki Son Ryang, Il Nam Mun, Chol Nam Om and Jang Su Kim**

*Institute of Information Technology, High-Tech Research and Development Center, Kim Il Sung University, Pyongyang, Democratic People's Republic of Korea*

***Abstract***

***Researches on software development process suitable for small size software organizations tend to the combination of CMMI and Agile methodology. Among them the combination of CMMI and Scrum has the most popularity. Research how we combine XP, one of main development model of agile methodology, with CMMI is still a challenge problem. In this paper, we identify performers in each practice areas and each practices when applying CMMI to small size software organizations which are experienced to XP methodology. We apply our solution to two IT organizations and compare the result with our previous work. Eventually we confirm that our novel solution needs less period of achievement of CMMI level 4.***

***Keywords:***

***Capability Maturity Model Integration (CMMI), eXtreme Programming (XP), Agile Software Development (ASD), Software Process Improvement (SPI)***

## 1. INTRODUCTION

In these days the matter of attribution and interest is how to combine CMMI with agile methodology. It results from the fact that the most of software organizations and projects are small sized.

Small size software organizations have so limited human, material, technical, financial resources available to SPI (software process improvement) that they are interested in agile methodology which seems to ensure higher efficiency and flexibility.

However, in spite of small size, more and more software organizations are forward to higher CMMI level with its own process improvement basis and make a distinct success in recent years. Also, the theoretic and realistic research for accelerating the combination of CMMI with Agile and applying to practices which were discussed since last years has more raised concern and ardor.

CMMI is inherited from CMM-SW developed by Software Engineering Institute (SEI) in order to describe principles and practices as the basis of software process maturity in 2002. CMMI is composed of the model, appraisal methods and training materials for SPI.

In past they have recognized that CMMI is suitable for only large software organizations. Because it has heavyweight processes and a burden of documentation. But in recent days small size software organizations have aimed to be appraised its capability maturity with CMMI and achieve higher level of capability maturity [1].

The standards such as ISO9001 and CMMI focus on processes, plans and documentations, while agile software development (ASD) focuses on collaboration, repetition and communication. Therefore, it has become more popular development philosophy in recent years [2].

Agile methods are lightweight methods that focus on a rapid delivery of valuable products, but the SPI standards focus on quality and documentation. So, the SPI standards are heavyweight and bureaucratic [3].

As an agile development methodology, XP is a lightweight iterative approach for small size development teams having incomplete or continuously changing requirements. Extreme Programming repeats small iterations who,m focus on close collaboration, simple design in order to produce high quality products with continuous testing [4].

We concluded that the most popular agile method was Scrum and XP through the analysis of recent research results [5], [6].

Up to now the combination of Scrum with CMMI made up the absolute majority in the combination agile with CMMI and have published many valuable research results and practices. However, the research for combining XP with CMMI have insufficiently published detailed methods and practices.

Because XP has occupied remarkable positions in agile methodology and has specific practices of its own, its superior development principles and practices have still welcomed in small size software development and the study for combining it with CMMI also take an important significance.

In this paper, we identified performers in each practice areas and a contribution rate to implement practice areas for combining XP with CMMI in small size software organizations and validate its effectiveness.

The rest of the paper is as follows. We briefly describe research results about the combination of CMMI with agile methods in section 2. In section 3 we define a performers in practice areas, a contribution rate to implement practice areas for combining XP with CMMI. In section 4 we present some case studies which demonstrate the effectiveness of our results and section 5 concludes the paper.

## 2. BACKGROUND AND RELATED WORK

### 2.1 CMMI

International standards and models which are widely used to support software process improvement are CMMI (Capability Maturity Model Integration), ISO 9001, ISO 15504 (so called SPICE) and etc [7]-[9].

Among them CMMI is world-famous. Based on the first version released in 1991, CMMI has been presented in 2000, integrating CMM for Software (SW-CMM), the Capability Model for Systems Development (EIA/IS 731) and the CMM for Integrated Product Development (IPD-CMM).

CMMI for Development consists of best practices for development and maintenance activities applied to products and services. It includes practices that cover the product's life cycle from conception through delivery and maintenance.

CMMI appraise the process capability maturity of the software organization as follows:

- Initial
- Managed
- Defined
- Quantitatively managed
- Optimizing

Software organization has to accomplish all practices of that level as well as those of lower levels in order to reach a certain level. A process area is a summary of all requirements for a specific topic, e.g. project planning, configuration management or verification.

To satisfy a certain process area both its specific goals and generic goals have to be met.

Specific goals apply to a process area and specify the inherent characteristics that describe what has to be implemented to satisfy the process area. To meet a specific goal CMMI proposes a set of specific practices. A specific practice is an activity that is important in attaining the associated specific goal.

Generic goals are composed of the same goal statement appears in multiple process areas. Each process area has only one generic goal in the staged representation. To meet a generic goal, CMMI proposes a set of generic practices.

Generic practices must provide institutionalization to ensure that the processes associated with the process area will be effective, repeatable, and remaining [10].

## 2.2 AGILE METHODOLOGY AND XP

Agile methods have raised its position in small size software development and even large software organizations show a tendency to consult agile methods. Typical agile methods are XP (eXtreme Programming), Scrum, Crystal, DSDM (Dynamic Systems Development Model), FDD (Feature-Driven Development), Lean, agile version of RUP (Rational Unified Process) and the most popular agile methods are XP and Scrum [10], [11].

Extreme Programming (XP), which was invented by K. Beck in 1990s is the group of 12 key practices with 4 core values including communication, simplicity, courage, feedback [12], [13].

Whole Team Involvement, Planning Game, Customer Tests, Simple Design, Pair Programming, Test Driven Development, Design Improvement, Continuous Integration, Collective Code Ownership, Coding Standard, metaphor, Sustainable Pace are principles of good programming practices [14].

The life cycle of XP has five phases: Exploration, Planning, Iterations to Release, Production, Maintenance, Death.

- In the Exploration phase, the customers write the story cards and the programmers become accustomed to the technology, tools and practices they will use in the project.
- In the Planning phase, they determine the priority order of the stories and an agreement of the functions, effort, schedule of the first release is made. It takes a few days.
- The Iterations to Release phase includes several iterations. In every iteration the customer decides the stories for the iteration and the customer run the functional tests. When they finished the last iteration they had the system for final product.
- In the Production phase the customer and project team run extra testing and check the performance of the system and decide if the system can be released to the customer.
- In the Maintenance phase, project teams must keep the system in the production running while requires a new effort also for customer support.
- The Death phase is near when the changes to the architecture are no longer, design or code and documentation are completed.

There are different roles in XP for different tasks and purposes.

- Programmer writes tests and keeps the program code as simple and clear as possible.
- Customer writes the stories, sets the priority for the requirements, runs functional tests and decides which requirement is satisfied.
- Tester helps the customer write functional tests, runs functional tests regularly, informs the other members test results and maintains testing tools.
- Tracker traces the effort estimates and duration estimates, gives feedback on its accuracy, traces the progress of each iteration, evaluates the possibility of project success or if any changes are needed in the process.
- Coach has to understand all the processes and practices of XP, guide the other members to following processes.
- Consultant helps the project team in addressing their specific technical problems.
- Manager determines the current situation and distinguishes any difficulties or deficiencies in the process.

A number of researchers showed the result that XP had performed better than the other traditional development methods in the different aspects of productivity, quality, cost and schedule deviations, customer satisfaction, and team moral and so on [15]-[21]. Angela Martin showed the case study results for making clear the role of customer as the only non-developer in XP [22]. Some results have presented that aimed to creating maturity paths related to agile software development values [23].

## 2.3 A STUDY FOR COMBINATION CMMI WITH AGILE METHODOLOGY

A lot of authors have discussed the combination of CMMI with agile methods. Many works have presented that CMMI and Agile methods are not conflict, useful results could be obtained if combine them rationally with understanding of advantages and defects of both.

Some results discussed the compatibility of CMMI and agile methods or the application of specific agile methods (such as XP and Scrum) to CMMI or ISO 9001 standards [24], [25].

Kevin Scott discussed the possibility of implementation of CMMI in small size software organizations and asserted that it is possible to implement based on case studies [26].

Paul E. presented some experiences that improve software processes through the integration of CMMI and agile methods [27].

Some researchers regard that Agile methods and the SPI standards are contradict, and some think of that these SPI standards are not reasonable or no longer useful. [3], [28] But recent research results show that one can combine CMMI with Agile, and improve software development processes to any degree.

Boehm B asserted that software organizations need both agility and discipline for them to win the success [29]. And other researchers agreed with that agile methods and the SPI standards can be complemented each other in software organizations following SPI standards such as ISO 9001 and CMMI, if two approaches are combined on a proper base [29]-[31].

In the field of agile software development, two main kinds of research have been studied maturity [23]. The first kind of research focuses on adapting agile practices to current software maturity model. The second kind of research focuses on creating the path of maturity related to the values of agile software development [32], [33]. They assumed that the improvement road map for agile methods should not based on process definition because agile methods place people and interaction over processes and tools [34]. Rafaela M. F. showed empirical cases on how agile teams evolve to obtain the maturity for agile software development and presented framework to describe the agile software development maturing process [23].

# 3. THE APPLICATION OF CONTRIBUTION RATE TO IMPLEMENT PRACTICE AREAS

## 3.1 COVERAGE OF CMMI PRACTICE AREAS BY XP

Table.1. Coverage of CMMI practice areas by XP (Conflicting: –, Not addressed: 0, Partially supported: +, Supported: ++, Largely supported: +++)

| No | Practice area | XP |
|---|---|---|
| 1 | Casual Analysis and Resolution (CAR) | 0 |
| 2 | Configuration Management (CM) | +++ |
| 3 | Decision Analysis and Resolution (DAR) | - |
| 4 | Estimation (EST) | +++ |
| 5 | Governance (GOV) | - |
| 6 | Implementation Infrastructure (II) | - |
| 7 | Managing Performance and Measurement (MPM) | + |
| 8 | Monitor and Control (MC) | +++ |
| 9 | Organizational Training (OT) | ++ |
| 10 | Peer Reviews (PR) | +++ |
| 11 | Planning (PLAN) | +++ |
| 12 | Process Asset Development (PAD) | - |
| 13 | Process Management (PCM) | - |
| 14 | Process Quality Assurance (PQA) | + |
| 15 | Product Integration (PI) | +++ |
| 16 | Requirements Development and Management (RDM) | ++ |
| 17 | Risk and Opportunity (RSK) | +++ |
| 18 | Supplier Agreement Management (SAM) | 0 |
| 19 | Technical Solution (TS) | +++ |
| 20 | Verification and Validation (VV) | +++ |

The 20 practice areas can be classified into 4 categories including Execution, Management, Support and Improvement [7]. Martin Fritzsche identified in detail that some of the CMMI v1.3 process areas are supported by XP and Scrum, and some are in conflict. And he summarized the coverage of CMMI process areas by XP and Scrum [24].

So we reanalyzed practice areas of CMMI v2.0 supportable by XP for software organizations. Table.1 shows the coverage of CMMI practice areas by XP.

## 3.2 A PRINCIPLE OF COMBINATION CMMI WITH XP

As you can see in Table.1, Organizational-level Practice areas(OPAs) such as Practice area “Organizational Training”, “GOV”, “II” are not covered by XP like the other agile methods. The major cause is that these practice areas focus on organizational management, not purely related with project management. Therefore, in order to establish and execute these practice areas, organization’s management system should be improved, not change XP itself. It is important to note that the advantage of XP shouldn’t become weakened when implementing CMMI practice areas in small size software organizations.

To combine CMMI with XP, we have to establish a principle as follows: 1) Firstly, we have to decide performers for CMMI practices areas. All practice areas should be classified into organizational-level practice areas and project-level practice areas. In addition, the organizational management personnel or project management personnel have to take charge of each practice areas. There must be the performers in charge of process execution in order to establish and follow practice areas of CMMI as every work has its own master. 2) Secondly, we must not weaken the advantages of XP while establishing project management and organization management system with the combination of XP with CMMI. Because XP reflects best practices of software development, it has widely adapted in small size software development teams and has many advantages such as customer satisfaction, continuous release, quality assurance by TDD. So the combination of CMMI and XP must focus on improvement of organizational activities according to organizational practice areas of CMMI while keeping agility in the project-level activities.

## 3.3 RECLASSIFICATION OF CMMI PRACTICE AREAS AND IDENTIFICATION OF PERFORMERS

From the above mentioned principles, 20 practice areas can be divided into organizational-level practice areas and project-level practice areas.

Project-level practice areas include RSK, PR, CM, RDM, TS, PI, VV, PLAN, EST, PQA, and MC practice areas.

Organizational-level practice areas include DAR, CAR, GOV, II, OT, PCM, PAD, MPM, and SAM practice areas.

Table.2. Practice areas and Practices by its performer. (PM: Project Manager, QA: Quality Assurance Director, CEO: Chief Executive Officer, TRK: Tracker, P: Programmer, CNSLT: Consultant, TST: Tester)

| No | Practice area | Performer | Practice |
|---|---|---|---|
| 1 | CAR | PM<br>QA | 1.1, 2.1, 2.2<br>3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2, 5.1 |
| 2 | CM | P<br>PM<br>QA | 1.1, 2.5<br>2.1, 2.2, 2.3, 2.4<br>2.6 |
| 3 | DAR | PM<br>QA<br>CEO | 1.1, 1.2, 2.2, 2.3, 2.4, 2.5<br>2.1, 3.1<br>2.2, 2.3, 2.4, 2.5 |
| 4 | EST | PM<br>TRK | 1.1, 2.1, 2.2, 2.3<br>3.1, 3.2 |
| 5 | GOV | CEO | 1.1, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2, 4.1 |
| 6 | II | QA<br>CEO<br>PM | 1.1, 2.2, 3.2, 3.3<br>2.1<br>3.1, 3.3 |
| 7 | MPM | TRK<br>PM<br>QA | 1.1, 2.1, 2.2, 2.3, 2.4, 2.5, 4.1, 4.2, 4.5<br>1.2, 2.6<br>3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 4.1, 4.3, 4.4, 5.1, 5.2, 5.3 |
| 8 | MC | PM<br>TRK | 1.1, 1.2, 2.4, 3.1, 3.4<br>1.2, 2.1, 2.2, 2.3, 3.2, 3.3, 3.4 |
| 9 | OT | PM<br>CEO | 1.1, 2.1, 2.2<br>3.1, 3.2, 3.3, 3.4, 3.5, 3.6 |
| 10 | PR | P<br>CNSLT<br>QA<br>PM | 1.1, 2.2, 2.3, 2.4<br>1.1, 2.2, 2.3, 2.4<br>2.1<br>3.1 |
| 11 | PLAN | PM<br><br>QA | 1.1, 1.2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 3.1, 3.2, 3.3, 3.4, 4.1<br>4.1 |
| 12 | PAD | PM<br>P<br>QA | 1.1, 2.1, 2.2, 3.6<br>2.1, 2.2, 2.3<br>3.1, 3.2, 3.3, 3.4, 3.5, 3.7 |
| 13 | PCM | QA<br><br>CEO | 1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6<br>3.4, 3.5, 3.6 |
| 14 | PQA | TRK<br>PM<br>QA | 1.1, 2.3, 2.4<br>1.1, 2.2, 2.3<br>2.1, 2.2, 3.1 |
| 15 | PI | PM<br>TST | 1.1, 2.1, 2.2, 2.3, 2.6, 3.1,<br>2.4, 2.5, 3.2, 3.3 |
| 16 | RDM | PM<br><br>TST<br>TRK | 1.1, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6<br>2.6, 2.5, 3.7<br>2.5 |
| 17 | RSK | PM | 1.1, 2.1, 2.2, 3.1, 3.2, 3.3, 3.4, 3.5 |
| 18 | SAM | PM<br>QA | 1.1, 1.2, 2.1, 2.2, 2.3, 3.1, 3.2,<br>2.3, 4.1 |
| 19 | TS | PM | 1.1, 2.1, 2.2, 2.3, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 |
| 20 | VV | TST<br>PM | 1.1, 1.2, 2.1, 2.2, 2.3, 3.1, 3.2<br>2.1, 2.2, 2.3, 3.1, 3.2 |

The performer of organizational-level practice areas should be chief executive officer (CEO) of the organization and QA director and the performer of project-level practice areas should be project members such as project manager. It is important to note that customer take a central role in the practice areas related to requirement and testing.

The Table.2 shows the performers in charge of all practices of 20 practices areas in small size software organizations with the combination of CMMI and XP.

Some practices (e.g. 2.2, 2.3, 2.4, 2.5 in DAR) are in charge of several members. We assigned performers to practices that are not allowed in XP, so examined the possibility of introducing them in small size organizations with XP.

The Table.3 shows the result of reclassifying 20 practice areas in CMMI and the performers of each practice. Contribution rate $W_{in}$ for member $i$ in practice area $n$ is calculated as follows:

$$W_{in} = P_{in}/P_{Tn}$$

Here $P_{in}$ and $P_{Tn}$ presents the numbers of practices in charge of member i and the number of all practices in practice area n.

## 3.4 CONTRIBUTION RATES TO IMPLEMENTING PRACTICES BY PERSONNEL

Based on the identification of performers in practice areas and practices in CMMI as above, we analyzed contribution to process improvement by personnel.

The rate in charge of practice areas, $PA_i$ and the rate in charge of practices, $practice_i$ are defined as follows respectively.

$$PA_i = \sum_{n=1}^{M_T} \frac{W_{in}}{M_T}$$

$$practice_i = R_i/R_T$$

$R_i$ denote the number of practices in charge of personnel $i$. $M_T$=20 and $R_T$=193 are the number of practice areas of CMMI and the number of practices of CMMI respectively [7].

The rate of contributing to the implementing practices by each personnel, $S_i$ is defined as a mean of $PA_i$ and $practice_i$

$$S_i = \frac{PA_i + practice_i}{2}$$

The Table.4-Table.7 shows the contribution rates to implementing CMMI level 2, 3, 4 and 5 practices.

In Fig.1, we show the contribution rates to implementing the process areas by each personnel as a chart. As you can see in Table.and chart, the most of the implement and improvement of practice areas are charge of Project Manager and QA Director in the small size software organizations. So the result of

implementing and improvement of practices depends largely on Project Manager and QA Director and CEO.

In Fig.2 we present the rate of contributing to implement practices areas by CMMI level.

Table.3. Classification of CMMI Practice areas and their Performers with rate.

| No | Practice area | Level | Performer | Rate |
|---|---|---|---|---|
| 1 | CAR | Organization | QA<br>PM | 0.727<br>0.273 |
| 2 | CM | Project | PM<br>P<br>QA | 0.571<br>0.286<br>0.143 |
| 3 | DAR | Organization | PM<br>CEO<br>QA | 0.5<br>0.25<br>0.25 |
| 4 | EST | Project | PM<br>TRK | 0.667<br>0.333 |
| 5 | GOV | Organization | CEO | 1 |
| 6 | II | Organization | QA<br>PM<br>CEO | 0.583<br>0.25<br>0.167 |
| 7 | MPM | Organization | QA<br>TRK<br>PM | 0.523<br>0.386<br>0.091 |
| 8 | MC | Organization | TRK<br>PM | 0.6<br>0.4 |
| 9 | OT | Organization | CEO<br>PM | 0.667<br>0.333 |
| 10 | PR | Project | P<br>CNLST<br>QA<br>PM | 0.333<br>0.333<br>0.167<br>0.167 |
| 11 | PLAN | Project | PM<br>QA | 0.967<br>0.033 |
| 12 | PAD | Organization | QA<br>PM<br>P | 0.545<br>0.273<br>0.182 |
| 13 | PCM | Organization | QA<br>CEO | 0.864<br>0.136 |
| 14 | PQA | Organization | QA<br>PM<br>TRK | 0.417<br>0.25<br>0.333 |
| 15 | PI | Project | PM | 1 |
| 16 | RDM | Project | PM<br>TST<br>QA | 0.75<br>0.179<br>0.071 |
| 17 | RSK | | PM | 1 |
| 18 | SAM | Organization | PM<br>QA | 0.813<br>0.187 |
| 19 | TS | Project | PM | 1 |
| 20 | VV | Project | TST<br>PM | 0.643<br>0.357 |

Table.4. Contribution Rates to implementing Level 2 Practices

| | PM | QA | CEO | TRK | P | CNSLT | TST |
|---|---|---|---|---|---|---|---|
| $PA_i$ | 0.5619 | 0.1312 | 0.081 | 0.0867 | 0.0593 | 0.02 | 0.06 |
| $Practice_i$ | 0.5566 | 0.1179 | 0.0754 | 0.1085 | 0.0566 | 0.0189 | 0.066 |
| $S_i$ | 0.5593 | 0.1246 | 0.0782 | 0.0976 | 0.058 | 0.0194 | 0.063 |

Table.5. Contribution Rates to implementing Level 3 Practices

| | PM | QA | CEO | TRK | P | CNSLT | TST |
|---|---|---|---|---|---|---|---|
| $PA_i$ | 0.4772 | 0.2065 | 0.111 | 0.0848 | 0.04 | 0.0167 | 0.064 |
| $Practice_i$ | 0.4972 | 0.2095 | 0.0978 | 0.0893 | 0.0335 | 0.0112 | 0.0615 |
| $S_i$ | 0.4872 | 0.208 | 0.1044 | 0.087 | 0.058 | 0.0139 | 0.0627 |

Table.6. Contribution Rates to implementing Level 4 Practices

| | PM | QA | CEO | TRK | P | CNSLT | TST |
|---|---|---|---|---|---|---|---|
| $PA_i$ | 0.4641 | 0.2186 | 0.111 | 0.0857 | 0.04 | 0.0167 | 0.064 |
| $Practice_i$ | 0.4735 | 0.2302 | 0.0979 | 0.0979 | 0.0317 | 0.0106 | 0.0582 |
| $S_i$ | 0.4688 | 0.2244 | 0.1044 | 0.0918 | 0.0358 | 0.0136 | 0.0611 |

Table.7. Contribution Rates to implementing Level 5 Practices

| | PM | QA | CEO | TRK | P | CNSLT | TST |
|---|---|---|---|---|---|---|---|
| $PA_i$ | 0.4849 | 0.2237 | 0.111 | 0.0826 | 0.0401 | 0.0167 | 0.0411 |
| $practice_i$ | 0.4845 | 0.2461 | 0.0959 | 0.0959 | 0.0311 | 0.0104 | 0.0363 |
| $S_i$ | 0.4847 | 0.2349 | 0.1034 | 0.0892 | 0.0356 | 0.0135 | 0.0387 |

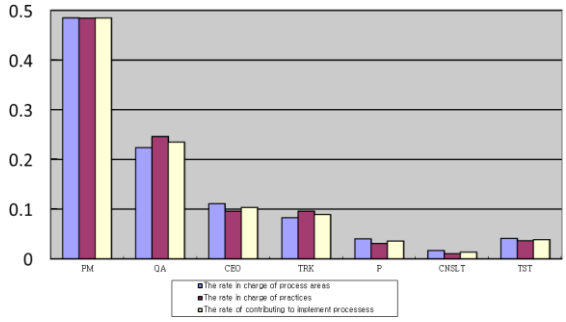


Fig.1. The rate of contributing to implement practice areas

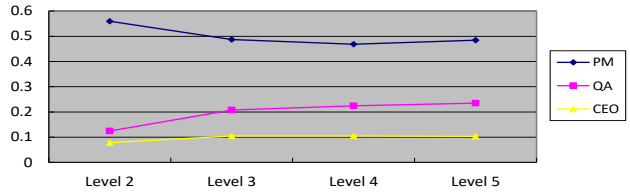


Fig.2. The rate of contributing to implement practice areas by CMMI level

## 4. CASE STUDY RESULTS

Our new solution is applied to practice implementation of two IT organizations in DPR of Korea.

Organization A has been used XP methods and has 36 developers and 4 management personnel. Organization B also has been used XP methods and has 72 developers and 5 management personnel.

In these organizations each project has 5~8 developers. So these organizations could be viewed as typical small-size software organizations in the domestic software industry.

Table.8 shows the field, average number of developers, duration of projects in two organizations.

Table.8. Summary of the case organizations

| Organization | Field of project applied new model | Developers per project (average) | Project duration (months) |
|---|---|---|---|
| Org. A | Android application<br>Web application<br>Artificial intelligence | 5<br>4<br>4 | 10<br>7<br>12 |
| Org. B | Multimedia communication<br>Android application<br>Network communication | 6<br>4<br>6 | 8<br>6<br>8 |

Table.9. A Period of achievement of CMMI per Level

| Org. | CMMI Level | | |
|---|---|---|---|
| | Level 2 | Level 3 | Level 4 |
| Org. A | 10 | 25 | 30 |
| Org. B | 8 | 20 | 32 |

Table.10. Comparison of Period of achievement of CMMI level 4.

| Solution | CMMI Level | Mean of Period (months) |
|---|---|---|
| Proposal | 4 | 31 |
| [1] | 4 | 36 |

Organization A had not launched software process improvement based on CMMI yet. Organization B had introduced the combination CMMI with XP a year ago, but they were not confident of them. Above all, they were troubled by deciding responsibilities and roles of each member in their organization.

Both of these organizations welcomed our solution for assigning performers to each practice to combine CMMI with XP and they started implementing CMMI practice areas by introducing our solution.

The main obstacle in defining practice areas and assigning performers to each practice is the incomprehension of project managers. They were displeased with CMMI practices.

In Org. A we advised CEO to appoint QA to project member in a trial project. QA with experience as project manager took a role of project manager to demonstrate the effectiveness of the combination with XP. In Org. B we convinced project managers of the CMMI certification's value so they agreed to introduce our solution.

Since both organizations introduced our solution they have successfully established CMMI practice areas while keeping XP methods as before and they reached level 4 in 30 months and in 32 months respectively.

The Table.9 shows a period of achievement of CMMI by level. Our novel solution and [1] are compared in the aspect of period of achievement of CMMI level 4. The Table.10 shows that our novel solution is capable of saving the period and effort for reaching CMMI level 4 in small size organization.

As you can see above table, our novel solution for combining CMMI practice areas with XP methods needs less period of achievement of CMMI level 4 in small size software organizations.

## 5. CONCLUSIONS

We presented the solution for identifying performers of CMMI practice areas and practices in XP-based agile teams and small software organizations and applied our proposal to 2 small size software organizations which had different experiences and histories of process management. The case studies show that our solution is more effective for small software organizations and helps them to reach CMMI level 4 with less period.

In the future, we should make clear the responsibilities and roles in each practice areas, its effect on the structure of organization in more details.

## REFERENCES

[1] K.S. Ryang and I.N. Mun, "Software Process Improvement by Changing Roles in Small Size Organizations", *International Journal of Software Engineering*, Vol. 10, No. 2, pp. 3-20, 2017.

[2] H. Andre, "Agile Project Management: A Case Study on Agile Practices", Master's Thesis, Department of Business Administration, Arctic University of Norway, pp. 1-86, 2016.
[3] T. Stalhane and G.K. Hanssen, "The Application of ISO 9001 to Agile Software Development", *Proceedings of 9th International Conference on Product-Focused Software Process Improvement*, pp. 371-385, 2008.
[4] S. Shafiq and N.M. Minhas, "Integrating Formal Methods in XP–A Conceptual Solution", *Journal of Software Engineering and Applications*, Vol. 7, No. 4, pp. 299-310, 2014.
[5] S. Stavru, "A Critical Examination of Recent Industrial Surveys on Agile Method Usage", *Journal of Systems and Software*, Vol. 94, pp. 87-97, 2014.
[6] S.V. Shrivastava and U. Rathod, "Categorization of Risk Factors for Distributed Agile Projects", *Information and Software Technology*, Vol. 58, pp. 373-387, 2015.
[7] CMMI Product Team, "CMMI-DEV V2.0", Software Engineering Institute (SEI), Carnegie Mellon University, 2010.
[8] K.E. Emam and W. Melo, "*SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*", IEEE Computer Society Press, 1997.
[9] ISO 9001:2000, Quality anagement Systems-Requirements, 3rd Edition, Pearson Press, 2000.
[10] A. Raninen, "*Practical Process Improvement: How to Initiate Software Process Improvement in a Small Company*", Ph.D. Dissertation, Department of Forestry and Natural Sciences, University of Eastern Finland, pp. 1-150, 2014.
[11] C. Konstantinos, "Measuring Agility: A Validity Study on Tools Measuring the Agility Level of Software Development Teams", Master of Science Thesis, Department of Software Engineering, University of Gothenburg, pp. 1-98, 2015.
[12] S.V. Shrivastava and U. Rathod, "Categorization of Risk Factors for Distributed Agile Projects", *Information and Software Technology*, Vol. 58, pp. 373-387, 2015.
[13] K. Beck and C. Andres, "*Extreme Programming Explained*", Addison-Wesley Publisher, 2004.
[14] A. Jackson, S.L. Tsang, A. Gray, C. Driver and S. Clarke, "Behind the Rules: XP Experiences", *Proceedings of International Conference on Agile Development*, pp. 87-94, 2004.
[15] P. Abrahamsson and J. Ronkainen, "*Agile Software Development Methods, Review and Analysis*", VTT Publications, 2002.
[16] K. Beck, "*Test-Driven Development*", Addison-Wesley Publisher, 2002.
[17] Extreme Programming: A Gentle Introduction, Available at: http://www.extremeprogramming.org/, Accessed at 2013.
[18] A. Tarhan and S.G. Yilmaz, "Systematic Analyses and Comparison of Development Performance and Product Quality of Incremental Process and Agile Process", *Information and Software Technology*, Vol. 56, No. 5, pp. 477-494, 2014.
[19] F. Macias, M. Holcombe and M. Gheorghe, "A Formal Experiment Comparing Extreme Programming with Traditional Software Construction", *Proceedings of Mexican International Conference on Computer Science*, pp.73-80, 2003.
[20] S. Ilieva, P. Ivanov and E. Stefanova, "Analyses of an Agile Methodology Implementation", *Proceedings of International Conference on Euromicro*, pp.326-333, 2004.
[21] L. Layman, L. Williams and L. Cunningham, "Exploring Extreme Programming in Context: an Industrial Case Study", *Proceedings of International Conference on Agile Development*, pp. 32-41, 2004.
[22] A. Martin, R. Biddle and J. Noble, "The XP Customer Role in Practice: Three Studies", *Proceedings of International Conference on Agile Development*, pp. 42-54, 2004.
[23] R.M. Fontana, V.J. Meyer, S. Reinehr and A. Malucelli, "Progressive Outcomes: a Framework for Maturing in Agile Software Development", *Journal of Systems and Software*, Vol. 102, pp. 88-108, 2015.
[24] M. Fritzsche and P. Keil, "Agile Methods and CMMI: Compatibility or Conflict?", *e-Informatica Software Engineering Journal*, Vol. 1, No. 1, pp. 9-26, 2007.
[25] C. Santana, C. Gusmao, L. Soares, C. Pinheiro, T. Maciel, A. Vasconcelos and A. Rouiller, "Agile Software Development and CMMI: What We Do Not Know about Dancing with Elephants", *Proceedings of International Conference on Agile Processes in Software Engineering and Extreme Programming*, pp. 124-129, 2009.
[26] K. Scott, "Capability Maturity Model Integration (CMMI) for Small Organizations", Master Thesis, Department of Management Studies, Regis University, pp. 1-90, 2019.
[27] P.E. McMahon, "*Integrating CMMI and Agile Development*", Addison-Wesley Publisher, 2010.
[28] R. Turner and A. Jain, "Agile Meets CMMI: Culture Clash or Common Cause?", Springer, 2002.
[29] B. Boehm and R. Turner, "*Balancing Agility and Discipline: A Guide for the Perplexed*", Addison Wesley Publisher, 2003.
[30] C.R. Jakobsen and J. Sutherland, "Scrum and CMMI-Going from Good to Great Are You Ready-Ready to Be Done-Done?", *Proceedings of International Conference on Agile Model in Software Engineering*, pp. 333-337, 2009.
[31] G. Lami and F. Falcini, "Is ISO/IEC 15504 Applicable to Agile Methods?", *Proceedings of International Conference on Agile Processes in Software Engineering and Extreme Programming*, pp. 130-135, 2009.
[32] M. Leppanen, "*A Comparative Analysis of Agile Maturity Models*", Springer, 2013.
[33] T. Schweigert, R. Nevalainen, D. Vohwinkel, M. Korsaa and M. Biro, "Agile Maturity Model: Oxymoron or the Next Level of Understanding", *Proceedings of International Conference on Software Process Improvement and Capability Determination*, pp. 289-294, 2012.
[34] R. M. Fontana, I. M. Fontana, P. A. R. Garbuio, S. Reinehr and A. Malucelli, "Processes Versus People: How Should Agile Software Development Maturity Be Defined?", *Journal of Systems and Software*, Vol. 97, pp. 140-155, 2014.