***I.J. Information Technology and Computer Science*, 2018, 5, 55-67**
Published Online May 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2018.05.06

Product Focused Software Process Improvement through Integrated Framework of Agile and CMMI: A Case in Small Settings

Tatek Engdashet Kabitimer
IT Doctoral Program, Addis Ababa University, Addis Ababa, Ethiopia
E-mail: mail4te@gmail.com

Dida Midekso
Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia
E-mail: dida.midekso@aau.edu.et

Ricardo J. Machado
Department of Information Systems, University of Minho, Guimar ães, Portugal
E-mail: rmach@dsi.uminho.pt

Received: 11 April 2017; Accepted: 19 March 2018; Published: 08 May 2018

Abstract**—**Software process improvement (SPI) is an important requirement in a software company. The search for better approach brought different kinds of models with multiple sets of principles for SPI to be founded. The framework is proposed to mainly address an alternative way of achieving a better process capability. The approach focuses on the implementation of SPI which can seamlessly align with the organization nature, day to day business activities, and financial capability. The paper provides the detailed implementation guideline and application of the framework through case study results.

The case study is performed in a software development unit placed under academic institution. The unit is founded specifically for application development for internal and external customers. The case study is designed to be implemented in two software development projects in the development unit. From the ongoing case study, the results from the first project which is completed in six iterations are presented in this paper. Considering SPI implementation, the development team followed the framework and associated procedures throughout the development process. The results obtained in terms of aligning SPI to the daily development task and CMMI KPAs capability improvement achieved showed promising results.

Index Terms**—**Integrated framework of agile and CMMI, institutionalization, agile development, product-focused software process improvement.

I. Introduction

The quality of a software product has been referred to be highly related to the output of the software process used in the development process. To address the issue, the software engineering research brought implementation and evaluation frameworks for process maturity. Software development companies follow varieties of maturity models to improve the software process to maintain and upgrade the quality of software developed [1].

Software Process Improvement (SPI) has been implemented in software development companies based on different frameworks. The primary objective is upgrading software process capability which is one of the main requirements for better product quality. It is generally accepted that software process improvement is the crucial factor for the quality of the software product development. The search for SPI strategy has brought many dimensions of observation into the situation; as a result, many important approaches have been made. In general, considering the context of the companies before implementing the SPI program is more emphasized by the researchers in the field [1], [2], [3]. In line with this, a literature review prevailed a number of models developed to be used in different contexts. The implementation of the models prevailed varieties of factors influencing the success of SPI implementation [4, 5].

In other research, Niazi et. al [6] identified nine wide-ranging factors from two data sets that are generally considered critical for successful implementation of SPI [6]. These factors include; allocation of resources, awareness, creating process action teams, defined SPI implementation methodology, experienced staff, and support from higher management, staff involvement, training, and reviews. Among the nine factors mentioned, SPI awareness and defined SPI implementation methodology were not identified in the literature. Rather, these factors have been considered important by practitioners participated in the survey [6].

The implementation of SPI programs differs according to the priority of parameters considered for a particular context. Studying the initiatives and their success stories is a valuable input to study and evaluate alternative approaches to develop cost-effective, simple, and context-aware SPI framework. A review made by Wangenheim et al. [7] showed 52 different software process capability maturity models. According to the research, the models are developed focusing on customizing SPI to different contexts as the main focus. The contexts such as suitability for small and medium enterprises, testing and quality assurance and the like are presented [7]. The sources for most of the models were CMM, CMMI-DEV, and ISO.

Based on the review, the implication of developing varieties of models is mainly not related to what a matured process should possess, but how these properties can be achieved. The search is mainly a way of achieving a better process capability which can seamlessly align with the organization nature, day to day business activities and financial capability. As a result, a reference model is developed by Christiane et al. [8] to be used as a guide in crafting software process capability maturity model. The model developed contains five phases and seventeen steps to be followed for successful development of software process capability maturity model.

According to the research and experience reports, the well-known standards, like CMMI and ISO, have been dominant for a number of years [4, 7, 9, 10]. The standard models brought a significant contribution to the software industry in terms of upgrading software process maturity. Following the well-known SPI frameworks for the implementation of process improvements has an advantage in terms of ensuring their result to achieve the intended goal. This is mostly related to the reason that, their applicability and impact in similar purpose, is relatively proved. Despite the success stories, problems associated with the difficulty of their implementation in small and medium scaled companies is reported [9], [10]. The main threat for such frameworks is difficulty in implementation due to the need of a separate financial, time, and skill requirement which has a critical impact on small companies. Hence, implementing the well-known standards like CMMI in such companies create implementation difficulties [11], [12]. The requirements stipulated by the standard CMMI model are not affordable by small and medium companies. This has made the SPI in those companies difficult, ineffective and unusable [13]. The challenges also include the difficulty of aligning SPI implementation with the daily development activities, which can reduce its effectiveness [14].

On the other hand, the recently introduced agile methods got the attention in the software industry as an alternative way of software development approach. Agile methods suitability is closer to small and medium companies. Even though the approach claim practitioners reports of better quality software development, some difficulties have also been reported. The main limitation associated related to long term and formal process improvement [15]. The current research focus brought the issue of combining practices from agile and CMMI [16], [17]. The approach focuses on their similarities and some identified complementary relationships.

The process of combining agile practices with CMMI has been tried and encouraging results have been achieved [18], [19]. The approach taken to assess agile methods using CMMI has shown that the practices can work together in selected process areas of CMMI and agile methods [20]. The research attempts made in the area mainly focus on how to introduce agile practices to the already existing plan driven environments implementing CMMI and other industry standards. Some attempts also have been made using CMMI as an assessment tool for agile practices [20]. The researches done so far have demonstrated the possibilities and advantages associated with the collaboration of CMMI and agile methods [21], [22]. However, a unified framework which incorporates the two approaches of software development with the defined role of the practices is not presented yet. Such a framework can be an alternative path for software companies to formalize and upgrade their process maturity based on industry standards. The approach focuses on how to retain knowledge and experience by targeting problems related to limited resource and resistance to change. It is expected to have a significant advantage in sketching the road map for SPI by retaining the advantage of both agile methods and CMMI framework standards.

To address the issue mentioned above, this research attempt to develop an SPI framework of agile and CMMI. The first step is determining complementary relationships between practices of the agile methods and specific practices of CMMI KPAs. As a result, mapping agile practices with specific practices of CMMI level 2 and 3 KPAs is completed in the first phase of the research. In the next step, "best fit" approach to design an integrated framework using agile methods and CMMI is proposed. The approach implements institutionalization process flow to align capability levels of CMMI continuous representation with agile practices. In this paper, a case study discussing the implications of using an integrated framework of agile and CMMI for software process improvement is discussed. The next part discusses related work in combined implementation of agile and CMMI. In the next part presents the framework and detailed features of implementation guidelines. It followed by research methodology selected, study design, and data collection strategy. Finally, discussion of the case study process with observed results and their interpretation is presented.

II. Related Works

Combining the two approaches has taken the attention of researchers recently. Accordingly, research findings related to combining some agile methodologies with the KPAs in the CMMI framework have demonstrated the two approaches can work together and can also be even better if they are implemented thoughtfully than they are individually [9, 11, 12, 19]. In line with this, the attempts made and the results obtained have shown tangible evidence of achievements in software development and SPI through combining CMMI and agile methods. Experience reports from Wake [23] shown that agile practices are accelerators of SPI with a benefit of providing a quality product with time and improve the capability of the organization. Performance improvement has been achieved in both small and large projects as a result of using scrum with CMMI. The combination of Agile practices and CMMI is explained as a means to "Amplify Learning and Deliver Fast" [23]. The following summary of related works demonstrates the approaches considered, methodologies used, and findings reported by the researches.

According to the research from M. Yousef et al. [17], it is claimed to be possible to cover twenty (twelve largely and eight partial) out of twenty-two of CMMI level 2 to 5 KPAs through XP. The research presented description of the relationship between CMMI and XP at the KPA level, but it did not provide any objective evidence. Other researches argued that, such coverage is impossible to attain in the current form of the two sets of methods [24, 25]. In a similar context, a reference model called a "CMMI-Scrum (C-S)" model is developed by Miler et. al [19]. The model used for mapping specific goals of the second and the third level of maturity in the CMMI staged representation of the CMMI (V 1.2) model onto the activities described by the Scrum methodology. The study considers Scrum to cover some practices of level 2 and 3 of staged representation of CMMI and claim 40% coverage of specific practices. The objective of the approach is to manage the compromise between scrum agility and CMMI maturity through selection of practices and introducing new ones. The research did not demonstrate the process whereby specific practices or maturity levels improve their capability. The approach is limited to, application of scrum in a CMMI environment and identify problems associated with a project than an organized capability improvement plan.

Introducing agile methods to a plan driven environment is also an approach followed in combining agile and CMMI. A research from Sofia et al. [21], show introduction of scrum practices to the CMMI practices implemented environment. Introducing agility to a plan driven environment is one of the approaches demonstrated by researchers. The approach taken was focused on aligning practices from scrum to CMMI project management process areas with assessment of their relationships and differences. The process areas other than project management were not considered. The research concluded that, an improvement on the relationship level between CMMI project management process areas and scrum can be attained through tailoring scrum practices. From another perspective, the combined implementation of agile and CMMI is presented through application of CMMI framework for assessment of agile software development. Such approach is demonstrated by Pikkarainen et. al. [20]. The approach defines a model and associated description of the guidelines for process assessment. The research defined relationship between agile practices and CMMI specific goals as a main component. Its main focus is, on assessment aspect, than providing set of activities with defined target for process improvement. Baker [18] demonstrated an approach to introduce CMMI to agile contexts with the objective of improving stability of the organization’s process while keeping the desired agility in product development. Three companies with different development approach, implemented CMMI and certified to some level were used. The approach first examines the existing development approach and accordingly tailor the development through the introduction of scrum practices. The detailed guideline and common approach to implementation of CMMI and agile method is not the focus. Rather, the approach focused on defining a framework where different development entities can introduce scrum to their development culture.

In other researches, combined implementation of scrum and XP from agile with the CMMI approach was considered as an approach to SPI. A research from Fritzsche et. al [22], presented analysis of all CMMI capability levels with practices from XP and scrum. The detailed analysis presented the compatibility, collaboration and conflict between level 2 to level 5 KPAs from CMMI and practices of XP and scrum. The research concluded that, the agile method (XP and scrum) support CMMI level 2 and 3 KPAs. To fully utilize the combination of the relationship between these two sets of practices, a practice catalogue of agile practices is proposed to be studied and developed. The researchers further recommended analysis of other agile methods regarding their relationship with CMMI. The research didn’t define a guideline or approach to be followed to implement SPI through a combination of practices from the two approaches.

Since the idea is introduced recently, and is coming out of the idea that “agile and CMMI are completely incompatible set of practices”, it needs a thorough investigation in all dimensions to enrich the results achieved so far. The approaches taken mostly are fragmented approach by picking practices from both approaches and considering their complementation to benefit from. One of the areas which is not addressed in detail is the integration of the two paradigms as a unified framework to be used as an alternative path for companies starting product focused SPI implementation. Integrating the practices in a unified model can provide an excellent opportunity, especially software companies with a low maturity level, benefit from their process improvement in parallel with the daily development activity

III. Integrated Process Improvement Framework

Implementing an SPI framework mediated between agile and CMMI is considered one of the possible approaches to address the limitations reported of SPI implementation in different contexts. Focusing on closing the gap between SPI activity and daily development activities, this research developed an integrated SPI framework of agile and CMMI. The first step in this regard is, determining complementary relationships between practices of the agile methods and specific practices of CMMI KPAs. In the next step, "best fit" approach to design an integrated framework using agile methods and CMMI is proposed. The framework is composed of three components. The integrated capability improvement process flow, which is the first component, is defined based on the concept of institutionalization of practices of agile methods on the continuous representation of CMMI framework. The SPI tracking model is the second component of the framework which is used to record and track capability level of CMMI KPAs. The third component of the framework which is the Post Iteration and Process Improvement workshop (PIPIW) define the process steps to be followed in the implementation of the SPI activities.

The first step of developing the framework execute analysis of the relationship between CMMI KPAs and practices from agile methods. The analysis and mapping are done based on definitions of the concepts represented by the practices, case studies, and experiments reported by different researchers. The relationship between the KPAs in CMMI level 2 and 3 with the specific practices is covered in this research. Each specific practice is mapped with the corresponding agile practice which can address it. The definition of each specific practice of the KPAs is directly taken from CMMI v1.3 to keep the context of the specific practices as presented by the CMMI framework. In general 112 specific practices from the 14 KPAs in level 2 and 3 of CMMI are mapped with agile practices. In this regard, some specific practices are found to be not addressed by an agile practice due to lack of proven evidence for the relationship. Apart from the mapping between the practices, the analysis didn't specify the synergistic relationship between the two practices as a result of the implementation of both. The synergy which can be obtained from this effect could be vague to quantify and present with explicit mapping. Table 1 show part of this mapping considering Project Monitoring and Control (PMC) KPA. The mapping is done for all specific practices in each KPAs of CMMI level 2 and 3, with related practices from agile methods.

Table 1. Sample relationship matrix of CMMI KPAs specific practices and practices of agile methods

		Specific Practice	Definition	XP practice	Scrum practice
KPA 3. Project Monitoring and Control (PMC)	SG 1 Monitor the Project Against the Plan	SP 1.1 Monitor Project Planning Parameters	Monitor actual values of project planning parameters against the project plan	Onsite customer	Daily sprint meeting
		SP 1.2 Monitor Commitments	Monitor commitments against those identified in the project plan.	Pair programming	Daily sprint meeting, Sprint review meeting
		SP 1.3 Monitor Project Risks	Monitor risks against those identified in the project plan.	Continuous integration, small release	Sprint planning meeting
		SP 1.4 Monitor Data Management	Monitor the management of project data against the project plan.		Daily sprint meeting, Sprint review meeting
		SP 1.5 Monitor Stakeholder Involvement	Monitor stakeholder involvement against the project plan.	Onsite customer	Sprint planning meeting
		SP 1.6 Conduct Progress Reviews	Periodically review the project's progress, performance, and issues.		Daily sprint meeting, Sprint review meeting
		SP 1.7 Conduct Milestone Reviews	Review the project's accomplishments and results at selected project milestones.	Small release	Sprint review meeting
	SG 2 Manage Corrective Action to Closure	SP 2.1 Analyze Issues	Collect and analyze issues and determine corrective actions to address them		Daily sprint meeting, Sprint review meeting
		SP 2.2 Take Corrective Action	Take corrective action on identified issues.		Sprint planning meeting
		SP 2.3 Manage Corrective Actions	Manage corrective actions to closure.		Sprint review meeting

Following the mapping of the practices, a framework which is used to define the guideline for process improvement is defined. The integrated framework of agile and CMMI for software process improvement is defined based on the concept of institutionalization. According to Software process improvement is the institutionalization of practices in a predefined structure, through identification of KPAs and associated specific practices based on the organization context. The approach proposes to institutionalize practices of agile methods in a based on the CMMI framework of software process improvement. The detailed description of the framework is covered in an earlier publication of the first part of the research [26]. Among the CMMI approaches of SPI, the CMMI v1.3 continuous representation of process improvement is used. It is selected due to its flexibility in selecting any KPA at any instance of software development activity [27]. The approach is expected to make the software process improvement more product focused. It is expected to enable companies to execute process improvement with minimal resource allocation and without losing focus on the daily software development activity. Capability improvement of selected KPAs follows the institutionalization process defined in the institutionalization theory [28]

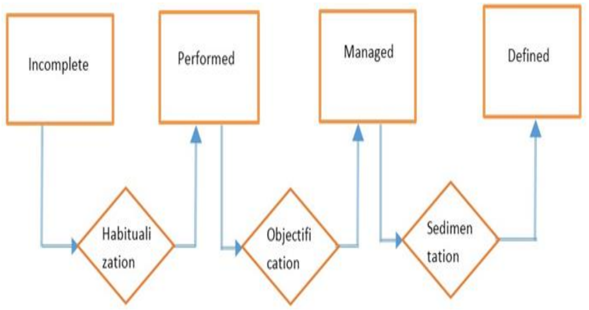


Fig.1. Integrated capability improvement process flow [26]

Habitualization: - Deals with the creation of new structural arrangements in response to a specific organizational problem or set of problems. Based on the definition of institutionalization, the concept is contextualized to a software development structure. In this regard, an agile practice performed in a particular problem-solving process is evaluated. The evaluation is on how consistently the practices selected, procedures followed, and templates used in a software development process for similar tasks.

Objectification: - Implies the diffusion of a structure. It involves the development of some level of recognition by decision-makers and adoption of a structure and the increasing adoption by organizations. According to Wiseman, a structure at this level of institutionalization is considered as trustworthy, useful and reliable by adopters of the structure [29]. Based on the definition of institutionalization the concept and requirement are contextualized to a software development structure. Accordingly, a development group can identify patterns of activities to be executed for a particular software development activity by a certain development group from the outset. When this pattern match with the activity performed by the development group then the set of practices can be considered as compliant with all requirement of managed process.

Sedimentation: - Refers full institutionalization of a structure and continuity of the structure throughout the context where the structure is sited and its existence over a period of time [28]. Contextualized to the requirement of SPI, those practices and associated procedures need to be accumulated through a rigorous application and reputation similarity of application. Hence, in SPI implementation of continues representation (CMMI v1.3), sedimentation level refers to defined practices of KPA. This is a level where an organization's standard and procedure related to the practices have been made, and any procedure from the standards can be tailored to a specific approach. The tailoring process follows the predefined procedure.

As it is discussed earlier, the capability improvement of specific practices of CMMI KPAs implemented through their level of institutionalization. In line with this, the capability improvement of a KPA is evaluated through the capability improvement of its specific practices. Hence, the level of institutionalization will be determined by the capability levels defined. Improving capability of specific practices can be implemented through institutionalization of the agile practices. The tracking model developed and discussed in detail in the earlier phase of this research used to track and display the capability level of a given KPA [26]. The graph is designed to visually demonstrate the status of KPAs through their corresponding specific practices. Hence, the status of specific practices of each KPA.

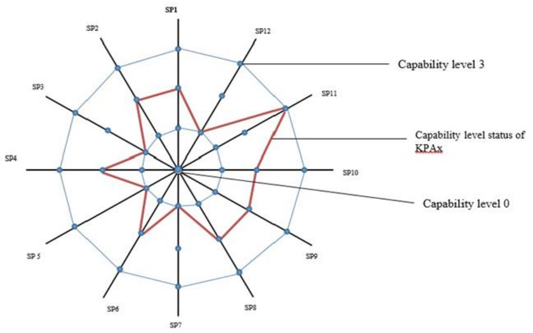


Fig 2. SPI tracking model [26].

The steps used for analysis and detailed implementation procedures of the framework is defined based on the process flow defined in post iteration and process improvement workshop (PIPIW). The process flow is crafted based on the activities of the post-iteration workshop. The detailed activities in the process flow are presented in the next section.

A. The post-Iteration and Process Improvement Workshop (PIPIW)

In the agile development process, post iteration workshop (PIW) provides project teams a way to examine and accordingly shape the practices while running projects [30]. Moreover, it makes easy to get a quick feedback on the improved practices. The steps used in the post-iteration workshop by Salo et al. is used as a baseline to develop the PIPIW. Based on the phases of PIW, the PIPIW include process improvement steps and guidelines from the integrated framework (Fig 1.), and SPI tracking model (Fig 2.). The PIPIW is presented in the process flow shown in Fig 3. and the detailed description of each component indicating their role in the PIPIW can be measured and tracked through the frameworks and associated practices included. The sequence of activities is shown in Fig 3. and the detailed description of each indicating their role in the PIPIW.

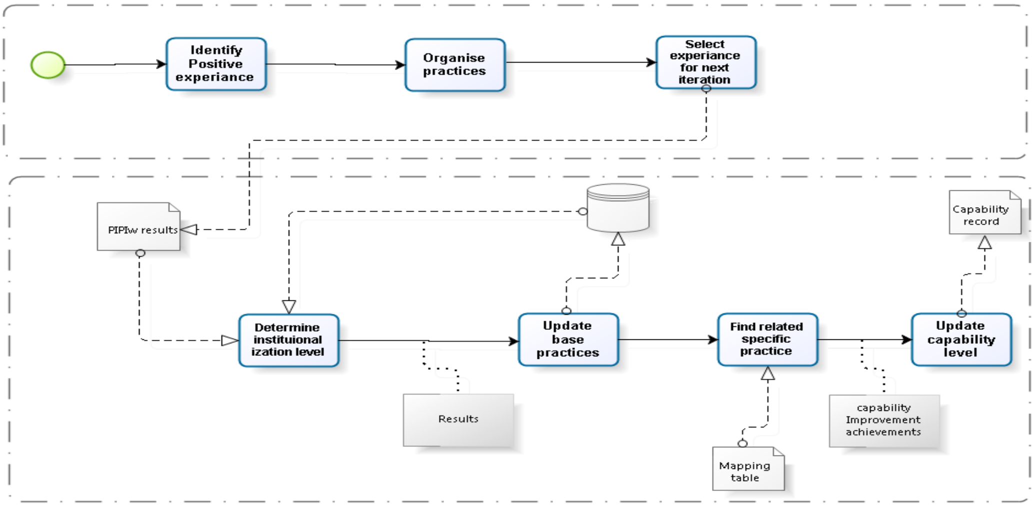

Identify Positive experiance
Organise practices
Select experiance for next iteration
PIPIw results
Capability record
Determine instituional ization level
Update base practices
Find related specific practice
Update capability level
Results
capability Improvement achievements
Mapping table

Fig.3. Post Iteration & Process Improvement Workshop (PIPIW) process steps

Identify and organize positive experiences: - Project participants identify the practices with positive outcomes. The fundamental activities are based on the practices of agile methods (XP and Scrum). The project participants determine the positive experiences in that specific iteration and explanation of the approach on how positively affect performance. This followed by organization of the practices with the associated procedures and templates (if any). Group discussion is held within the group members reviewing and creating common consensus on the practices identified and organized. In line with this, the group discussion can also review the negative experiences, then the next iteration planning can avoid those practices from happening.

Define activities for next iteration: - Based on the collected positive experiences identified in the iteration undertaken by the team and the discussion made, the group members define the activities for the upcoming iteration. The activities are defined in terms of procedures to be followed for future similar tasks. The list of practices is recorded on a storyboard for each iteration to enable comparisons to be made in each iteration to trace the improvement achieved. The different teams can see the practices with their procedures and templates selected and refined in the iterations carried out by the team.

Analysis of similarity: - In this step, the improvement group collects the predefined set of activities defined by the development groups and evaluate their level of similarity. The base practices compiled by the company are used as a benchmark for the analysis and will get updates at the end of each iteration. The resulting output is used to determine the status of institutionalization of those practices in that particular setting. The Integrated capability improvement process flow is used to guide the evaluation of the different levels of institutionalization of specific practices. The level of institutionalization of the practices is defined according to the requirements specified at each level of institutionalization process components.

Mapping table reference: - The relationship matrix of CMMI and agile practices is used to map the practices. It is used to identify specific practices of CMMI KPA addressed by the agile practices identified in the previous steps. An agile practice can be mapped to different specific practices and a single practice can be addressed by more than one agile practice. In the former case, all the specific practices are considered as addressed by the corresponding agile practices. In the latter case, all the agile practices should be addressed to improve the capability level of the corresponding specific practice.

Indicating progress on the SPI tracking model: - Basically, the SPI tracking model is used to follow up and display the improvement progress of KPAs through its specific practices. In the previous steps, the agile practices are evaluated based on their level of institutionalization. It follows with the determination of specific practices of KPAs they can be mapped with. In the SPI tracking model, the capability level of specific practice is updated based on the level of institutionalization recorded in earlier steps. The complete capability level of the KPA is represented through the capability level of its specific practice. Following the PIPIW the capability improvement of a particular KPA can be achieved through the improvement of capability level of its associated specific practices. This can show where the gap for improvement is, and to consider them in the upcoming capacitation plan of the organization.

IV. Research Approach and Methodology

Case study research is selected for the study due to a number of reasons related to the research context. According to Yin, a case study is a suitable approach to investigate a contemporary phenomenon in a real life context. It is a good approach to answering questions related to how and why, which are related to operational links to be traced over time [31]. In software process improvement when studying a change as a result of introducing an approach case study is the case study is the suitable approach to follow [32] As pointed out by Runeson [33] "Case study research lends itself naturally to software process improvement (SPI) because of the focus of case studies on individual sites within their natural context". The study followed case study steps iteratively based on the guides in case study research described by Benbasat. These are preparing data collection, collecting evidence, analyzing case study evidence, and presenting case studies [34]. The framework is applied in a software development environment. In this process, the researchers' role was as a participant observer and on some occasions lead meetings with a group discussion.

The research targets, investigating and examining the integrated SPI framework of agile and CMMI. The evaluation is done based on the improvement of the capability of specific practices in CMMI KPAs using individual agile practices. Therefore, the unit of analysis of the case study is the capability improvement of CMMI KPA specific practices through agile practices. The capability determination follows the requirements specified at each level of process components of institutionalization

A. Data Collection and Analysis

According to Benbasat, good quality case study research considers three basic principles. These are the use of multiple sources of evidence, creating a case study database, and maintaining the chain of evidence. It is generally recommended to use six data sources in conducting case study research. These are documentation, archival records, interviews, direct observations, participant observations, and physical artifacts [31]. The research used structured and semi-structured individual and group interviews, case project teams' software development documentations, direct observations, and discussion workshops according to the scenario to be investigated.

The PIPIW technique is used to guide the collection of data through evaluation of which practice's template and procedure have been consistently implemented. The technique is used to review activities of the previous iteration of the project. In this research, the outcomes of the post-iteration workshop are used to measure the level of institutionalization of agile practices. The analysis is done on software development project being developed by the development group considered for the case study. All relevant documents and the base practices are organized and kept in the development repository.

In addition to the data collection techniques mentioned, agile project planning and tracking tool called JIRA agile is used as one method of collecting data. In addition, the tool is used as communication means among group members while performing case projects. The tool used to organize, allocate, schedule and track tasks as they proliferate. The development team used the tool to put stories, decompose stories into tasks, and assignment of tasks to developers.

B. Research Context and Case Description

According to Yin, the case can be organized as a holistic case or embedded case [31]. The former is recommended in contexts where within a single case, attention is also given to a subunit or subunits. On the other hand embedded case is considered suitable where the case study examined only the global nature of an organization or of a program. The context of this research is a software development environment. Hence the different software development projects can be considered as units of analysis in a software development environment. Accordingly, embedded case is considered more relevant to follow in the research context considered for this study.

In the case study setting, where the candidate companies selected for the case are fully engaged in software development. The researcher made subsequent presentations to a number of companies about the basics and the objectives of the SPI framework. Following the full acceptance of the companies, one has been selected for the case. The timing to conduct the study in the shortest time possible and availability of the development professionals to be fully engaged in the project has been the main factor to select the company. The company where the case study hosted is totally voluntary and agreed to implement the approach in the company. Accordingly, from the software development company selected to conduct the case study, the findings of the first case study are presented in this paper. A brief description of the company followed by the case project description is discussed in the subsequent paragraphs.

The company where the case study is situated is a software development placed under academic institution. The development unit is a separate section founded specifically for application development for internal and external customers. The application development unit has autonomous business orientation in terms of engaging with any software development projects. The financial related tasks are under the hosting institution and clear agreement is established to manage the income associated with software development projects. At the time of the case study the software development department has 10 developers with varieties of job position (a team leader, programmers, system analyst, software architect, security expert). In terms of roles in a specific project, all experts take different roles according to the nature of the project to be developed.

In the past four years, the application development unit has been developing different types of software products to its customers. The software projects developed and being developed are business application software ranging from a simple desktop application, server based and web applications. At the commencement of the case study, the software development unit has started practicing agile methods mostly in unstructured manner. Practices selected from agile methods are used, but not in an organized way. The approach was through grouping two or three developers to take their development task using some agile practices, according to the convenience of the task at hand. In addition, practices in each iteration are not uniform and usually, no record of organized procedure or template is used. Software development capacitation programs are based on unplanned, but rather availability based training packages for developers. The main driver for the capacitation of the software development unit to produce better quality software products is not a predefined Software process improvement (SPI) plan. The selection and structuring of software development practices are not based on predefined and formalized approach. During completion of a project, the implementation procedures and related practices used for a particular software development, not revised and organized as experience package.

V. Case Study Discussion and Study Findings

The case study result presented in this paper is part of the ongoing case study research. This section presents the research process and the results obtained. It is based on the application of the integrated framework of agile and CMMI for process improvement. Since process improvement is a continuous activity, it is expected to develop in the context where it is implemented. The case study includes development of the first software project in the company discussed in the previous section.

A. Case Study Discussion

The case study started with training about the SPI framework developed in the research. In line with this, the guidelines to be used in doing the post-iteration and process improvement workshop is discussed. In addition, training on the operation of JIRA agile project management tool is also included which later used as one source of data for the study. Once the training is completed, a general discussion led by the development unit head and the researcher was made on selecting a project for the case. The first project, selected to be developed by the team, is a project management software. The project is selected targeting the need for software process improvement management environment and to establish the development environment based on the SPI guideline for the future development projects. The project management software (PMS) has functionalities to manage all activities of the ongoing projects and keep records of the accomplished tasks. The main features include task management, user management, and project and process data repository.

The development team started implementing the project following practices from scrum and XP of agile methods with available standard templates used. The source of most of the practices related templates and procedures is from "agile alliance" page. In line with this, the team started organizing the base practices based on scientific definition and procedures of standard methodologies. The base practices are not a complete copy of the standard methodologies, but they are selected practices to be used at the beginning of the first iteration. Team members can suggest a template or procedure for any practice from these methods, based on the practices from the standard methodologies. The improved practices are planned to be updated at the end of each iteration based on the achieved results of the PIPIW.

The development activity started in line with the implementation of SPI, based on the procedures of PIPIW, activities using the integrated framework. The project management software is developed in six iterations. The first two iterations took 20 days each and the rest iterations took 21 days. At the end of each iteration, the base practices database is updated with any improved template or procedure. In addition to updating the base practices, the process improvement tracking has a feature to manage the software proves the improvement. This feature is used to keep track of the process improvement achieved based on the PIPIW performed at the end of the iterations. Each PIPIW proposes practice related procedures or templates for the next iteration or improvement suggestions on the existing ones. The recommendation is based on the experience based practices and enhancements found positively affect the development activities in the completed iteration.

The improved practices proposed to be implemented in the next iteration are organized and submitted to the SPI team for analysis and evaluation. The SPI team performs the analysis of the practices in the base practices by identifying the relevant practices. The result of the analysis is used to update the base practice through improved templates and procedures. On the other hand, the discussion result from the SPI team is used to update the capability level of the specific practice. Before updating the capability any specific practice, the SPI team deliberates the procedures specified in Fig 1. to determine the institutionalization level of the practice. Once the level is determined capability level of the specific practices affected is updated based on the association mapping with the agile methods.

The SPI tracking record management is implemented through Microsoft excel application software. The specific practices pertinent to each KPA, and their institutionalization level, which is the measure of the capability level of the practices, is recorded. The initial record considers the levels to be started at level 0. This level is updated accordingly throughout the software development activities. The list of specific practices is stored in an excel sheet which represents the corresponding KPA where the practices belong. The values representing the capability level of specific practices are recorded in a range from 0 to 3 according to the capability levels defined in CMMI continuous representation. Based on the determined values the capability level of the KPA is represented graphically using a radar graph which gives a similar representation of the tracking model developed in Fig 2. The graph is used to visually demonstrate the status of KPAs, to guide the SPI planning based on the status of specific practices of each KPA.

B. Study Findings

In the following paragraphs, results of process improvement following the application of the framework is presented after the first project is completed in six iterations. Since the SPI activities defined in the framework follow the guidelines specified by The PIPW, the results of the study report follow the same procedure. The reported case study result focused on, what has been done in each phase of the PIPIW and the results discovered. The output of each iteration is presented by compiling the iterations with similar characteristic than discussing results from each iteration one by one.

First Iteration: - The team started the development task with practices and available templates and general procedures of the standard methodologies considered relevant at the beginning. In the first PIPIW, held at the end of the first iteration, some templates and procedures have been found to be relevant to be considered as good experience. The team collects those practices and proposed them to be followed in the upcoming similar tasks. But, the proposed procedures and templates of these practices were not considered as part of the base practices. This is done following the decision of the group members due to the fact that the development group just started organizing its experience. The procedures and templates have been included as part of the base practices. The development group was set free to consider them for the next iteration or add additional practices. Following the first iteration, the team organizes the practices with related procedures and templates in the base practice file.

Iteration (2 - 6): - Starting from the second iteration; the group started the development by selecting the practices, associated procedures, and templates from the base practice database. In the process of the development, the group used and accordingly made modifications to the procedures and templates. At the end of each iteration, the development group executes the PIPIW according to the predefined procedure. In each PIPIW, templates and procedures of practices used and considered as best experience in the development activities, are proposed. The proposed templates and procedures with their associated practices used in the development activities and modified to incorporate additional features. Every update, done in the templates and procedures, is initially selected by the team from the base practices. The detailed activities and reflections on the achieved results are presented in the subsequent paragraphs.

Identify and organize positive experiences: - In this step of the PIPIW, the team proposes practices for the next iteration, in each cycle. In due course of the development task, agile practices have been used according to their relevance for the task. Among the agile practices used, the team manages to develop a procedure to be followed, and template to be used to execute the associated agile practice. Starting from the first iteration, the development team reviewed the procedures and templates used in the development activities of the respective iterations. At the end of this phase of the PIPIW, the selected practices are organized with newly created or improved practices and proposed templates. The following snapshot of the base practice table. Later in the process, such data are planned to get updated at the end of each PIPIW and organized in the database of the PMS. Resources related to development activities are planned to be accessed from the PMS once the software is completed.

Table 2. List of base practices updated after six iterations

XP		
Practice	**Procedure**	**Template**
User story	Link	Link
Metaphor	Link	
On-site customer	Link	Link
Continuous integration:	Link	
Simple Design	Link	
Scrum		
Practice	**Procedure**	**Template**
Sprint Backlog	Link	Link
Product backlog	Link	Link
Sprint Review Meeting	Link	Link
Sprint Planning Meeting:	Link	Link
Daily Scrum Meeting	Link	

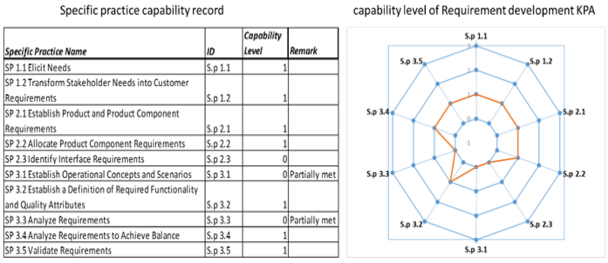
Define activities for next iteration: - From the practices and associated procedures and templates used, the team (the researcher and the department manager organize and lead the PIPIW) selects and propose those which found helpful to use them in the next iteration. In the second and third iteration, the team proposes additional templates and improvements on the existing ones. As an output of this phase of the PIPIW, the development team defines practices with their template and procedures for the upcoming iteration. The definition is in terms of the improved and newly proposed procedures and templates to be used. In the fourth and fifth iterations, most of the base practices were fully covered and the only modifications to the existing procedures and updates on templates were proposed by the development team.

Analysis of similarity: - In this phase of the PIPIW the SPI team evaluates the practices, procedures and associated templates at the end of every iteration. Using the integrated capability improvement process flow, the SPI team evaluates the state of institutionalization of those practices in the completed iteration. The SPI team strictly evaluates each practice (through procedures and templates) according to the requirements specified at each level of institutionalization process components. In the earlier two PIPIWs (2nd and 3rd) the procedures and templates of the agile practices used shown significant variation. In line with this, additional procedures and templates were proposed and included in the base practices. On some practices, the templates and procedures have shown a similar structure with additional features included. As a result, the SPI team decided to consider them as updates to the base practices.

The proposed procedures and templates of some practices shown reasonable consistency in the last three iterations (4 - 6). Following the corresponding PIPIWs, the SPI team proposes attainment of the first level of institutionalization of some agile practices. The level is claimed by some specific practices; through evaluation of the proposed procedures and templates in the last three iterations. Accordingly, the agile practices where their templates and procedures consistently proposed with similar structure by the development group in the last three iterations (4 to 6) considered being at level 1.

Mapping table referencing: - In this phase, the procedure defined in the steps of PIPIW executed to determine which specific practice is met by the agile practices. The SPI team used relationship matrix of CMMI and agile practices as a reference. Based on the evaluation of the capability level of practices of agile methods, specific practice of CMMI addressed by the agile practices is identified. During the first three iterations (1 to 3) the SPI team didn't go through this step since no process improvement is reported. The semi institutionalization level is achieved in some practices in the last three iterations (4-6). Following this, the SPI team identifies the specific practice of CMMI KPAs addressed by those practices. Using the agile and CMMI practice mapping table, the SPI team map the practices addressed based on the improvement achieved. During this phase, the SPI team considered the requirement specified in the PIPIW to determine the fully addressed specific practices of CMMI KPAs. Hence the process improvement achieved is formally recorded based on the CMMI continuous representation framework.

Indicating progress on the SPI tracking model: - At this level, a list of agile practices with their associated templates and procedures is compiled including their level of institutionalization. The earlier phase also identified with which specific practices they can be mapped. In this phase, the capability level of specific practices is updated in the SPI tracking model based on their level of institutionalization. The SPI team updates the capability level achieved in an excel tool following the identification of the associated specific practices addressed by agile practices used. The excel data sheet prepared for each KPA keep a record of the updated capability level and display the status graphically using radar graph. Using the graph, the capability level of each KPA presented based on the record updated on the corresponding data sheet. The SPI record, after completing the PMS software project carried out in six iterations, displays the SPI results achieved. Some selected KPAs with their capability improvement through their specific practices is presented by directly copying from the SPI record of the case site.



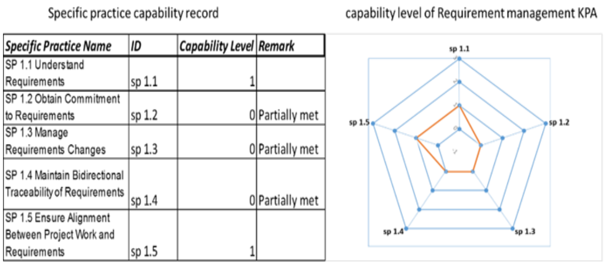
Specific practice capability record

Specific Practice Name	*ID*	*Capability Level*	*Remark*
SP 1.1 Elicit Needs	S.p 1.1	1	
SP 1.2 Transform Stakeholder Needs into Customer Requirements	S.p 1.2	1	
SP 2.1 Establish Product and Product Component Requirements	S.p 2.1	1	
SP 2.2 Allocate Product Component Requirements	S.p 2.2	1	
SP 2.3 Identify Interface Requirements	S.p 2.3	0	
SP 3.1 Establish Operational Concepts and Scenarios	S.p 3.1	0	Partially met
SP 3.2 Establish a Definition of Required Functionality and Quality Attributes	S.p 3.2	1	
SP 3.3 Analyze Requirements	S.p 3.3	0	Partially met
SP 3.4 Analyze Requirements to Achieve Balance	S.p 3.4	1	
SP 3.5 Validate Requirements	S.p 3.5	1	

capability level of Requirement development KPA

Fig.4. Snap of Requirements developments KPA capability record

The snap of capability tracking record and graph show capability level of requirements developments KPA as shown in Fig 4.. According to the graph, among ten specific practices of the requirements development, KPA seven specific practices reach the first capability level. The remaining three specific practices indicate where the SPI team should plan for the future steps or at least consider them as gaps to be filled. The SPI team put additional remark for the three specific practices to indicate that some agile practices mapped with those specific practices were qualified to the next level than indicated, but the specific practice is not fully addressed. This scenario relates to specific practices mapped with more than one agile practice, and some agile practices failed to meet the capability requirement at this level.



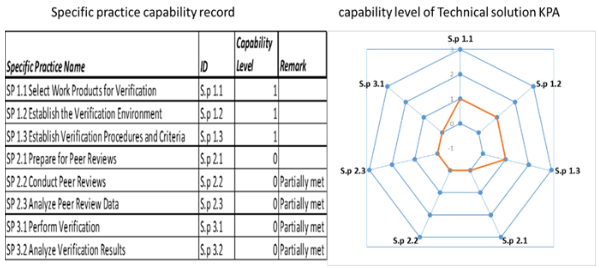
Specific practice capability record

Specific Practice Name	*ID*	*Capability Level*	*Remark*
SP 1.1 Understand Requirements	sp 1.1	1	
SP 1.2 Obtain Commitment to Requirements	sp 1.2	0	Partially met
SP 1.3 Manage Requirements Changes	sp 1.3	0	Partially met
SP 1.4 Maintain Bidirectional Traceability of Requirements	sp 1.4	0	Partially met
SP 1.5 Ensure Alignment Between Project Work and Requirements	sp 1.5	1	

capability level of Requirement management KPA

Fig.5. Snap of Requirements management KPA capability record

The capability record kept for requirement management KPA, by the development team, is shown in Fig 5. from the snap taken at the end of the project. As the record and graph show the KPA comprises five specific practices. From the five specific practices, at the completion of the first project, two of them qualified for the first capability level. The KPA capability shows that above half of its specific practices failed to claim the capability level one. The status of the KPA clearly shows where the effort for the future capacitation plan shall be focused on. Similarly, a specific practice in this requirement management with a level 0 doesn't mean it is not totally addressed, but all of the agile practices mapped to it didn't fully satisfy the requirement of the first level of institutionalization.



Specific practice capability record

Specific Practice Name	*ID*	*Capability Level*	*Remark*
SP 1.1 Select Work Products for Verification	S.p 1.1	1	
SP 1.2 Establish the Verification Environment	S.p 1.2	1	
SP 1.3 Establish Verification Procedures and Criteria	S.p 1.3	1	
SP 2.1 Prepare for Peer Reviews	S.p 2.1	0	
SP 2.2 Conduct Peer Reviews	S.p 2.2	0	Partially met
SP 2.3 Analyze Peer Review Data	S.p 2.3	0	Partially met
SP 3.1 Perform Verification	S.p 3.1	0	Partially met
SP 3.2 Analyze Verification Results	S.p 3.2	0	Partially met

capability level of Technical solution KPA

Fig.6 Snap of technical solution KPA capability record.

The previous two KPAs are selected from the record kept by the SPI team having a relatively varying capability in relative terms. From the fourteen KPAs in CMMI level 2 and 3 all of them has been part of the SPI activity with a wide difference of involvement. Some specific practices of the KPAs claim an improvement, but only one or few of them have shown improvements which left an assignment to the application development unit for future capacitation. Some of the agile practices such as pair programming and collective code ownership have been considered difficult to merge with the development culture of the unit as observed from the PIPIW. Hence defined procedure found difficult to develop and accordingly implement in the project. The unit has planned for on the job experience sharing with the practitioners with a better experience and training packages on selected practices. For comparison reason one of the KPA with the lowest capability record is also included in this research case study report.

One of the KPA with a relatively lower record of capability level after completion of the first project is verification KPA. As the snap taken from the development unit's record show, this process area shows one of the lower capability level records in relative terms. In this KPA more than half of the specific practices remain to be at level zero indicating that the templates and procedures are not well defined and habitualized to the level required at level one. Despite this, some specific practices show significant improvement useful for the development unit to keep them going. Based on the achieved record, the development unit can put some effort to qualify the KPA to higher capability level.

C. Post Project Review Discussion Results

At the end of the project, a three-day postmortem analysis is held in the development unit. In line with that data about the overall activity has been collected for the case study. The discussion is about the project performance and wide range of issues including the process improvement which is the subject of this study. The discussion made a review about all activities of the project including process improvement activities through evaluation of the capability status of each KPA. In doing so, the future steps to focus on SPI planning is analyzed and kept as a training plan for the application development unit. The KPA related with organizational training was considered to be addressed as a result of this discussion. Specifically addressing each specific practice is not evaluated as per the SPI framework including the PIPIW requirement defined in the earlier SPI actions. The plan related to identification of training plans focused on the specific practices with the lowest capability levels and some of them are not addressed by specific agile practice.

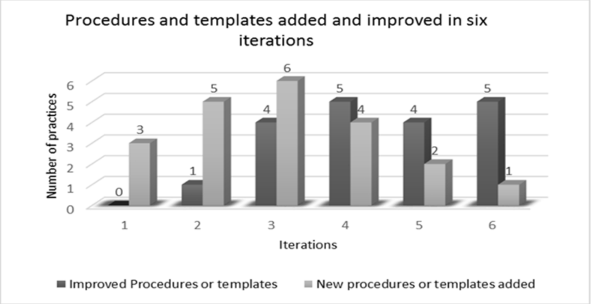


Fig.7. Improved and included practices and templates

Regarding the SPI activities, reflections on implementing the SPI framework has been collected from the participants describing the best sides and difficulties to fully utilize its benefits. The most important case study data is regarding the SPI activities performed based on the guideline defined in the components of the SPI framework. Accordingly, the templates and procedures developed and updated in each iteration have been summarized in the discussion. The number of procedures and templates has been represented in the graph in Fig 7..

The graph displays the compiled results of practice related procedures and templates introduced to the base practice database. It also shows improved procedures and templates at the end of each iteration. Templates and procedures discovered starting from the first iteration and according to the team's PIPIW results, some of them are recommended to be considered in the next iterations. The inclusion of new procedures and templates started in the first iteration and continued till the sixth iteration. The graph also displays records regarding improvement on the procedures and templates on the base practices throughout the six iterations.

In the first three iterations, the procedures and templates added to the base practice database increased as the team implement the practices and propose how the practices can be implemented through predefined procedures and templates. The inclusion of proposed templates and procedures continued to the last iteration but decreased after the third iteration. A template or procedure is considered to be new if the team executed the practice and develop an associated template or procedure for practice. In other cases, new templates and procedures were considered even though the team executed the practice with templates and procedures from the base practice, but define different procedure and/or template to be used for the next iterations. Hence, addressing all the practices through templates and procedures continued throughout the iterations.

The updated procedures and templates of base practices started with one in the second iteration as shown in the graph (Fig 7.). Following the second iteration, the number of updated procedures and templates increased as the team implement them in each iteration and propose extension on their components. Their number has shown similar status at the last iteration which indicates the update on the practice related procedures and templates may continue in the upcoming projects.

In general, the graph display how the procedures and template added to the base practice database throughout the six iterations. Discovery of new procedures and templates look decreasing towards the last iteration, implying the team keeps on following or adding enhancement on the existing ones. Concerning updates, even though it show some uniform pattern it looks like continuing in the iterations to come as well implying that the procedures and templates are not fully repeated.

VI. Conclusions and Future works

Even though the team just completed the first project, some practices have shown a significant difference in terms getting into the department development culture. The SPI activity has brought an opportunity to explicitly put knowledge skill and experience available to others. As it is explained by khan et al, a change in the development environment has a direct impact on the productivity of the development team [35]. This has given the chance to build organization knowledge base where the business application development department keep procedures, templates, techniques, and tools to be accessible for every member of the department. This can contribute developers to learn, guide their actions, and develop their skill in line with the improving organizational process capability.

The main focus given in the case study was to study how the integrated SPI framework with the PIPIW meet the sought effect. Following the six iterations, the development environment culture significantly upgraded to a new way of thinking. The advantage of proper documentation in scrum as explained by [36], stated how an improvement can benefit the team if proper tailoring of the method to include project related document management. The team responded positively in terms of organizing the base practices and add any improvements up on them. In addition, every team members develop a better understanding of the activities followed by the development team.

Process improvement is a prolonged activity which is expected to grow in the environment where it resides. The SPI is implemented at the micro level where the team learns from the experience at any phase of the development activity. In line with the alignment of SPI with the development task, made it easier to implement it and benefited it reasonably at a shorter time of commencement of organized SPI. The study prevailed important implications of the proposed approach to make more product focused SPI. Hence the integrated framework of SPI with the PIPIW and the associated components is being applied as a formal practice of daily development business. The case study will continue with the upcoming project for future evaluation of the framework.

Despite the encouraging results observed, the research is limited to demonstrating the application of integrated SPI framework in software development environment. The case results reported in this paper doesn't show a comparative analysis of the approach in terms of different factors of successful SPI implementation. The approach needs further rigorous analysis in different contexts. In addition, detailed empirical analysis of the context through comparative analysis via quantitative data can prevail reach insight into the area.

ACKNOWLEDGMENTS

The authors wish to thank Hawassa university ICT application development unit for hosting the case study and support the research process.

REFERENCES

[1] Pino, Francisco J., Félix Garcia, and Mario Piattini. "Key processes to start software process improvement in small companies." Proceedings of the 2009 ACM symposium on Applied Computing. ACM, 2009

[2] von Wangenheim, Christiane Gresse, Timo Varkoi, and Clênio F. Salviano. "Standard based software process assessments in small companies. "Software Process: Improvement and Practice 11.3 (2006): 329-335.

[3] Salviano, Clenio F., et al. "A method framework for engineering process capability models." 16th European Systems and Software Process Improvement and Innovation, Alcala, Spain (2009): 6-25.

[4] Pino, Francisco J., Félix García, and Mario Piattini. "Software process improvement in small and medium software enterprises: a systematic review." Software Quality Journal 16.2 (2008): 237-261.

[5] Dyba, Tore. "An empirical investigation of the key factors for success in software process improvement." IEEE Transactions on Software Engineering31.5 (2005): 410-424.

[6] Niazi, Mahmood, David Wilson, and Didar Zowghi. "Critical success factors for software process improvement implementation: an empirical study. "Software Process: Improvement and Practice 11.2 (2006): 193-211.

[7] von Wangenheim, Christiane Gresse, et al. "Systematic literature review of software process capability/maturity models." International Conference on Software Process Improvement and Capability Determination Spice. 2010.

[8] von Wangenheim, Christiane Gresse, et al. "Creating software process capability/maturity models." IEEE software 27.4 (2010): 92-94.

[9] Baddoo, Nathan, and Tracy Hall. "Demotivators for software process improvement: An analysis of practitioners' views." Quality control and applied statistics 49.3 (2004): 341-342.

[10] Staples, Mark, et al. "An exploratory study of why organizations do not adopt CMMI." Journal of systems and software 80.6 (2007): 883-895.

[11] Wong, Bernard, and Sazzad Hasan. "Software Process Improvement in Bangladesh." Software Engineering Research and Practice. 2006.

[12] Babar, Muhammad Ali, and Mahmood Niazi. "Implementing software process improvement initiatives: an analysis of Vietnamese practitioners' views."2008 IEEE International Conference on Global Software Engineering. IEEE, 2008.

[13] S. Tessler and A. Barr, "Software R & D Strategies of Developing Countries," pp. 1-17, 1996.

[14] Santos, Gleison, et al. "Implementing software process improvement initiatives in small and mediumsize enterprises in Brazil." Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference on the. IEEE, 2007.

[15] Patel, Chetankumar, and Muthu Ramachandran. "Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices." International Journal of Software Engineering, IJSE2.1 (2009): 3-28.

[16] Jakobsen, Carsten Ruseng, and Kent Aaron Johnson. "Mature Agile with a Twist of CMMI." Agile, 2008. AGILE'08. Conference. IEEE, 2008.

[17] Russwurm, W.: Hidden Treasure: The Implementation of CMMI Practices by Agile Methods. Procedures of the SEPG Conference (2010)

[18] Baker, Steven W. "Formalizing agility: an agile organization's journey toward CMMI accreditation." Agile Development Conference (ADC'05). IEEE, 2005.

[19] Marçal, Ana Sofia C., et al. "Blending Scrum practices and CMMI project management process areas." Innovations in Systems and Software Engineering 4.1 (2008): 17-29.

[20] M. Pikkarainen and A. Mäntyniemi, "An Approach for Using CMMI in Agile Software Development Assessments : Experiences from Three Case Studies," in

SPICE, 2006, no. May.

[21] Lukasiewicz, Katarzyna, and J. Miler. "Improving agility and discipline of software development with the Scrum and CMMI." IET software 6.5 (2012): 416-422.

[22] Anderson, David J. "Stretching Agile to fit CMMI level 3." Microsoft Corporation (2005).

[23] B. I. Benbasat, “The Case Research Strategy in Studies of Information Systems Case Research : Definition,” MIS Quarterly, vol. 11, no. 3, pp. 369-386, 1987.

[24] Fritzsche, Martin, and Patrick Keil. "Agile methods and CMMI: compatibility or conflict?." e-Informatica Software Engineering Journal 1.1 (2007).

[25] T. Kähkönen and P. Abrahamsson, “Achieving CMMI Level 2 with Enhanced Extreme,” in PROFES, 2004, no, pp. 378-392.

[26] T. Engdashet, R. J. Machado, and D. Midekso, “Integrated Framework of Agile and CMMI : An Alternative Path towards Product Focused SPI for Small Companies,” Lecture Notes in Software Engineering, vol. 4, no. 1, pp. 1-6, 2016.

[27] Chrissis, Mary Beth, Mike Konrad, and Sandra Shrum. CMMI for development: guidelines for process integration and product improvement. Pearson Education, 2011.

[28] Tolbert, Pamela S., and Lynne G. Zucker. "The institutionalization of institutional theory." Studying Organization. Theory & Method. London, Thousand Oaks, New Delhi (1999): 169-184.

[29] Wiseman, E. R. I. C. A. "The institutionalization of organizational learning: A neoinstitutional perspective." Proceedings of OLKC (2007).

[30] Salo, Outi. "Improving software process in agile software development projects: results from two XP case studies." Euromicro Conference, 2004. Proceedings. 30th. IEEE, 2004.

[31] R. K. Yin, Case study research Design and Methods, 3rd ed. Delhi India: SAGE Publications, 2003.

[32] Per Runeson and Martin Höst, “Guidelines for conducting and reporting case study research in software engineering,” Empirical Software Engineering, vol. 14, no. 2, pp. 131-164, 2008.

[33] P. Runeson, M. H. St, A. Rainer, and B. R. Regnell, Case Study Research in Software Engineering: Guidelines and Examples. United States of America: John Wiley & Sons, Inc, 2012.

[34] Runeson, Per, and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering." Empirical software engineering14.2 (2009): 131-164.

[35] Khan, M. Y. A., & Sayed, M. A. (2017). A Simple Software Engineering Environment for Coming Decades. *International Journal of Education and Management Engineering (IJEME).* 7(1) PP.46-53, 2017

[36] Ashraf, S., & Aftab, S. (2017). IScrum: An Improved Scrum Process Model. *International Journal of Modern Education and Computer Science, 9*(8), PP16-24. 2017

Authors’ Profiles

Tatek Engdashet is assistance professor of software engineering. Tatek has been a lecturer in Institute of Technology Hawassa University, Ethiopia in the Department of Informatics. His research interests include software process improvement, agile methods, software measurement, and process improvement and knowledge management.

Dida Midekso is Associate Professor of Computer Science and software engineering in Addis Ababa University (AAU). Coordinator of the Software Engineering track in the IT Ph.D. program. Has been served as head of the Mathematics Department who also oversees the CS undergraduate program at the college on Natural sciences. He specialized in computer aided learning. His research interests include software engineering, Software project management, and pervasive information systems.

Ricardo J. Machado is Full Professor at the Department of Information Systems (DSI) and Director of the ALGORITMI Research Centre, School of Engineering of University of Minho (U.Minho). Holds U.Minho’s Chair for Information Systems Engineering and Technology (ETSI) [2014-]. Member of the Scientific Council of the School of Engineering of U.Minho [2013-]. Member of the Coordinating Committee of the Department of Information Systems [2010-].

How to cite this paper: Tatek Engdashet Kabitimer, Dida Midekso, Ricardo J. Machado, "Product Focused Software Process Improvement through Integrated Framework of Agile and CMMI: A Case in Small Settings", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.5, pp.55-67, 2018. DOI: 10.5815/ijitcs.2018.05.06